GEOMETRIC REPRESENTATION CONDITION IMPROVES EQUIVARIANT MOLECULE GENERATION

Anonymous authors
Paper under double-blind review

ABSTRACT

Recent advancements in molecular generative models have demonstrated substantial potential in accelerating scientific discovery, particularly in drug design. However, these models often face challenges in generating high-quality molecules, especially in conditional scenarios where specific molecular properties must be satisfied. In this work, we introduce GeoRCG, a general framework to enhance the performance of molecular generative models by integrating geometric representation conditions. We decompose the molecule generation process into two stages: first, generating an informative geometric representation; second, generating a molecule conditioned on the representation. Compared to directly generating a molecule, the relatively easy-to-generate representation in the first-stage guides the second-stage generation to reach a high-quality molecule in a more goal-oriented and much faster way. Leveraging EDM (Hoogeboom et al., 2022) as the base generator, we observe significant quality improvements in unconditional molecule generation on the widely-used QM9 and GEOM-DRUG datasets. More notably, in the challenging conditional molecular generation task, our framework achieves an average 31% performance improvement over state-of-the-art approaches, highlighting the superiority of conditioning on semantically rich geometric representations over conditioning on individual property values as in previous approaches. Furthermore, we show that, with such representation guidance, the number of diffusion steps can be reduced to as small as 100 while maintaining superior generation quality than that achieved with 1,000 steps, thereby significantly accelerating the generation process.

1 INTRODUCTION

Recent years have seen rapid development in generative modeling techniques for molecule generation (Garcia Satorras et al., 2021; Hoogeboom et al., 2022; Luo & Ji, 2022; Wu et al., 2022; Xu et al., 2023; Morehead & Cheng, 2024), which have demonstrated great promise in accelerating scientific discoveries such as drug design (Graves et al., 2020). By modeling molecules as *point clouds of chemical elements* embedded in Euclidean space and employing equivariant models as backbone architectures, such as EGNN (Satorras et al., 2021), these approaches can ensure the *O(3)- (or SO(3)-) invariance* of the modeled molecule probability and have shown significant promise in both unconditional and conditional molecule generations.

Despite the advances, precisely modeling the molecular distribution $q(\mathcal{M})$ still remains a challenge, with current models often falling short of satisfactory results. This is especially true in the more practical scenarios where the goal is to capture the conditional distribution $q(\mathcal{M}|c)$ for conditional generation, with c representing a desired property such as the HOMO-LUMO gap. In such cases, recent models still produce molecules with property errors *significantly larger* than the data lower bound (Hoogeboom et al., 2022; Xu et al., 2023). Such challenge arises partly because molecules are naturally supported on a lower-dimensional manifold (Mislow, 2012; De Bortoli, 2022; You et al., 2023), yet are embedded in a 3D space with much higher ambient dimensions ($N \times (3 + d)$, where N is the number of atoms and d the atom feature dimension). Consequently, directly learning these distributions without additional guidance or conditioning solely on a single property can result in substantial errors (Song et al., 2021), often leading to unstable or undesirable molecular samples.

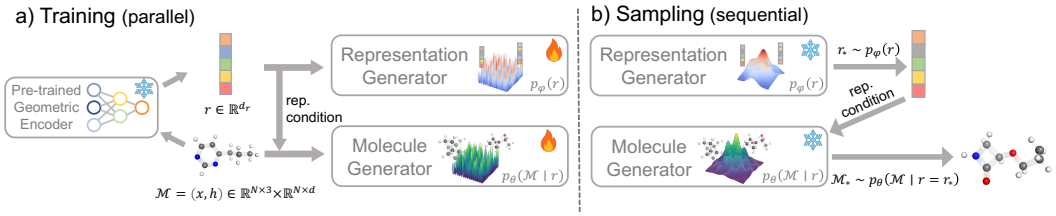


Figure 1: Training and sampling procedure of GeoRCG in unconditional molecule generation scenario. a) During training, each molecule $\mathcal{M}$ is mapped into an informative representation r by a pre-trained, frozen geometric encoder E. The distribution of representations is then learned by a lightweight representation generator. The molecule generator is trained in a self-conditioned manner, generating a molecule $\mathcal{M}$ conditioned on its own representation $E(\mathcal{M})$. b) During sampling, an informative representation is first generated, which subsequently guides the molecule generator to produce high-quality molecules.

In this work, we propose GeoRCG (**Geo**metric-**R**epresentation-**C**onditioned Molecule **G**eneration), a general framework for improving the generation quality of molecular generative models by leveraging *geometric representation conditions* for both unconditional and conditional generation. See Figure 1 for an overview of the framework. At a high level, rather than directly learning the extrinsic molecular distribution, we aim to first transform it into a more compact and semantically meaningful representation distribution, with the help of a well-pre-trained geometric encoder E such as Unimol (Zhou et al., 2023) and Frad (Feng et al., 2023). This distribution is much simpler because it does not exhibit any group symmetries, such as $O(3)/SO(3)$ and $S(N)$ groups which are present in extrinsic molecular distributions. As a result, a lightweight representation generator (Li et al., 2023) can effectively capture this reduced distribution. In the second stage, we employ a standard molecular generator to achieve the ultimate objective: molecular generation. Unlike conventional approaches, our molecular generator is directly informed by the first-stage geometric representation, which encapsulates crucial molecular structure and property information. This guidance enables the generation of high-quality molecular structures with improved fidelity.

Our approach is directly inspired by RCG (Li et al., 2023), which, however, focuses on image data with fixed size and positions and does not necessitate handling Euclidean and permutation symmetries—factors that are markedly different in molecular data. Compared to recent work GraphRCG (Wang et al., 2024), which applies the RCG framework to 2D graph data, we explicitly handle 3D geometry that is more complex due to the additional Euclidean symmetry. Moreover, we avoid the complicated step-wise bootstrapped training and sampling process proposed in Wang et al. (2024) that requires noise alignment, sequential training, and simultaneous encoder training. Instead, we employ a simple and intuitive framework that enables parallel training and leverages advanced pre-trained geometric encoders containing valuable external knowledge (Zaidi et al., 2022; Feng et al., 2023), thus achieving competitive results while avoiding complex training procedures. Notably, while Li et al. (2023) primarily focus on empirical evaluation, we also provide *theoretical characterizations* of general representation-conditioned diffusion models for both unconditional and conditional generation, which offers a rigorous understanding of the improved performance.

To illustrate the effectiveness of our approach, we select one of the *simplest* and most classical equivariant generative models, EDM (Hoogeboom et al., 2022), as the base molecular generator of GeoRCG. Experimentally, our method achieves the following significant improvements:

- Substantially enhancing the quality (e.g., molecule stability) of the generated molecules on the widely used QM9 and GEOM-DRUG datasets. On QM9, GeoRCG not only improves the performance of EDM by a large margin, but also significantly surpasses several recent baselines with state-of-the-art performance (Wu et al., 2022; Xu et al., 2023; Morehead & Cheng, 2024; Song et al., 2024a).
- More remarkably, in conditional molecule generation tasks, GeoRCG yields an average **31% improvement** in performance (i.e., difference of generated molecule's property with specified conditions), while many contemporary models struggled to achieve even *marginal gains*.
- By incorporating classifier-free guidance in the molecule generator (Li et al., 2023) and employing low-temperature sampling for representation generation (Ingraham et al., 2023),

GeoRCG demonstrate a **flexible trade-off between molecular quality and diversity** without additional training, which is especially advantageous in molecular generation tasks that prioritizes quality over diversity.

- With the assistance of the representation guidance, GeoRCG can significantly **reduce the number of diffusion steps** required by approximately 10x, while preserving the molecular generation quality, thereby considerably accelerating the generation process.

2 Related Works

Molecular Generative Models Early work has primarily focused on modeling molecules as 2D graphs (composed of atom types, connection, and edge types), utilizing 2D graph generative models to learn the graph distribution (Vignac et al., 2022; Jang et al., 2023; Jo et al., 2023; Luo et al., 2023; Zhou et al., 2024). However, since molecules inherently exist in 3D space, where physical laws govern their behavior and geometry provides critical information related to key properties, recent research has increasingly focused on utilizing 3D generative models to directly learn the *geometric* distribution by modeling molecules as point clouds of chemical elements. Notable early autoregressive models include G-SchNet (Gebauer et al., 2019) and G-SphereNet (Luo & Ji, 2022). More recently, diffusion models have demonstrated effectiveness in this domain, as evidenced by models like EDM (Hoogeboom et al., 2022) and subsequent advancements (Xu et al., 2023; Wu et al., 2022; Morehead & Cheng, 2024) that enhance EDM with latent space, prior information and more powerful backbones respectively. Furthermore, recent advances in flow methods (Lipman et al., 2022; Liu et al., 2022b) have inspired the development of geometric, equivariant flow methods including EquiFM (Song et al., 2024b) and GOAT (Hong et al., 2024), which can provide much faster molecule generation speed. Beyond these, there are also methods jointly model 2D and 3D information (Vignac et al., 2023; You et al., 2023; Irwin et al., 2024) (also called 3D graph (You et al., 2023)), where a representative method is MiDi (Vignac et al., 2023) which uses a diffusion framework to jointly diffuse and denoise atom type, bond type, formal charges and coordinates.

Latent Generative Models At a high level, our framework can also be viewed as a latent generative model, where data distributions are learned in a latent space (our stage 1) and decoded back through some decoder (our stage 2). Most prior works in this domain either focus on regular data forms with fixed positions and sizes (Van Den Oord et al., 2017; Razavi et al., 2019; Dai & Wipf, 2019; Aneja et al., 2021; Rombach et al., 2022; Li et al., 2023), or on data without Euclidean symmetry and require explicit modeling (Wang et al., 2024). Molecular data, however, presents unique challenges in both aspects. One of the key issues in this context is how to define the latent space—defining it as "latent coordinates and features" as in GeoLDM (Xu et al., 2023) still results in a geometrically structured and thus complex space, while defining it on representations as we do introduces the challenge of effectively "decoding" a global, non-symmetric embedding back into geometric objects. LGD (Zhou et al., 2024) trains a diffusion model on a unified Euclidean latent space obtained by jointly training a powerful encoder and a simple decoder, and performs both generation and prediction tasks focusing on 2D graphs. LDM-3DG (You et al., 2023) also adopts representation latent space but employs a cascaded (2D+3D) auto-encoder (AE) framework, where the decoder is designed (or trained) to be *deterministic*, rendering poor performance on the 3D part as evidenced in our experiments. In contrast, we model the decoder as a powerful *generative model*, focusing solely on geometric learning while demonstrating superior effectiveness. We leave the related works of pre-trained geometric encoders in Appendix A.

3 Methods

3.1 Preliminaries

In this work, we represent molecules as point clouds of chemical elements in 3D space, denoted by $\mathcal{M} = (\mathbf{x}, \mathbf{h})$, where $\mathbf{x} = (\mathbf{x}_1, \ldots, \mathbf{x}_N) \in \mathbb{R}^{N\times 3}$ represents the atomic coordinates of N atoms, and $\mathbf{h} = (\mathbf{h}_1, \ldots, \mathbf{h}_N) \in \mathbb{R}^{N\times d}$ captures the node features of dimension d, such as atomic numbers and charges. This formulation follows the approach of Hoogeboom et al. (2022); Xu et al. (2023); Morehead & Cheng (2024) and is widely utilized in molecular representation learning (Thomas et al., 2018; Li et al., 2024a; Zaidi et al., 2022), facilitating the integration of pre-trained molecular

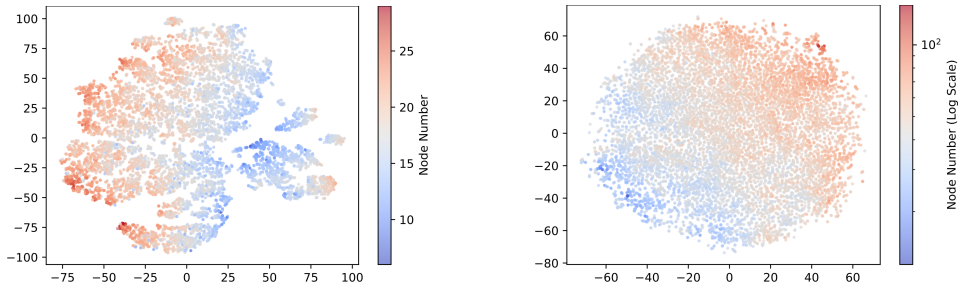


Figure 2: T-SNE visualizations of the representations produced by Frad (Feng et al., 2023) for the QM9 dataset (left) and by Unimol (Zhou et al., 2023) for the GEOM-DRUG dataset (right). The representations exhibit clear clustering based on node count.

encoders (Zaidi et al., 2022; Feng et al., 2023). We use q to denote the underlying data distribution, such as molecule distributions $q(\mathcal{M})$, and p to denote the approximated distributions modeled by parametric methods.

We denote the pre-trained geometric encoder as $E : \bigcup_{N=1}^{+\infty}(\mathbb{R}^{N\times 3} \times \mathbb{R}^{N\times d}) \to \mathbb{R}^{d_r}$, which embeds a molecule $\mathcal{M}$ with an arbitrary number of nodes N into a representation vector r of fixed dimension d_r. The geometric encoder exhibits *E(3)- (or SE(3)-) invariance*, meaning that $E(\mathcal{M}) = E((\mathbf{x}, \mathbf{h})) = E((\mathbf{x}\mathbf{R}^T + \mathbf{t}, \mathbf{h}))$ for any $\mathbf{t} \in \mathbb{R}^3$ and $\mathbf{R} \in O(3)$ (or $SO(3)$), where $O(3)$ is the set of orthogonal matrices (and $SO(3)$ being the set of special orthogonal matrices).

3.2 GeoRCG: Geometric-Representation-Conditioned Molecular Generation

Geometric Representation Generator To improve the quality of the generated molecules, we propose first transforming the geometrically structured molecular distribution $q(\mathcal{M})$ into a non-geometric representation distribution $q(r)$ using a well-pre-trained geometric encoder E that maps each molecule $\mathcal{M}$ to its representation r. Learning the representation distribution $q(r)$ is considerably easier, since representations do not exhibit any symmetry as in explicit molecular generative models (Xu et al., 2022; Hoogeboom et al., 2022). We thus leverage a simple yet effective MLP-based diffusion architecture proposed in (Li et al., 2023) for the representation generator $p_\varphi(r)$, which adopts the DDIM architecture (Song et al., 2020) with MLP backbones and is optimized via the denoising score matching scheme (Vincent, 2011).

One deviation from previous practices (Li et al., 2023; Wang et al., 2024) is that we additionally condition the representation generator on the molecule's node number N *by default*[1]. This is crucial for ensuring *consistency* between the size of the representation's underlying molecule and the size of the molecule it guides to generate. Moreover, molecules with different sizes often have distinct modes in structures and properties (Hoogeboom et al., 2022), which is reflected in their geometric representations learned by modern pre-trained geometric encoders (Zhou et al., 2023; Feng et al., 2023), as shown in Figure 2. From the figures, it is evident that by conditioning on N, the learning process for the representation generator becomes simpler and more effective, leading to the following loss function of our representation generator:

$$\mathcal{L}_{\text{rep}} = \mathbb{E}_{(r,N)\in\mathcal{D}_{\text{train}}^{\text{rep}},\epsilon\sim\mathcal{N}(0,I),t}\left[||r - f_\varphi(r_t, t, N)||^2\right], \tag{1}$$

where $\mathcal{D}_{\text{train}}^{\text{rep}} = \{\!\{(E(\mathcal{M}), N(\mathcal{M}))|\mathcal{M} \in \mathcal{D}_{\text{train}}^{\text{mol}}\}\!\}$, with $N(\mathcal{M})$ representing atom number of $\mathcal{M}$ and $\mathcal{D}_{\text{train}}^{\text{mol}}$ denoting the molecule dataset. Here, f_φ is the MLP backbone (Li et al., 2023), and $r_t = \sqrt{\alpha_t}r + \sqrt{1-\alpha_t}\epsilon$ is the noisy representation computed with the predefined schedule $\alpha_t \in (0, 1]$.

Molecule Generator Since the ultimate goal of our framework is to generate molecules from $q(\mathcal{M})$, we decompose the molecular distribution as $q(\mathcal{M}) = \int q(\mathcal{M}|r)q(r)\,dr$ to explicitly enable geometric-representation conditions. Consequently, a geometric-representation-conditioned

[1]We omit the condition N in our probability decompositions and mathematical derivations for statement simplicity, as its inclusion does not affect the overall framework and conclusions.

molecular generator $p_\theta(\mathcal{M}|r)$ is required. In principle, we can use many modern molecule generators (Hoogeboom et al., 2022; Xu et al., 2023; Morehead & Cheng, 2024), as these models can all take additional conditions. To illustrate the effectiveness of our approach, however, we choose a relatively simple model EDM (Hoogeboom et al., 2022) as the base generator. EDM is designed to ensure the $O(3)$-invariance, i.e., for any $\mathbf{R} \in O(3)$, $p_\theta(\mathcal{M}) = p_\theta(\mathbf{x}, \mathbf{h})) = p_\theta((\mathbf{x}\mathbf{R}^T, \mathbf{h}))$. To accomodate EDM to representation conditions, we use the following training objective:

$$\mathcal{L}_{\text{mol}} = \mathbb{E}_{(\mathcal{M},r)\sim\mathcal{D}_{\text{train}}^{\text{mol-rep}},t\sim\mathcal{U}(0,T),\epsilon\sim\hat{\mathcal{N}}(0,\mathbf{I})}\left[||\epsilon - f_\theta(\mathcal{M}_t, t, r)||^2\right], \tag{2}$$

where $\mathcal{D}_{\text{train}}^{\text{mol-rep}} = \{\!\{(\mathcal{M}, E(\mathcal{M}))|\mathcal{M} \in \mathcal{D}_{\text{train}}^{\text{mol}}\}\!\}$, and sampling from $\hat{\mathcal{N}}(0, \mathbf{I})$ entails drawing $\epsilon_0 = [\epsilon_0^{(x)}, \epsilon_0^{(h)}]$ from $\mathcal{N}(0, \mathbf{I})$, adjusting $\epsilon_0^{(x)}$ by subtracting its geometric center to obtain $\epsilon^{(x)}$, and setting $\epsilon = [\epsilon^{(x)}, \epsilon_0^{(h)}]$. This ensures the zero center-of-mass property, as the distribution is defined on this subspace to ensure translation invariance (Hoogeboom et al., 2022). The noisy molecule is given by $\mathcal{M}_t = \alpha_t^{(\mathcal{M})}[\mathbf{x}, \mathbf{h}] + \sigma_t^{(\mathcal{M})}\epsilon$, with time-dependent schedules $\alpha_t^{(\mathcal{M})}$ and $\sigma_t^{(\mathcal{M})}$, while the diffusion backbone f_θ, which is instantiated with EGNN (Satorras et al., 2021), is conditioned on r.

Combining the Two Generators Together The representation generator $p_\varphi(r)$ and the molecule generator $p_\theta(\mathcal{M}|r)$ together model the molecular distribution $p_{\varphi,\theta}(\mathcal{M}) = \int p_\theta(\mathcal{M}|r)p_\varphi(r)\,dr$, which approximates the data distribution $q(\mathcal{M}) = \int q(\mathcal{M}|r)q(r)\,dr$ that we aim to capture. One notable advantage of the framework is that the decomposition enables **parallel training** of the two generators. The entire training and sampling procedure is summarized in Algorithm 1.

There are several key properties of GeoRCG that facilitate high-quality molecule generation. First, GeoRCG preserves all symmetry properties of the base molecule generator $p_\theta(\mathcal{M})$:

Proposition 3.1. ***(Symmetry Preservation)*** *Assume the original molecular generator $p_\theta(\mathcal{M})$ is O(3)- or SO(3)-invariant. Then, the two-stage generator $p_{\varphi,\theta}(\mathcal{M})$ is also O(3)- or SO(3)-invariant.*

Proof. This result follows directly from the definition. Specifically, $p_{\varphi,\theta}(\mathcal{M}) = \int p_\theta(\mathcal{M}|r)\,p_\varphi(r)\,dr = \int p_\theta((\mathbf{x}\mathbf{R}^T, \mathbf{h})|r)\,p_\varphi(r)\,dr = p_{\varphi,\theta}((\mathbf{x}\mathbf{R}^T, \mathbf{h}))$ for any $\mathbf{R} \in O(3)$ (or $SO(3)$). The second equality holds due to the symmetric property of $p_\theta(\mathcal{M})$, which remains valid when additional non-symmetric conditions r are applied. □

Moreover, representation-conditioned diffusion models can achieve no higher overall total variation distance than traditional diffusion models, and can arguably yield better results, as the representation encodes key data information that may further reduce estimation error. We present the rigorous bound in Theorem 3.1, and provide the proof and detailed discussions in Appendix E.1. Notably, this is a *general theoretical characterization* that applies to prior *experimental* work (Li et al., 2023).

Theorem 3.1. *Consider the random variable $x \in \mathbb{R}^{d+3} \sim q(x)$, and assume that the second moment m_x of x is bounded as $m_x^2 := \mathbb{E}_{q(x)}[\|x-\bar{x}\|^2] < \infty$, where $\bar{x} := \mathbb{E}_{q(x)}[x]$. Further, assume that the score $\nabla \ln q(x_t)$ is L_x-Lipschitz for all t, and that the score estimation error in the second-stage diffusion is bounded by $\epsilon_{\varphi,\theta,cond}$ such that $\mathbb{E}_{r\sim p_\varphi(r),\, x_t\sim q_t(x_t|r)}[\|s_\theta(x_t, t, r) - \nabla \ln q_t(x_t|r)\|^2] \le \epsilon_{\varphi,\theta,cond}^2$. Denote the step size as $h := T/N$, where T is the total diffusion time and N is the number of discretization steps, and assume that $h \preceq 1/L_x$. Suppose we sample $x \sim p_\theta(x|r)$ from Gaussian noise, where $r \sim p_\varphi(r)$, and denote the final distribution of x as $p_{\theta,\varphi}(x)$. Define $p_0^{q_{T|\varphi}}$, which is the endpoint of the reverse process starting from $q_{T|\varphi}$ instead of Gaussian noise. Here, $q_{T|\varphi}$ is the T-th step in the forward process starting from $q_{0|\varphi} := \frac{1}{A}\int_r q(x_0|r)p_\varphi(r)\,dr$, where A is the normalization factor. Denote k-dim isotropic Gaussian distribution as γ^k. Then the following holds,*

$$\text{TV}(p_{\theta,\varphi}(x), q(x)) \preceq \underbrace{\sqrt{\text{KL}(q_{0|\varphi}||\gamma^{d+3})}\exp(-T)}_{\textit{convergence of forward process}} + \underbrace{(L_x\sqrt{(d+3)h} + L_x m_x h)\sqrt{T}}_{\textit{discretization error}} \tag{3}$$

$$+ \underbrace{\epsilon_{\varphi,\theta,\text{cond}}\sqrt{T}}_{\textit{conditional score estimation error}} + \underbrace{\text{TV}(q_{0|\varphi}, q_0)}_{\textit{representation generation error}} \tag{4}$$

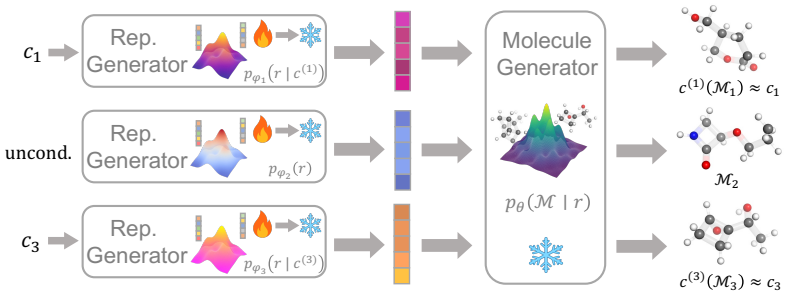


Figure 3: A single molecule generator can be employed for both unconditional and conditional molecule generation with respect to various properties. For conditional generation, only the representation generator is re-trained on (molecule, property) pairs, allowing it to conditionally sample property-meaningful representations during the sampling stage.

Balancing Quality and Diversity of Molecule Generation In many scientific applications, researchers prioritize generating higher-quality molecules over more diverse ones. To address this preference, we introduce a feature that allows fine-grained control over the trade-off between diversity and quality *in the sampling stage (thus without retraining)*. This is achieved by integrating two key techniques: low-temperature sampling (Ho & Salimans, 2022) (controlled via the temperature $\mathcal{T}$) for the representation generator, and classifier-free guidance (Ho & Salimans, 2022) (controlled via the coefficient w) for the molecule generator. We provide more details about the two techniques in Appendix B. Combining the two features enables a flexible and explicit control, which we refer to as "Balancing Controllablility" and demonstrate its effectiveness in Section 4.2.

Handling Conditional Molecule Generation The framework discussed thus far focuses on unconditional molecule generation, where no specific property c (e.g., HOMO energy) is pre-specified. However, for molecule generation, a more practical and desired scenario is conditional (also called controllable) generation, where additional conditions c, such as HOMO-LUMO gap energy, is introduced, and our objective shifts to generating molecules from the distribution $q(\mathcal{M}|c)$. In GeoRCG, this conditional generation is naturally decomposed as $p_{\theta,\varphi}(\mathcal{M}|c) = \int p_\theta(\mathcal{M}|r)p_\varphi(r|c)\,dr$, meaning that we first generate a "property-meaningful" molecular representation r, which is then independently used to condition the second-stage molecule generation. See Figure 3 for illustration. A key advantage of this modeling approach is that, when different properties (e.g., HOMO, LUMO, GAP energy) need to be captured, **only the representation generator requires retraining** under the new conditions. This retraining is highly efficient due to the lightweight nature of the representation generator. Notably, GeoRCG demonstrates outstanding conditional generation performance, as shown in Section 4.3. Moreover, we theoretically demonstrate that, under mild assumptions, the representation generator can provably estimate the conditional distribution and generate representations that lead to provable reward improvements toward the target, which subsequently benefits the second-stage generation. Further details are provided in Appendix E.2.

4 Experiments

4.1 Experiment Setup

Datasets and Tasks As a method for 3D molecule generation, we evaluate GeoRCG on the widely used datasets QM9 (Ramakrishnan et al., 2014) and GEOM-DRUG (Gebauer et al., 2019; 2022; Axelrod & Gomez-Bombarelli, 2022). We focus on two tasks: unconditional molecule generation, where the goal is to sample from $q(\mathcal{M})$, and conditional (or controllable) molecule generation, where a property c is given, and we aim to sample from $q(\mathcal{M}|c)$. To ensure fair comparisons, we follow the dataset split and configurations exactly as in Anderson et al. (2019); Hoogeboom et al. (2022); Xu et al. (2023). Without further clarification, we **bold** the highest scores and underline the second-highest one. Additionally, to illustrate the direct improvement over our base model, EDM (Hoogeboom et al., 2022), we display green numbers next to the score to indicate the av-

Table 1: Quality comparison of unconditional molecular generation across different methods. The gray cells denotes the base molecule generator employed in GeoRCG.

Methods \ Metrics	QM9				DRUG	
	Atom Sta (%) ↑	Mol Sta (%) ↑	Valid (%) ↑	Valid & Unique (%) ↑	Atom Sta (%) ↑	Valid (%) ↑
Data	99	95.2	97.7	97.7	86.5	99.9
G-Schnet	95.7	68.1	85.5	80.3	-	-
GDM	97	63.2	-	-	75	90.8
GDM-AUG	97.6	71.6	90.4	89.5	77.7	91.8
GraphLDM	97.2	70.5	83.6	82.7	76.2	97.2
GraphLDM-AUG	97.9	78.7	90.5	89.5	79.6	98
EDM	98.7	82	91.9	90.7	81.3	92.6
EDM-Bridge	98.8	84.6	92	90.7	82.4	92.8
GeoLDM	98.9(0.1)	89.4(0.5)	93.8(0.4)	92.7(0.5)	84.4	**99.3**
GCDM	98.7(0.0)	85.7(0.4)	94.8(0.2)	93.3(0.0)	**89**	95.5
ENF	85	4.9	40.2	39.4	-	-
EquiFM	98.9(0.1)	88.3(0.3)	94.7(0.4)	**93.5(0.3)**	84.1	98.9
GOAT	98.4	84.1	90.9	89.99	81.8	96.0
GeoBFN (1k)	99.08(0.03)	90.87(0.1)	95.31(0.1)	92.96(0.1)	86.1	91.66
GeoRCG	**99.12(0.03)** 0.43%	**92.32(0.06)** 12.59%	**96.52(0.2)** 5.03%	92.45(0.2) 1.93%	84.3(0.12) 3.69%	98.5(0.12) 6.37%

erage improvement, and red numbers to denote a decrease. All results are calculated based on 10k randomly sampled molecules, averaged over three runs, with standard errors reported in parentheses.

Instantiation of the Pre-trained Encoder We employ Frad (Feng et al., 2023), which was pre-trained on the PCQM4Mv2 dataset (Nakata & Shimazaki, 2017) using a hybrid noise denoising objective, as the geometric encoder for QM9 dataset. For GEOM-DRUG, we adopt Unimol (Zhou et al., 2023) and pre-train it on the GEOM-DRUG dataset itself, since GEOM-DRUG contains distinct chemical elements not present in PCQM4Mv2 or other commonly used pre-training datasets such as ZINC or ChemBL (Li et al., 2021).

Baselines A direct comparison is made with our base molecule generator, EDM (Hoogeboom et al., 2022). Additionally, we benchmark against the non-equivariant counterparts of EDM and GeoLDM (Xu et al., 2023), namely GDM(-AUG) (Hoogeboom et al., 2022) and GraphLDM(-AUG) (Xu et al., 2023), as well as the autoregressive method G-SchNet (Gebauer et al., 2019). For further comparison, we include advanced equivariant diffusion models like GeoLDM (Xu et al., 2023), EDM-Bridge (Wu et al., 2022), and GCDM (Morehead & Cheng, 2024), along with fast equivariant flow-based methods such as E-NF (Garcia Satorras et al., 2021), EquiFM (Song et al., 2024b), and GOAT (Hong et al., 2024). A recent Bayesian-based method GeoBFN (Song et al., 2024a) is also considered for its high-quality samples. Finally, we extend the comparison to 2D and 3D methods like MiDi (Vignac et al., 2023) and LDM-3DG (You et al., 2023).

Additionally, we provide further experiments, including ablation studies on the pre-trained encoder, in Appendix D.

4.2 Unconditional Molecule Generation

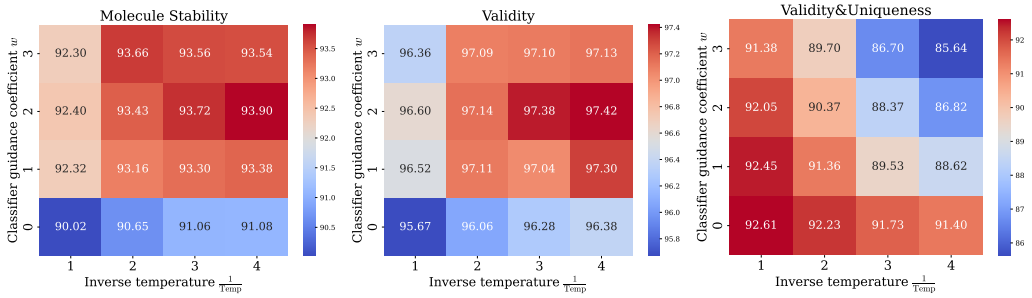
We first evaluate the quality of unconditionally generated molecules from GeoRCG, with the commonly adopted *validity* and *stability* metrics for assessing molecules' quality (Hoogeboom et al., 2022). See Appendix C for detailed descriptions of these metrics.

Main Results We present the main results on the QM9 and DRUG datasets in Table 1. Below, we highlight the key findings: (i) **Improvement over the base model (EDM):** By leveraging geometric representations, GeoRCG significantly outperforms the base model, EDM, on both QM9 and DRUG datasets. Notably, on QM9, it increases stable molecules from 82% to 93.9% and validity from 91.9% to 97.4%, while also improving molecule uniqueness. (ii) **Superior performance compared to advanced methods:** GeoRCG also surpasses included advanced models on the QM9 dataset. On the DRUG dataset, it outperforms models such as EDM-Bridge and GOAT, and gets a high score in validity. Although it falls short of achieving the best performance, we attribute this to the quality of the encoder's representations, as we pre-trained it on the GEOM-DRUG dataset, which may lack diversity and sufficient size. Crucially, many structures in GEOM-DRUG lack the *equilibrium conditions* necessary for pre-training methods that enable effective learning of force fields (Zaidi et al., 2022; Feng et al., 2023). Additionally, GeoRCG adopts EDM (Hoogeboom et al., 2022) as the

base molecule generator, which can be weaker compared to recent advanced models. Nevertheless, further improvement to GeoRCG can be achievabled by incorporating more advanced molecule generators or higher-quality pre-trained encoders.

Balancing Controllability We proceed to investigate the "Balancing Controllablility"" feature of GeoRCG introduced in Section 3.2. To this end, we conducted a grid search by varying both w and $\mathcal{T}$ on QM9 dataset, as depicted in Figure 4. The results indicate a clear trend: increasing w and decreasing $\mathcal{T}$ improve validity and stability at the expense of uniqueness, allowing for fine-grained, flexible control over molecule generation. At its best, this approach achieves a molecule stability of 93.9% and a validity of 97.42%, **approaching the dataset's upper bound**, with a trade-off in lower uniqueness&validity of 86.82%.

Figure 4: Balance controllable (unconditional) generation on QM9 dataset. Increasing w and decreasing $\mathcal{T}$ enhances molecular stability and validity, with the cost of a reduction in uniqueness.



Additional Comparison with 2D&3D methods We further compare with recent 2D&3D methods, including MiDi (Vignac et al., 2023) and LDM-3DG (You et al., 2023), which jointly learn and generate both 2D bond information and 3D geometry. However, these models rely on external tools like RDKit or OpenBabel (O'Boyle et al., 2011) for bond computation at the input stage, which enables them to *implicitly* leverage domain knowledge of such tools. In contrast, purely 3D methods like GeoRCG and those in our primary comparison utilize a coarse look-up table for bond estimation, which, while reflecting 3D learning, results in less accurate bond calculation, potentially biasing the comparison. Therefore, to provide a reference, we also report the performance of our models combined with the same external tools (e.g., OpenBabel) for precise bond calculations in our generated 3D conformations. Furthermore, as GeoRCG essentially captures 3D geometric distributions, we place more emphasis on 3D metrics that *directly evaluate 3D learning capabilities*, including *BondLengthW1* and *BondAngleW1* proposed by Vignac et al. (2023) and detailed in Appendix C. The results in Table 2 demonstrate that GeoRCG not only significantly outperforms MiDi and LDM-3DG on 3D metrics, highlighting the advantages of using a pure 3D model for learning 3D structures, but also further enhances EDM's performance, which has already shown considerable promise in 3D learning.

Table 2: 3D geometry statistics and generated molecule quality on QM9 across different methods. Models marked with * indicate results obtained from our own experiments; see Appendix C for the evaluation guidelines. The stability metrics for EDM are higher than in Table 1 due to using the MiDi codebase for evaluation, which permits more valency for atoms.

Methods \ Metrics	Angles (°) ↓	Bond Length (e-2 Å) ↓	Mol Sta (%) ↑	Atom Sta (%) ↑	Validity (%) ↑	Uniqueness (%) ↑
Data	∼0.1	∼0	98.7	99.8	98.9	99.9
MiDi (uniform)	0.67(0.02)	1.6(0.7)	96.1(0.2)	99.7(0.0)	96.6(0.2)	97.6(0.1)
MiDi (adaptive)	0.62(0.02)	0.3(0.1)	97.5(0.1)	99.8(0.0)	97.9(0.1)	97.6(0.1)
LDM-3DG*	3.56	0.2	94.03	99.38	94.89	97.03
EDM	0.44	0.1	90.7	99.2	91.7	**98.5**
EDM + OBabel	0.44	0.1	97.9	99.8	**99.0**	**98.5**
GeoRCG	0.21(0.04) 52.27%	**0.04(0.0)** 60%	95.82(0.16) 5.6%	99.59(0.02) 0.39%	96.54(0.27) 5.28%	95.74(0.18) 2.8%
GeoRCG + OBabel	**0.20(0.04)** 54.55%	0.07(0.06) 30%	**98.21(0.09)** 0.32%	**99.88(0.00)** 0.08%	**99.0(0.04)** 0.0%	95.74(0.16) 2.8%

4.3 CONDITIONAL MOLECULE GENERATION

We now turn to a more challenging task: generating molecules with a specific property value c from $q(\mathcal{M}|c)$. We strictly follow the evaluation protocol outlined in (Hoogeboom et al., 2022). Speicifically, QM9 is split into two halves, and an EGNN classifier (Satorras et al., 2021) is trained on the first half for evaluating the generated molecules' property, while the generator is trained on the second half[2]. We focus on six properties: polarizability (α), orbital energies ($\varepsilon_{\text{HOMO}}$, $\varepsilon_{\text{LUMO}}$), their gap ($\Delta\varepsilon$), dipole moment ($\mu$), and heat capacity ($C_v$).

Table 3: Conditional molecule generation on QM9. The metric used is the MSE between the target property value and the classifier-predicted value. The gray cells denotes the baseline molecule generator employed in our proposed approach. Models marked with $*$ indicate results obtained from our own experiments; these are provided only as a coarse reference due to potentially differing evaluation criteria, see Appendix C for details.

Methods \ Properties	α	$\Delta\varepsilon$	$\varepsilon_{\text{HOMO}}$	$\varepsilon_{\text{LUMO}}$	μ	C_v
QM9 (lower bound)	0.1	64	39	36	0.043	0.04
Random	9.01	1470	646	1457	1.616	6.857
N_atoms	3.86	866	426	813	1.053	1.971
EDM	2.76	655	356	584	1.111	1.101
GeoLDM	2.37	587	340	522	1.108	1.025
GCDM	1.97	602	344	479	0.844	0.689
EquiFM	2.41	591	337	530	1.106	1.033
GOAT	2.74	605	350	534	1.01	0.883
LDM-3DG*	12.29	1160	583	1093	1.42	5.74
GeoBFN	2.34	577	328	516	0.998	0.949
GeoRCG	**0.89(0.005)** 67.75%	**368.2(4.6)** 43.79%	**220.1(1.0)** 38.17%	**290.8(3.1)** 50.21%	**0.831(0.008)** 25.2%	**0.542(0.004)** 50.77%

The results are presented in Table 3. The first three baselines, as introduced by EDM (Hoogeboom et al., 2022), represent the classifier's inherent bias as the lower bound for performance, the random evaluation result as the upper bound, and the dependency of properties on N. For more details, please refer to Appendix C.

As shown, GeoRCG *nearly doubles the performance* of the best existing models for most properties, with an average 31% improvement over the best ones. This is a task where many recent models struggle to make even modest improvements, as evidenced in the table. Notably, for different properties, we *only re-train the representation generator*, as demonstrated in Section 3.2, significantly saving training time. In Figure 5, we visualize the generated samples, which exhibit minimal property errors and display a clear trend as the target values increase. Additional randomly generated molecules are provided in Appendix F.2.

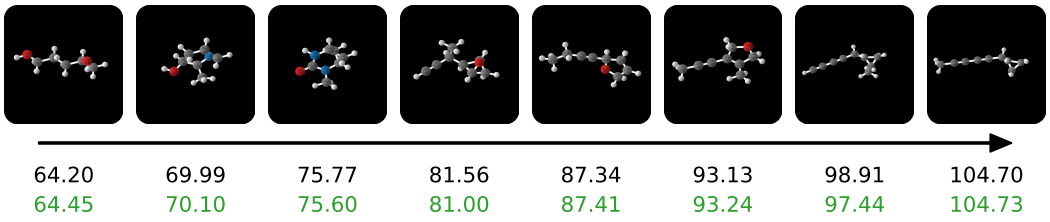


Figure 5: Conditionally generated molecules on property α using GeoRCG. The black number indicates the specified property value (condition), the green number represents the evaluated property value (computed by the classifier) for the generated molecule conformer.

A potential concern is that for a given property value c, $p_\varphi(r|c)$ may produce a representation corresponding to a molecule from the training dataset, allowing the molecule generator to simply recover its full conformation based on that representation. This could lead to small property errors but a lack of novelty. To address this, we conducted a thorough evaluation of the generated molecules across each property, finding that the novelty (the proportion of new molecules not present in the training

[2]Although Figure 3 shows that retraining the molecule generator is unnecessary, for a fair comparison, we still retrain the molecule generator *on the second half of the dataset*, rather than using the one in unconditional molecule generation trained on the entire dataset.

dataset) remains high at approximately 70%, comparable to other methods. Additionally, the conditionally generated molecules demonstrate much higher molecule stability than EDM (Hoogeboom et al., 2022). Further details can be found in Appendix D.

4.4 Faster Generation with Fewer Steps

With geometric representation condition, it is reasonable to expect that fewer discretization steps of the reverse diffusion SDE (Song et al., 2021) would still yield competitive results. Therefore, we reduce the number of diffusion steps and evaluate the model's performance. The results are presented in Table 4. As demonstrated, with the geometric representation condition, GeoRCG consistently outperforms other approaches across all step numbers. Notably, with approximately *100* steps, the performance of our method *nearly converges* to the optimal performance observed with 1000 steps, which already *surpasses all other methods* across all step numbers. This directly reflects the faster generation speed, as the time required for the first-stage representation generation is minimal and can be considered negligible.

Table 4: Quality of generated molecules on QM9 with fewer diffusion steps. The blue cells indicate the highest value among methods with the same number of diffusion steps, while **bold** font emphasizes values that outperform *all* other methods across *all* diffusion steps.

Methods \ Metrics	# Steps	Atom Sta (%) ↑	Mol Sta (%) ↑	Valid (%) ↑	Valid & Unique (%) ↑
Data	-	99	95.2	97.7	97.7
Best baseline in Table 1	1000	99.08	90.87	95.31	92.96
EquiFM	200	98.9(0.1)	88.3(0.3)	94.7(0.4)	93.5(0.3)
GOAT	90	98.4	84.1	90.9	89.99
EDM	50	97.0(0.1)	66.4(0.2)	-	-
EDM-Bridge	50	97.3(0.1)	69.2(0.2)	-	-
GeoBFN	50	98.28(0.1)	85.11(0.5)	92.27(0.4)	90.72(0.3)
GeoRCG	50	98.75(0.05) 1.80%	89.08(0.52) 34.16%	95.05(0.33)	91.32(0.37)
EDM	100	97.3(0.1)	69.8(0.2)	-	-
EDM-Bridge	100	97.9(0.1)	72.3(0.2)	-	-
GeoBFN	100	98.64(0.1)	87.21(0.3)	93.03(0.3)	91.53(0.3)
GeoRCG	100	**99.08(0.03)** 1.83%	**91.85(0.34)** 31.59%	**96.49(0.27)**	92.07(0.35)
EDM	500	98.5(0.1)	81.2(0.1)	-	-
EDM-Bridge	500	98.7(0.1)	83.7(0.1)	-	-
GeoBFN	500	98.78(0.8)	88.42(0.2)	93.35(0.2)	91.78(0.2)
GeoRCG	500	**99.09(0.01)** 0.60%	**91.89(0.24)** 13.17%	**96.57(0.12)**	92.08(0.36)
EDM	1000	98.7	82	91.9	90.7
EDM-Bridge	1000	98.8	84.6	92	90.7
GeoBFN	1000	99.08(0.06)	90.87(0.2)	95.31(0.1)	92.96(0.1)
GeoRCG	1000	**99.12(0.03)** 0.43%	**92.32(0.06)** 12.59%	**96.52(0.2)** 5.03%	92.45(0.2) 1.93%

5 Conclusions

In this work, we present GeoRCG, an effective framework for improving the generation quality of arbitrary molecule generators by incorporating geometric representation conditions. We use EDM (Hoogeboom et al., 2022) as the base molecule generator and demonstrate the effectiveness of our framework through extensive molecular generation experiments. Notably, in conditional generation tasks, GeoRCG achieves a 31% performance boost compared to recent state-of-the-art models. Additionally, the representation guidance enables significantly faster sampling with 10x fewer diffusion steps while maintaining near-optimal performance. Beyond these empirical improvements, we provide theoretical characterizations of general representation-conditioned generative models, which address a key gap in the existing literature (Li et al., 2023).

One limitation of our framework is that its generation quality could depend on the representation quality. For instance, on the GEOM-DRUG dataset, where the encoder was less thoroughly pre-trained, the improvements were less pronounced, and GeoRCG did not surpass SOTA methods. Future work could focus on improving the effectiveness of pre-trained models or exploring enhanced representation regularization techniques. Furthermore, while we employ EDM (Hoogeboom et al., 2022) as the base molecule generator, our framework is general and can be applied to any molecular

generative model. Integrating this framework with more advanced SOTA models offers a promising direction for future exploration and performance enhancement.

REFERENCES

Yasin Abbasi-yadkori, Dávid Pál, and Csaba Szepesvári. Improved algorithms for linear stochastic bandits. In J. Shawe-Taylor, R. Zemel, P. Bartlett, F. Pereira, and K.Q. Weinberger (eds.), *Advances in Neural Information Processing Systems*, volume 24. Curran Associates, Inc., 2011. URL https://proceedings.neurips.cc/paper_files/paper/2011/file/e1d5be1c7f2f456670de3d53c7b54f4a-Paper.pdf.

Brandon Anderson, Truong Son Hy, and Risi Kondor. Cormorant: Covariant molecular neural networks. *Advances in neural information processing systems*, 32, 2019.

Jyoti Aneja, Alex Schwing, Jan Kautz, and Arash Vahdat. A contrastive learning approach for training variational autoencoder priors. *Advances in neural information processing systems*, 34: 480–493, 2021.

Simon Axelrod and Rafael Gomez-Bombarelli. Geom, energy-annotated molecular conformations for property prediction and molecular generation. *Scientific Data*, 9(1):185, 2022.

Tom B Brown. Language models are few-shot learners. *arXiv preprint arXiv:2005.14165*, 2020.

Sitan Chen, Sinho Chewi, Jerry Li, Yuanzhi Li, Adil Salim, and Anru Zhang. Sampling is as easy as learning the score: theory for diffusion models with minimal data assumptions. In *The Eleventh International Conference on Learning Representations*, 2023. URL https://openreview.net/forum?id=zyLVMgsZ0U_.

Xinlei Chen, Saining Xie, and Kaiming He. An empirical study of training self-supervised vision transformers. In *Proceedings of the IEEE/CVF international conference on computer vision*, pp. 9640–9649, 2021.

Bin Dai and David Wipf. Diagnosing and enhancing vae models. *arXiv preprint arXiv:1903.05789*, 2019.

Valentin De Bortoli. Convergence of denoising diffusion models under the manifold hypothesis. *arXiv preprint arXiv:2208.05314*, 2022.

Alexey Dosovitskiy. An image is worth 16x16 words: Transformers for image recognition at scale. *arXiv preprint arXiv:2010.11929*, 2020.

Xiaomin Fang, Lihang Liu, Jieqiong Lei, Donglong He, Shanzhuo Zhang, Jingbo Zhou, Fan Wang, Hua Wu, and Haifeng Wang. Geometry-enhanced molecular representation learning for property prediction. *Nature Machine Intelligence*, 4(2):127–134, 2022.

Shikun Feng, Yuyan Ni, Yanyan Lan, Zhi-Ming Ma, and Wei-Ying Ma. Fractional denoising for 3d molecular pre-training. In *International Conference on Machine Learning*, pp. 9938–9961. PMLR, 2023.

Victor Garcia Satorras, Emiel Hoogeboom, Fabian Fuchs, Ingmar Posner, and Max Welling. E (n) equivariant normalizing flows. *Advances in Neural Information Processing Systems*, 34:4181–4192, 2021.

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

Niklas WA Gebauer, Michael Gastegger, Stefaan SP Hessmann, Klaus-Robert Müller, and Kristof T Schütt. Inverse design of 3d molecular structures with conditional generative neural networks. *Nature communications*, 13(1):973, 2022.

Jordan Graves, Jacob Byerly, Eduardo Priego, Naren Makkapati, S Vince Parish, Brenda Medellin, and Monica Berrondo. A review of deep learning methods for antibodies. *Antibodies*, 9(2):12, 2020.

Jonathan Ho and Tim Salimans. Classifier-free diffusion guidance. *arXiv preprint arXiv:2207.12598*, 2022.

Haokai Hong, Wanyu Lin, and Kay Chen Tan. Fast 3d molecule generation via unified geometric optimal transport. *arXiv preprint arXiv:2405.15252*, 2024.

Emiel Hoogeboom, Vıctor Garcia Satorras, Clément Vignac, and Max Welling. Equivariant diffusion for molecule generation in 3d. In *International conference on machine learning*, pp. 8867–8887. PMLR, 2022.

John B Ingraham, Max Baranov, Zak Costello, Karl W Barber, Wujie Wang, Ahmed Ismail, Vincent Frappier, Dana M Lord, Christopher Ng-Thow-Hing, Erik R Van Vlack, et al. Illuminating protein space with a programmable generative model. *Nature*, 623(7989):1070–1078, 2023.

Ross Irwin, Alessandro Tibo, Jon-Paul Janet, and Simon Olsson. Efficient 3d molecular generation with flow matching and scale optimal transport. *arXiv preprint arXiv:2406.07266*, 2024.

Yunhui Jang, Dongwoo Kim, and Sungsoo Ahn. Hierarchical graph generation with k^2-trees. In *ICML 2023 Workshop on Structured Probabilistic Inference $\{\backslash\&\}$ Generative Modeling*, 2023.

Rui Jiao, Jiaqi Han, Wenbing Huang, Yu Rong, and Yang Liu. Energy-motivated equivariant pretraining for 3d molecular graphs. In *Proceedings of the AAAI Conference on Artificial Intelligence*, volume 37, pp. 8096–8104, 2023.

Rui Jiao, Xiangzhe Kong, Ziyang Yu, Wenbing Huang, and Yang Liu. Equivariant pretrained transformer for unified geometric learning on multi-domain 3d molecules. *arXiv preprint arXiv:2402.12714*, 2024.

Jaehyeong Jo, Dongki Kim, and Sung Ju Hwang. Graph generation with diffusion mixture. *arXiv preprint arXiv:2302.03596*, 2023.

Jacob Devlin Ming-Wei Chang Kenton and Lee Kristina Toutanova. Bert: Pre-training of deep bidirectional transformers for language understanding. In *Proceedings of naacL-HLT*, volume 1, pp. 2, 2019.

Pengyong Li, Jun Wang, Yixuan Qiao, Hao Chen, Yihuan Yu, Xiaojun Yao, Peng Gao, Guotong Xie, and Sen Song. An effective self-supervised framework for learning expressive molecular global representations to drug discovery. *Briefings in Bioinformatics*, 22(6):bbab109, 2021.

Tianhong Li, Dina Katabi, and Kaiming He. Self-conditioned image generation via generating representations. *arXiv preprint arXiv:2312.03701*, 2023.

Zian Li, Xiyuan Wang, Yinan Huang, and Muhan Zhang. Is distance matrix enough for geometric deep learning? *Advances in Neural Information Processing Systems*, 36, 2024a.

Zian Li, Xiyuan Wang, Shijia Kang, and Muhan Zhang. On the completeness of invariant geometric deep learning models. *arXiv preprint arXiv:2402.04836*, 2024b.

Yaron Lipman, Ricky TQ Chen, Heli Ben-Hamu, Maximilian Nickel, and Matt Le. Flow matching for generative modeling. *arXiv preprint arXiv:2210.02747*, 2022.

Shengchao Liu, Hongyu Guo, and Jian Tang. Molecular geometry pretraining with se (3)-invariant denoising distance matching. *arXiv preprint arXiv:2206.13602*, 2022a.

Xingchao Liu, Chengyue Gong, and Qiang Liu. Flow straight and fast: Learning to generate and transfer data with rectified flow. *arXiv preprint arXiv:2209.03003*, 2022b.

Tianze Luo, Zhanfeng Mo, and Sinno Jialin Pan. Fast graph generation via spectral diffusion. *IEEE Transactions on Pattern Analysis and Machine Intelligence*, 2023.

Youzhi Luo and Shuiwang Ji. An autoregressive flow model for 3d molecular geometry generation from scratch. In *International conference on learning representations (ICLR)*, 2022.

Kurt Mislow. *Introduction to stereochemistry*. Courier Corporation, 2012.

Alex Morehead and Jianlin Cheng. Geometry-complete diffusion for 3d molecule generation and optimization. *Communications Chemistry*, 7(1):150, 2024.

Maho Nakata and Tomomi Shimazaki. Pubchemqc project: a large-scale first-principles electronic structure database for data-driven chemistry. *Journal of chemical information and modeling*, 57 (6):1300–1308, 2017.

Yuyan Ni, Shikun Feng, Xin Hong, Yuancheng Sun, Wei-Ying Ma, Zhi-Ming Ma, Qiwei Ye, and Yanyan Lan. Pre-training with fractional denoising to enhance molecular property prediction. *arXiv preprint arXiv:2407.11086*, 2024.

Noel M O'Boyle, Michael Banck, Craig A James, Chris Morley, Tim Vandermeersch, and Geoffrey R Hutchison. Open babel: An open chemical toolbox. *Journal of cheminformatics*, 3:1–14, 2011.

Raghunathan Ramakrishnan, Pavlo O Dral, Matthias Rupp, and O Anatole Von Lilienfeld. Quantum chemistry structures and properties of 134 kilo molecules. *Scientific data*, 1(1):1–7, 2014.

Ali Razavi, Aaron Van den Oord, and Oriol Vinyals. Generating diverse high-fidelity images with vq-vae-2. *Advances in neural information processing systems*, 32, 2019.

Robin Rombach, Andreas Blattmann, Dominik Lorenz, Patrick Esser, and Björn Ommer. High-resolution image synthesis with latent diffusion models. In *Proceedings of the IEEE/CVF conference on computer vision and pattern recognition*, pp. 10684–10695, 2022.

Vıctor Garcia Satorras, Emiel Hoogeboom, and Max Welling. E (n) equivariant graph neural networks. In *International conference on machine learning*, pp. 9323–9332. PMLR, 2021.

Jiaming Song, Chenlin Meng, and Stefano Ermon. Denoising diffusion implicit models. *arXiv preprint arXiv:2010.02502*, 2020.

Yang Song, Jascha Sohl-Dickstein, Diederik P Kingma, Abhishek Kumar, Stefano Ermon, and Ben Poole. Score-based generative modeling through stochastic differential equations. In *International Conference on Learning Representations*, 2021. URL https://openreview.net/forum?id=PxTIG12RRHS.

Yuxuan Song, Jingjing Gong, Yanru Qu, Hao Zhou, Mingyue Zheng, Jingjing Liu, and Wei-Ying Ma. Unified generative modeling of 3d molecules via bayesian flow networks. *arXiv preprint arXiv:2403.15441*, 2024a.

Yuxuan Song, Jingjing Gong, Minkai Xu, Ziyao Cao, Yanyan Lan, Stefano Ermon, Hao Zhou, and Wei-Ying Ma. Equivariant flow matching with hybrid probability transport for 3d molecule generation. *Advances in Neural Information Processing Systems*, 36, 2024b.

Philipp Thölke and Gianni De Fabritiis. Torchmd-net: equivariant transformers for neural network based molecular potentials. *arXiv preprint arXiv:2202.02541*, 2022.

Nathaniel Thomas, Tess Smidt, Steven Kearnes, Lusann Yang, Li Li, Kai Kohlhoff, and Patrick Riley. Tensor field networks: Rotation-and translation-equivariant neural networks for 3d point clouds. *arXiv preprint arXiv:1802.08219*, 2018.

Aaron Van Den Oord, Oriol Vinyals, et al. Neural discrete representation learning. *Advances in neural information processing systems*, 30, 2017.

A Vaswani. Attention is all you need. *Advances in Neural Information Processing Systems*, 2017.

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

Clement Vignac, Igor Krawczuk, Antoine Siraudin, Bohan Wang, Volkan Cevher, and Pascal Frossard. Digress: Discrete denoising diffusion for graph generation. *arXiv preprint arXiv:2209.14734*, 2022.

Clement Vignac, Nagham Osman, Laura Toni, and Pascal Frossard. Midi: Mixed graph and 3d denoising diffusion for molecule generation. In *Joint European Conference on Machine Learning and Knowledge Discovery in Databases*, pp. 560–576. Springer, 2023.

Pascal Vincent. A connection between score matching and denoising autoencoders. *Neural computation*, 23(7):1661–1674, 2011.

Song Wang, Zhen Tan, Xinyu Zhao, Tianlong Chen, Huan Liu, and Jundong Li. Graphrcg: Self-conditioned graph generation via bootstrapped representations. *arXiv preprint arXiv:2403.01071*, 2024.

Lemeng Wu, Chengyue Gong, Xingchao Liu, Mao Ye, and Qiang Liu. Diffusion-based molecule generation with informative prior bridges. *Advances in Neural Information Processing Systems*, 35:36533–36545, 2022.

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

Minkai Xu, Alexander S Powers, Ron O Dror, Stefano Ermon, and Jure Leskovec. Geometric latent diffusion models for 3d molecule generation. In *International Conference on Machine Learning*, pp. 38592–38610. PMLR, 2023.

Yuning You, Ruida Zhou, Jiwoong Park, Haotian Xu, Chao Tian, Zhangyang Wang, and Yang Shen. Latent 3d graph diffusion. In *The Twelfth International Conference on Learning Representations*, 2023.

Hui Yuan, Kaixuan Huang, Chengzhuo Ni, Minshuo Chen, and Mengdi Wang. Reward-directed conditional diffusion: Provable distribution estimation and reward improvement. In *Thirty-seventh Conference on Neural Information Processing Systems*, 2023.

Sheheryar Zaidi, Michael Schaarschmidt, James Martens, Hyunjik Kim, Yee Whye Teh, Alvaro Sanchez-Gonzalez, Peter Battaglia, Razvan Pascanu, and Jonathan Godwin. Pre-training via denoising for molecular property prediction. *arXiv preprint arXiv:2206.00133*, 2022.

Cai Zhou, Xiyuan Wang, and Muhan Zhang. Latent graph diffusion: A unified framework for generation and prediction on graphs. *arXiv preprint arXiv:2402.02518*, 2024.

Gengmo Zhou, Zhifeng Gao, Qiankun Ding, Hang Zheng, Hongteng Xu, Zhewei Wei, Linfeng Zhang, and Guolin Ke. Uni-mol: A universal 3d molecular representation learning framework. 2023.

A More Related Works

Pre-training for Molecular Encoders Learning meaningful molecular representations is crucial for downstream tasks like molecular property prediction (Fang et al., 2022). The paradigm of pre-training on large-scale datasets followed by fine-tuning on smaller datasets has been shown to significantly enhance model performance in both vision and language domains (Kenton & Toutanova, 2019; Brown, 2020; Dosovitskiy, 2020), and building on this success, recent studies have proposed self-supervised pre-training methods for point-cloud-formatted molecules, aiming to achieve similar performance gains (Zhou et al., 2023; Feng et al., 2023; Liu et al., 2022a; Fang et al., 2022; Jiao et al., 2024; Ni et al., 2024). Typical proxy tasks involve masking and recovering atom types, bond lengths, and bond angles (Fang et al., 2022; Zhou et al., 2023). However, since molecules reside in continuous 3D space, a more effective approach is to introduce carefully designed noise to the coordinates and train the model to denoise it. Examples include isotropic Gaussian noise (Zaidi et al., 2022; Zhou et al., 2023), Riemann-Gaussian noise (Jiao et al., 2023), and complex hybrid noise (Ni et al., 2024; Feng et al., 2023; Jiao et al., 2024). Notably, Zaidi et al. (2022) demonstrated that denoising equilibrium structures corresponds to learning the force field, thereby producing representations that are both physically and chemically informative.

B Algorithms

Parallel Training and Sequential Sampling We provide the high-level training and sampling algorithm for GeoRCG in Algorithm 1.

Algorithm 1 Parallel Training and Sequential Sampling for GeoRCG

Input: Molecule dataset $\mathcal{D}_{\text{train}}^{\text{mol}} \subset \bigcup_{N=1}^{+\infty} \left(\mathbb{R}^{N\times 3} \times \mathbb{R}^{N\times d}\right)$, pre-trained geometric encoder E, initial representation generator $p_{\varphi_0}(r)$ and molecule generator $p_{\theta_0}(\mathcal{M}|r)$, classifier-free guidance coefficient w, temperature $\mathcal{T}$.
Output: Trained representation generator $p_{\varphi}(r)$ and molecule generator $p_{\theta}(\mathcal{M}|r)$, molecule samples from $p_{\varphi,\theta}(\mathcal{M})$.

Parallel Training

1: Pre-process to obtain the representation dataset $\mathcal{D}_{\text{train}}^{\text{rep}} = \{\!\{(E(\mathcal{M}), N(\mathcal{M}))|\mathcal{M} \in \mathcal{D}_{\text{train}}^{\text{mol}}\}\!\}$ and the mol-rep dataset $\mathcal{D}_{\text{train}}^{\text{mol-rep}} = \{\!\{(E(\mathcal{M}), \mathcal{M})|\mathcal{M} \in \mathcal{D}_{\text{train}}^{\text{mol}}\}\!\}$.
2: (Parallelly) Train the representation generator with $\mathcal{D}_{\text{train}}^{\text{rep}}$ with loss $\mathcal{L}_{\text{rep}}$ defined in Equation (1); Train the molecule generator with $\mathcal{D}_{\text{train}}^{\text{mol-rep}}$ with loss $\mathcal{L}_{\text{mol}}$ defined in Equation (2), along with representation perturbation introduced in Appendix B.

Sequential Sampling

1: Sample a representation $r_* \sim p_{\varphi}(r)$ with temperature control $\mathcal{T}$.
2: Sample a molecule $\mathcal{M}_* \sim p_{\theta}(\mathcal{M}|r_*)$, conditionally with classifier-free guidance w.

Return: Trained representation generator $p_{\varphi}(r)$, molecule generator $p_{\theta}(\mathcal{M}|r)$, and generated molecule sample.

Representation Perturbation Unlike typical conditional training scenarios, GeoRCG faces a unique challenge: the representations that condition the molecule generator during training may not always coincide with those generated by the representation generator during the sampling stage. This issue is *particularly pronounced* in molecular generation than image case (Li et al., 2023), where pre-trained encoders are typically not trained on that large datasets with advanced regularization techniques like MoCo v3 (Chen et al., 2021). Consequently, the molecule generator is susceptible to *overfitting* to the training representations, as evidenced by our preliminary experiments on QM9 molecule generation shown in Table 5.

We find that applying a simple way by perturbing the geometric representation with Gaussian noise $\sigma_{\text{rep}}\epsilon$ (where $\epsilon \sim \mathcal{N}(0, I)$ and σ_{rep} is a small variance) during the molecule generator's training is particularly effective. Formally, after sampling a data point $(E(\mathcal{M}), \mathcal{M})$ from $\mathcal{D}_{\text{train}}^{\text{mol-rep}}$, we use $(\mathcal{M}, E(\mathcal{M}) + \sigma_{\text{rep}}\epsilon)$ for training. Ablation study in Appendix D show this simple method can effectively prevent overfitting and ensure that performance on novel representations matches those from the training dataset.

Table 5: Quality of molecules generated by GeoRCG trained on the QM9 dataset without using the representation perturbation technique, comparing different representation sources. "Training Dataset" refers to representations sampled from $\mathcal{D}_{\text{train}}^{\text{rep}}$, while "Sampled" refers to representations generated by the trained representation generator $p_\varphi(r)$. See Appendix C for detailed descriptions of the metrics.

Rep source \ Metrics	Mol Sta (%) (↑)	Valid (%) (↑)
Training Dataset	93.20 (0.50)	97.07 (0.32)
Sampled	86.93 (0.50)	89.12 (0.21)

Low-Temperature Sampling We adopt the low-temperature sampling algorithm introduced by Chroma (Ingraham et al., 2023), however, applying it to an MLP-based diffusion model rather than the equivariant diffusion model that processes geometric objects as Chroma.

The objective of low-temperature sampling is to perturb the learned representation distribution $p_\varphi(r)$ by rescaling it with an inverse temperature factor, $\frac{1}{\mathcal{T}}$, where $\mathcal{T}$ is a tunable temperature parameter during sampling, and finally enables sampling from $\mathcal{Z}_\mathcal{T} p_\varphi^{\frac{1}{\mathcal{T}}}$, where $\mathcal{Z}_\mathcal{T}$ is a normalization constant. The method proposed in Chroma (Ingraham et al., 2023) scales the score ϵ_t estimated at each diffusion time step using a time-dependent factor λ_t. The approach is derived from and has theoretical guarantees for simplified toy distributions, and its performance on complex distributions, though lacking strict guarantees, has shown consistent results when combined with annealed Langevin sampling (Song et al., 2021). Here we briefly introduce it for self-containess, and recommend the readers to Ingraham et al. (2023) for detailed derivation and illustration.

Consider the vanilla reverse SDE used in DDPM sampling (VP formulation) (Song et al., 2021):

$$dr = -\frac{1}{2}\beta_t r - \beta_t \nabla_r \log q_t(r) dt + \sqrt{\beta_t} d\bar{\mathbf{w}}, \tag{5}$$

where $\bar{\mathbf{w}}$ is a reverse-time Wiener process, $q_t(r)$ denotes the ground-truth representation distribution at time t, and β_t represents the time-dependent diffusion schedule. To incorporate low-temperature sampling, we utilize the following Hybrid Langevin Reverse-time SDE:

$$dr = -\frac{1}{2}\beta_t r - \left(\lambda_t + \frac{\lambda_0 \psi}{2}\right) \beta_t \nabla_r \log q_t(r) dt + \sqrt{\beta_t (1+\psi)} d\bar{\mathbf{w}}, \tag{6}$$

where λ_t is a time-dependent temperature parameter defined as $\frac{\lambda_0}{\alpha_t^2 + (1-\alpha_t^2)\lambda_0}$, with $\lambda_0 = \frac{1}{\mathcal{T}}$. α_t satisfies $\frac{1}{2}\beta_t = \frac{d \log \alpha_t}{dt}$. The parameter ψ controls the rate of Langevin equilibration per unit time, and as shown in Ingraham et al. (2023), it helps align more effectively with the reweighting objective in complex distributions. In our implementation, we employ the explicit annealed Langevin process (the corrector step from (Song et al., 2021)) to achieve similar results, with the corrector step number fixed to 5.

Classifier-Free Guidance We employ the classifier-free guidance algorithm, as introduced in (Ho & Salimans, 2022), for our molecule generator. Specifically, we introduce a trainable "fake" representation, denoted as l, which serves as the unconditional signal. During the training phase, l is initialized as learnable parameters, and with a probability of p_{fake}, the true representation r is replaced by l. This ensures that the model is capable of generating molecules unconditionally, i.e., $p_\theta(\mathcal{M}|l)$ approximates $q(\mathcal{M})$. During sampling, the final score estimate produced by the molecule generator is adjusted using the formula $(1+w) f_\theta(\mathcal{M}_t, t, r) - w f_\theta(\mathcal{M}_t, t, l)$, allowing flexible control over the strength of the representation guidance.

C Experiment Details

Metrics and Baseline Descriptions We adopt the evaluation metrics, guidelines and baselines commonly used in prior 3D molecular generative models to ensure a fair comparison (Hoogeboom et al., 2022).

- In the unconditional setting, we assess the generated molecules using several key metrics:
  - **Atom Stability**: The proportion of atoms with correct valency.
  - **Molecule Stability**: The proportion of molecules where all atoms within the molecule are stable.
  - **Validity**: The proportion of molecules that can be converted into valid SMILES using RDKit.
  - **Validity & Uniqueness**: The proportion of unique molecules among the valid molecules.

  Following prior work on direct 3D molecular conformations $\mathcal{M} = (\mathbf{x}, \mathbf{h})$ (Hoogeboom et al., 2022), we infer bond information based on atom types and bond lengths using look-up table methods.

  Following the approach of Hoogeboom et al. (2022); Vignac & Frossard (2021), we do not report **Novelty** scores in the main text, since QM9 represents an exhaustive enumeration of molecules satisfying a predefined set of constraints, therefore, "novel" molecule would necessarily violate at least one of these constraints, which indicates that a model fails to fully capture the properties of the dataset. Consistent with Hoogeboom et al. (2022), we observe that the novelty score decreases as stability and validity scores improve, eventually stabilizing at approximately 60%. This is comparable to the final novelty score of EDM, which is around 65% (see Table 8).

  When comparing with 2D & 3D models, we evaluate two 3D metrics introduced by MiDi (Vignac et al., 2023), which directly assess the geometry learning ability:

  - **BondLengthW1**: The weighted 1-Wasserstein distance between the bond-length distributions of the generated molecules and the training dataset, with weights corresponding to different bond types.
  - **BondAngleW1**: The weighted 1-Wasserstein distance between the atom-centered angle distributions of the generated molecules and the training dataset, with weights based on atom types.

  They are formally defined as:

$$\text{BondLengthsW1} = \sum_{y \in \text{bond types}} q^Y(y) \mathcal{W}1(\hat{D}_{\text{dist}}(y), D_{\text{dist}}(y)), \tag{7}$$

  where $q^Y(y)$ is the proportion of bonds of type y in the training set, $\hat{D}_{\text{dist}}(y)$ is the generated distribution of bond lengths for bond type y, and $D_{\text{dist}}(y)$ is the corresponding distribution from the test set. And

$$\text{BondAnglesW1} = \sum_{x \in \text{atom types}} q^X(x) \mathcal{W}1(\hat{D}_{\text{angles}}(x), D_{\text{angles}}(x)), \tag{8}$$

  where $q^X(x)$ denotes the proportion of atoms of type x in the training set, restricted to atoms with two or more neighbors, and $D_{\text{angles}}(x)$ represents the distribution of geometric angles of the form $\angle(r_k - r_i, r_j - r_i)$, where i is an atom of type x, and k and j are neighbors of i.

- In the conditional generation setting, as described in (Hoogeboom et al., 2022), we evaluate our approach on the QM9 dataset across six properties: polarizability α, orbital energies $\varepsilon_{\text{HOMO}}$, $\varepsilon_{\text{LUMO}}$, and their gap $\Delta\varepsilon$, dipole moment μ, and heat capacity C_v. The generative model is trained conditionally on the second half of the QM9 dataset, and an EGNN (Satorras et al., 2021) classifier, trained on the first half, is employed to evaluate the *MAE* property error of the generated samples.

  Three baselines are adopted in Table 3:

  - **QM9 (lower bound)**: The mean error of a classifier trained on the first half of the QM9 dataset and evaluated on the second half. This baseline represents the inherent bias/error of the classifier, setting a lower bound for model performance and reflecting the best possible performance a model can achieve.
  - **Random**: The classifier's performance when evaluated on the second half of QM9 with randomly shuffled molecule property labels. This baseline provides an upper bound, representing the worst achievable performance.
  - **N_atoms**: The performance of a classifier trained exclusively on the number of atoms N and evaluated using only N as input. This baseline captures the intrinsic relationship between molecular properties and the number of atoms, which a generative model must surpass to demonstrate effectiveness.

Model Architectures, Hyperparameters and Training Details

- **Representation Generator.** We use the same architecture for the representation generator as the MLP-based diffusion model proposed in Li et al. (2023). We use 18 blocks of residual MLP layers with 1536 hidden dimensions, 1000 diffusion steps, and a linear noise schedule for β_t. The representation generator is trained for 2000 epochs with a batch size of 128 for both the QM9 and DRUG datasets. Training on QM9 takes approximately 2.5 days on a single Nvidia 4090, while training on DRUG takes around 4 days on a single Nvidia A800. Training time can indeed be further reduced, as the model shows minimal progress after approximately half of the reported time.
- **Molecule Generator.** We adopt EDM (Hoogeboom et al., 2022) as the base molecule generator, using the same EGNN (Satorras et al., 2021) architecture, with the exception of the conditioning module. Specifically, we incorporate a cross-attention (Vaswani, 2017) module for better conditioning on representations, placing it between every block of the EGNN to enhance regularization and increase expressiveness. For the EGNN hyperparameters, we use 9 layers with 256 hidden dimensions for QM9 and 4 layers with 256 hidden dimensions for DRUG. The number of diffusion steps is set to 1000 (except for cases in Table 4 that generate molecules with fewer steps), and we employ the polynomial scheduler for $\alpha_t^{(\mathcal{M})}$. Notably, all model hyperparameters are identical to those in EDM for fair comparison. During training, we use a batch size of 128 and 3000 epochs on QM9, and a batch size of 64 and 20 epochs on DRUG. The representation perturbation values are set to $\sigma_{\text{rep}} = 0.3$ on QM9 and $\sigma_{\text{rep}} = 0.5$ on DRUG. Training takes approximately 6 days on QM9 using a single Nvidia 4090, and around 10 days on DRUG using two Nvidia A800 GPUs.
- **Sampling Details.** During representation sampling, we use the predictor-corrector method from Song et al. (2021) for VP SDE to achieve better equilibrium distributions, which facilitates low-temperature sampling. For molecule sampling, the process remains the same as in EDM, with the addition of classifier-free guidance. For the unconditional generation results in Table 1, we use $(w, \mathcal{T}) = (1.0, 1.0)$ in the QM9 dataset, except for the results in Figure 4, and $(w, \mathcal{T}) = (0.0, 0.5)$ in the GEOM-DRUG dataset (note that $w = 0.0$ indicates that we do not use classifier-free guidance, not that the representation condition itself is not applied). Further tuning of these two hyperparameters may lead to improved results.

Evaluation of LDM-3DG (You et al., 2023) We evaluate the performance of LDM-3DG (You et al., 2023), an Auto-Encoder-based method that also leverages the compactness of the representation space to achieve good performance.

- For the unconditional results in Table 1, we utilize the 3D conformations unconditionally generated by LDM-3DG (You et al., 2023) and compute the bond information using the look-up table method from EDM (Hoogeboom et al., 2022). Notably, although LDM-3DG predicts both the 2D molecular graph and the 3D conformation, *we do not use the bond information it predicts* for the following reasons:
  1. For the calculation of 3D geometry statistics, we observe significant inconsistencies between the generated 2D graphs and 3D geometries (e.g., valid molecules with bond lengths exceeding 100 m), leading to unreliable statistics (e.g., BondLengthW1 exploding to 3900).
  2. For stability and validity metrics, which are fundamentally 2D and computed based on molecular graphs (atoms and bond types), using the generated 2D graph would ignore the contribution of the 3D module, preventing an evaluation of its 3D learning performance.
  3. Most critically, their 2D module is explicitly designed to *filter out* invalid (sub-)molecules during generation using the RDKit method. This means that if invalid molecules or sub-molecules are generated, they are regenerated. This explicit filtering deviates from our standard evaluation criteria and is unsuitable for a fair comparison.
- For the conditional results in Table 3, we first note a potential issue with LDM-3DG (You et al., 2023): The model cannot explicitly specify the node number N during molecule generation, as it uses an auto-regressive 2D generator that automatically stops adding

atoms/motifs when deemed sufficient. However, the evaluation in Table 3 requires specifying both N and property c, following the ground-truth distribution $q(N, c)$ from the training dataset. To ensure fair evaluation, conditions feeding to LDM-3DG must also satisfy this distribution. As the authors claim the model can implicitly learn $q(N)$ and thus $q(N|c)$, we first sample 10,000 values from $q(c)$ and feed them to LDM-3DG, expecting it to infer N from c implicitly as argued, and thus matching the $q(N, c)$ conditions.

D Additional Experiment Results

Ablation Study: Representation Encoders Geometric representations play a pivotal role in GeoRCG. To evaluate the importance of representation quality, we conduct an ablation study comparing the quality of molecule samples generated by GeoRCG trained under different geometric encoder configurations.

We first assess the benefits provided by the pre-training stage. Specifically, we utilize the pre-trained encoder Frad (Feng et al., 2023), trained on the PCQM4Mv2 dataset (Nakata & Shimazaki, 2017) with a hybrid coordinates denoising task (Feng et al., 2023). This approach has been proven to equivalently learn force fields (Feng et al., 2023; Zaidi et al., 2022), and is therefore expected to produce informative representations that capture high-level molecular information. We train GeoRCG using representations from a well-pretrained Frad and a Frad with randomly initialized weights. The molecule generation quality on QM9, as shown in Table 6, clearly underscores the critical role of pre-training on large datasets with advanced techniques in improving representation quality, ultimately enhancing GeoRCG's performance.

Table 6: Quality of molecules generated by GeoRCG with different encoders trained on the QM9 dataset. "Random" indicates that the weights were initialized randomly without any pre-training.

Encoder \ Metrics	Atom Sta (%)	Mol Sta (%)	Valid (%)	Valid & Unique (%)
Random Enc	98.55(0.01)	78.66(0.07)	94.68(0.09)	55.99(0.83)
Pretrained Enc	**99.10(0.02)**	**92.15(0.23)**	**96.48(0.08)**	**92.45(0.21)**

Next, we investigate the impact of different pre-trained encoders, which could vary in model structure and proxy tasks used for pre-training. Specifically, we compare Unimol (Zhou et al., 2023), which employs a message-passing neural network framework incorporating distance features (i.e., DisGNN in (Li et al., 2023; 2024b)) and primarily uses naive coordinates denoising, with Frad (Feng et al., 2023), which adopts TorchMD (Thölke & De Fabritiis, 2022) as the backbone and utilizes carefully designed hybrid-denoising tasks. Both Unimol (Zhou et al., 2023) and Frad (Feng et al., 2023) are pre-trained on the GEOM-DRUG dataset until convergence. We visualize the T-SNE of the representations generated for GEOM-DRUG. As shown in Figure 6, the T-SNE of the Unimol representations exhibits a clearer clustering pattern based on node numbers compared to the Frad representations, which may suggest better representation learning. To further investigate, we utilize both encoders to train GeoRCG and subsequently evaluate the quality of molecule generation. The Frad-based GeoRCG achieves a Validity of $96.9(0.44)$ and Atom Stability of $84.4(0.27)$, while the Unimol-based GeoRCG achieves a Validity of $98.5(0.12)$ and Atom Stability of $84.3(0.12)$: Although the Frad-based GeoRCG produces slightly higher atom stability, its high variance and significantly lower validity suggest inferior performance. These findings, along with our main results, offer insights into the types of representations more effective for guiding molecule generation, suggesting that *sensitivity to molecule size may be a critical factor*.

Ablation Study: Representation Perturbation As discussed in Appendix B, we investigate the effectiveness of the straightforward representation perturbation technique by introducing random noise to perturb the representations during training. Additionally, we apply extra dropout in the conditioning module of our molecule generator to mitigate overfitting on the representation conditions. Ablation experiments presented in Table 7 demonstrate the efficacy of these simple yet impactful methods.

Quality of Conditionally Generated Molecules Detailed molecular metrics for conditionally generated molecules are provided in Table 8. For comparison, we also include the stability metrics

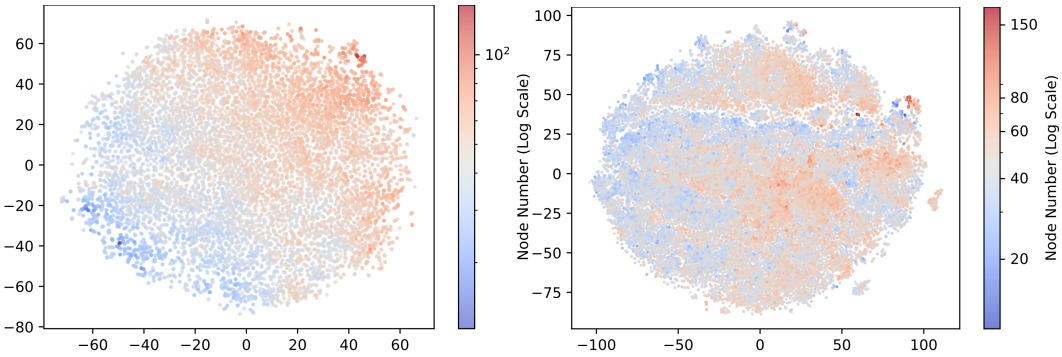


Figure 6: T-SNE visualization of representations produced by the pre-trained encoders for the GEOM-DRUG dataset, colored by node number. The left plot corresponds to Unimol (Zhou et al., 2023), and the right plot corresponds to Frad (Feng et al., 2023).

Table 7: Quality of molecules generated by GeoRCG trained on the QM9 dataset, with and without representation perturbation and representation condition dropout.

Hyper-parameters \ Metrics		Atom Sta (%)	Mol Sta (%)	Valid (%)	Valid & Unique (%)
rep noise ✗	cond. dropout ✗	98.53(0.08)	86.93(0.5)	93.69(0.09)	89.12(0.21)
rep noise ✗	cond. dropout ✓	98.62(0.08)	87.9(0.35)	94.64(0.18)	90.15(0.02)
rep noise ✓	cond. dropout ✗	99.05(0.01)	91.69(0.08)	96.48(0.11)	92.38(0.12)
rep noise ✓	cond. dropout ✓	**99.10(0.02)**	**92.15(0.23)**	**96.48(0.08)**	**92.45(0.21)**

of molecules conditionally generated by EDM, which highlight a notable improvement in stability with GeoRCG. Specifically, EDM's stability scores are: α (80.4%), $\Delta\varepsilon$ (81.73%), $\varepsilon_{\text{HOMO}}$ (82.81%), $\varepsilon_{\text{LUMO}}$ (83.6%), μ (83.3%), and C_v (81.03%).

Table 8: Supplementary evaluation of conditionally generated molecules from GeoRCG. The right side reports metrics for *unconditionally* generated molecules from other methods for reference. Note that conditional models (left) were trained on half of the QM9 dataset, while unconditional models (right) were trained on the full dataset, which may account for slight decreases in stability and validity metrics.

	α	$\Delta\varepsilon$	$\varepsilon_{\text{HOMO}}$	$\varepsilon_{\text{LUMO}}$	μ	C_v	Ours	EDM	GeoLDM	EquiFM
Atom Sta (%)	98.93(0.06)	98.84(0.02)	98.81(0.04)	98.85(0.02)	98.85(0.02)	98.8(0.03)	99.12(0.03)	98.7	98.9(0.1)	98.9(0.1)
Mol Sta (%)	88.89(0.51)	88.83(0.25)	88.50(0.09)	89.04(0.09)	88.66(0.15)	88.7(0.29)	92.32(0.06)	82	89.4(0.5)	88.3(0.3)
Valid (%)	94.85(0.42)	94.83(0.06)	94.84(0.15)	95.01(0.15)	94.82(0.11)	94.95(0.16)	96.52(0.2)	91.9	93.8(0.4)	94.7(0.4)
Valid & Unique (%)	90.31(0.58)	90.42(0.04)	90.44(0.28)	90.73(0.12)	90.52(0.18)	90.65(0.19)	92.45(0.24)	90.7	92.7(0.5)	93.5(0.3)
Valid & Unique & Novelty (%)	71.38(0.46)	72.23(0.58)	71.95(0.30)	71.79(0.41)	72.47(0.59)	72.53(0.72)	61.32(0.77)	65.7	58.1	57.4

E THEORETICAL ANALYSIS

In this section, we provide rigorous theoretical analysis on representation-conditioned diffusion models. Our theory is not limited to molecule generation, and is the first theoretical breakthrough for the RCG framework (Li et al., 2023).

Our analysis is organized as follows. In Appendix E.1, we analyze the generation bound of representation-conditioned diffusion models in *unconditional generation* tasks by showing: (i) the representation can be well generated by the first-stage diffusion model with mild assumptions (Appendix E.1.1); (ii) the second-stage representation-conditioned diffusion model exhibits no higher generalization error than traditional one-stage diffusion model, and can arguably achieve lower error leveraging the informative representations (Appendix E.1.2). Then in Appendix E.2, we analyze *conditional generation* tasks as follows: (iii) under mild assumptions of representations and targets, we provide novel bound for score estimation error (Appendix E.2.1); (iv) generated representations have provable reward improvement towards the target, with the suboptimality composed of offline regression error and diffusion distribution shift (Appendix E.2.2), thus would improve the second stage of conditional generation (Appendix E.2.3).

Notations. In this section, we use SDE and score matching formulations of diffusion models to present our theoretical results, given their equivalence with the DDPM family (Song et al., 2021). We consider the random variable $x \in \mathbb{R}^d$, and use $q(\cdot)$ to denote the ground truth distributions, $p(\cdot)$ to denote the posterior distribution predicted by diffusion models. For instance, $q(x)$ is the ground truth distribution of the underlying data x, while $p_\varphi(r)$ is the predicted distribution of latent representations. We use T to denote the total time of diffusion models, and N to represent the discretization step number. We consider a SDE with continuous time $[0, T]$, as well as its discretized DDPM which has N diffusion steps with step size $h := T/N$. The forward process is denoted as $(x_t)_{t\in[0,T]} \sim q_t$, and the reverse process is denoted as $(\bar{x}_t)_{t\in[0,T]} \sim p_t$. If the reverse process is predicted by the score matching network, we use its parameters as the subscript. Please note that there are two different initialization of the reverse process: the end of forward process q_T and standard Gaussian noise γ^d. We use superscript q_T to differentiate the former from the latter.

E.1 Unconditional Generation

E.1.1 Provable Generation of Representations

Recall the two-stage generation process of representation-conditioned generation: $p(x, r) = p_\theta(x|r)p_\varphi(r)$. To quantitatively evaluate the generation process, we consider two stages separately. In this subsection, we first provide theoretical analysis on the provable generation of representations $p_\varphi(r)$.

Assumption E.1. *(Second moment bound of representations.)*

$$m_r^2 := \mathbb{E}_{q(r)}[||r - \bar{r}||^2] < \infty \tag{9}$$

where $q(r)$ is the ground truth distribution of the representations, and $\bar{r} := \mathbb{E}_{q(r)}[r]$.

Assumption E.2. *(Lipschitz score of representations). For all $t \geq 0$, the score $\nabla \ln q(r_t)$ is L_r-Lipschitz.*

where $q(r_t)$ is the distribution of noisy latent r_t at diffusion step t in the forward process.

Finally, the quality of diffusion models obviously depends on the expressivity of score network φ with prediction $s_\varphi^{(t)}$.

Assumption E.3. *(Score estimation error of representations). For all $t \in [0, T]$,*

$$\mathbb{E}_{q(r_t)}[||s_\varphi^{(t)} - \nabla \ln q(r_t)||^2] \leq \epsilon_{\varphi,\text{score}}^2 \tag{10}$$

These are similar assumptions to the ones in (Chen et al., 2023).

Proposition E.1. *Suppose Assumption E.1, Assumption E.2, Assumption E.3 hold, and the step size $h := T/N$ satisfies $h \preceq 1/L_r$. Then the following holds,*

$$\mathrm{TV}(p_\varphi(r_0), q(r)) \preceq \underbrace{\sqrt{\mathrm{KL}(q(r)||\gamma^{d_r})}\exp(-T)}_{\textit{convergence of forward process}} + \underbrace{(L_r\sqrt{d_r h} + L_r m_r h)\sqrt{T}}_{\textit{discretization error}} + \underbrace{\epsilon_{\varphi,\text{score}}\sqrt{T}}_{\textit{score estimation error}} \tag{11}$$

This is a direct conclusion from (Chen et al., 2023). In typical DDPM implementation, we choose $h = 1$ and thus $T = N$. Remarkably, Proposition E.1 indicates the benefit of generating the representation first: since $d_r \ll d$, the generation quality (measured by the TV distance in Proposition E.1) of the low-dimensional representation can easily outperform directly generating the high-dimensional data points x. The theorem also accounts for applying a lightweight MLP as the denoising network while in the stage of generating the representation.

E.1.2 Provable Second-Stage Generation

Tractable Training Loss. Now we analyze the generation quality of the second-stage diffusion model. Since we sample from $p_\theta(x, r)$, we have representations as conditions even for unconditional generation tasks. To learn the score function conditioning on the representations, consider the

following loss for score matching,

$$\mathcal{L}(\theta) = \int_0^T \lambda(t)\mathbb{E}_{x_t,r}[||s_\theta(x_t, r, t) - \nabla_{x_t} \log q_{t|r}(x_t|r)||^2]dt \tag{12}$$

However, since $q_{t|r}(x_t|r)$ is intractable, we use the following equivalent losses:

$$\mathcal{L}(\theta) = \int_0^T \lambda(t)\mathbb{E}_{x_0,r}\Big[\mathbb{E}_{x_t|x_0}\big[||s_\theta(x_t, r, t) - \nabla_{x_t} \log q_{t|0}(x_t|x_0)||^2|x_0\big]\Big]dt + C \tag{13}$$

Proposition E.2. *(Tractable representation-conditioned score matching loss.)*

$$\mathcal{L}(\theta) := \int_0^T \lambda(t)\mathbb{E}_{x_t,r}[||s_\theta(x_t, r, t) - \nabla_{x_t} \log q_{t|r}(x_t|r)||^2]dt \tag{14}$$

$$= \int_0^T \lambda(t)\mathbb{E}_{x_0,r}\Big[\mathbb{E}_{x_t|x_0}\big[||s_\theta(x_t, r, t) - \nabla_{x_t} \log q_{t|0}(x_t|x_0)||^2|x_0\big]\Big]dt + C \tag{15}$$

Proof. The key is the following important property holds since the gradient is taken w.r.t. x_t only:

$$\nabla_{x_t} \log q_{t|r}(x_t|r) = \nabla_{x_t} \log q_{t,r}(x_t, r) \tag{16}$$

The remaining of the derivation parallels to traditional DDPM. We can replace $\nabla_{x_t} \log q_{t,r}(x_t, r)$ with $\nabla x_t \log q_{t,r|0}(x_t, r|x_0)$:

$$\nabla x_t \log q_{t,r}(x_t, r) = \mathbb{E}_{x_0,r|x_t}\Big[\nabla_{x_t} \log q_{t,r|0}(x_t, r|x_0)\Big|x_t\Big] \tag{17}$$

Thus,

$$\mathbb{E}_r\mathbb{E}_{x_t\sim q(x_t|r)}[||s_\theta(x_t, r, t) - \nabla_{x_t} \log q_{t|r}(x_t|r)||^2] \tag{18}$$

$$=\mathbb{E}_r\mathbb{E}_{x_0\sim q(x_0|r)}\mathbb{E}_{x_t\sim q(x_t|r,x_0)}[||s_\theta(x_t, r, t) - \nabla_{x_t} \log q_{t|r}(x_t|x_0, r)||^2] \tag{19}$$

$$=\mathbb{E}_r\mathbb{E}_{x_0\sim q(x_0|r)}\mathbb{E}_{x_t\sim q(x_t|x_0)}[||s_\theta(x_t, r, t) - \nabla_{x_t} \log q_{t|r}(x_t|x_0)||^2] \tag{20}$$

which is equivalent to our tractable score matching loss. □

Rigorous Error Bound for Second-Stage Generation. Utilizing Proposition E.2, analysis of the second-stage diffusion parallels to the first stage, except that the score network takes additional inputs r.

Assumption E.4. *(Second moment bound of molecule features.)*

$$m_x^2 := \mathbb{E}_{q(x)}[||x - \bar{x}||^2] < \infty \tag{21}$$

where $q(x)$ is the ground truth distribution of the molecule features, and $\bar{x} := \mathbb{E}_{q(x)}x$.

Assumption E.5. *(Lipschitz score of second stage). For all $t \geq 0$, the score $\nabla \ln q(x_t)$ is L_x-Lipschitz.*

where $q(x_t)$ is the distribution of noisy latent x_t at diffusion step t in the forward process.

Finally, we make some assumptions of the score network estimation error.

Assumption E.6. *(Score estimation error of second-stage diffusion). For all $t \in [0, T]$,*

$$\mathbb{E}_{r\sim p_\varphi(r),x_t\sim q_t(x_t)}[||s_\theta(x_t, t, r) - \nabla \ln q_t(x_t)||^2] \leq \epsilon^2_{\theta,\text{score}} \tag{22}$$

This assumption contains the error brought by generating representations, i.e., the TV distance shown in Proposition E.1. Later in Theorem E.1 we explicitly deal with the error brought by representation generation, which results in a more fine-grained error bound.

We now present a key lemma which facilitates analysis and the proof of the central Theorem E.1.

Lemma E.1. *Suppose Assumption E.4, Assumption E.5, Assumption E.6 hold, and the step size $h := T/N$ satisfies $h \preceq 1/L_x$. Suppose we sample $x \sim p_\theta(x|r)$ from Gaussian noise where $r \sim p_\varphi(r)$, and denote the final distribution of x as $p_{\theta,\varphi}(x)$. Then the following holds,*

$$\mathrm{TV}(p_{\theta,\varphi}(x), q(x)) \preceq \underbrace{\sqrt{\mathrm{KL}(q(x)||\gamma^{d+3})}\exp(-T)}_{\text{convergence of forward process}} + \underbrace{(L_x\sqrt{(d+3)h} + L_x m_x h)\sqrt{T}}_{\text{discretization error}} + \underbrace{\epsilon_{\theta,\mathrm{score}}\sqrt{T}}_{\text{score estimation error}} \tag{23}$$

Proof. Recall the notation that $p_{\theta,\varphi}(x) := \int_r p_{0|r}(x_0|r)p_r(r)dr = p_0$ predicted by denoising networks θ, φ starting from Gaussian noise γ^{d+3}. Consider the reverse process $p_0^{q_T}(x_0)$ starting from q_T instead of γ^{d+3},

$$\mathrm{TV}(p_0, q(x)) \le \mathrm{TV}(p_0, p_0^{q_T}) + \mathrm{TV}(p_0^{q_T}, q_0) \tag{24}$$

Using the convergence of the OU process in KL divergence (see (Chen et al., 2023)), the following holds for the first term,

$$\mathrm{TV}(p_0, p_0^{q_T}) \le \mathrm{TV}(\gamma^{d+3}, q_T) \le \sqrt{\mathrm{KL}(q(x)||\gamma^{d+3})}\exp(-T) \tag{25}$$

The second term is caused by score estimation error and discretization error, which can be bounded by

$$\mathrm{TV}(p_0^{q_T}, q_0)^2 \le \mathrm{KL}(q_0||p_0^{q_T}) \preceq (\epsilon_{\theta,\mathrm{score}}^2 + L_x^2(d+3)h + L_x^2 m_x^2 h^2)T \tag{26}$$

We start proving Equation (26) by proving

$$\sum_{k=0}^{N-1} \mathbb{E}_{q_0, r\sim p_\varphi} \int_{kh}^{(k+1)h} ||s_\theta^{(kh)}(x_{kh}, kh, r) - \nabla \ln q_t(x_t)||^2 dt \preceq (\epsilon_{\theta,\mathrm{score}}^2 + L_x^2(d+3)h + L_x^2 m_x^2 h^2)T \tag{27}$$

For $t \in [kh, (k+1)h]$, we decompose

$$\mathbb{E}_{q_0, r\sim p_\varphi}[||s_\theta^{(kh)}(x_{kh}, kh, r) - \nabla \ln q_t(x_t)||^2] \tag{28}$$

$$\preceq \mathbb{E}_{q_0, r\sim p_\varphi}[||s_\theta^{(kh)}(x_{kh}, kh, r) - \nabla q_{kh}(x_{kh})||^2] + \mathbb{E}_{q_0}[||\nabla q_{kh}(x_{kh}) - \nabla q_t(x_{kh})||^2] \tag{29}$$

$$+ \mathbb{E}_{q_0}[||\nabla q_t(x_{kh}) - \nabla q_t(x_t)||^2] \tag{30}$$

$$\preceq \epsilon_{\theta,\mathrm{score}}^2 + \mathbb{E}_{q_0}\left[\left|\left|\nabla \ln \frac{q_{kh}}{q_t}(x_{kh})\right|\right|^2\right] + L^2\mathbb{E}_{q_0}[||x_{kh} - x_t||^2] \tag{31}$$

Note that we omit the term r in expectation of last two terms because they are independent of r.

Utilizing Lemma 16 from (Chen et al., 2023), we bound

$$\left|\left|\nabla \ln \frac{q_{kh}}{q_t}(x_{kh})\right|\right|^2 \preceq L_x^2(d+3)h + L_x^2 h^2||x_{kh}||^2 + (L_x^2+1)h^2||\nabla \ln q_t(x_{kh})||^2 \tag{32}$$

For the last term,

$$||\nabla \ln q_t(x_{kh})||^2 \preceq ||\nabla \ln q_t(x_t)||^2 + ||\nabla \ln q_t(x_{kh}) - \nabla \ln q_t(x_t)||^2 \tag{33}$$

$$\preceq ||\nabla \ln q_t(x_t)||^2 + L^2||x_{kh} - x_t||^2 \tag{34}$$

where the second term is absorbed into the third term of the decomposition Equation (28). Thus,

$$\mathbb{E}_{q_0, r\sim p_\varphi}[||s_\theta^{(kh)}(x_{kh}, kh, r) - \nabla \ln q_t(x_t)||^2] \tag{35}$$

$$\preceq \epsilon_{\theta,\mathrm{score}}^2 + L_x^2(d+3)h + L_x^2 h^2 \mathbb{E}_{q_0}[||x_{kh}||^2] + L_x^2 h^2 \mathbb{E}_{q_0}[||\nabla \ln q_t(x_t)||^2] + L_x^2 \mathbb{E}_{q_0}[||x_{kh} - x_t||^2] \tag{36}$$

$$\preceq \epsilon_{\theta,\mathrm{score}}^2 + L_x^2(d+3)h + L_x^2 h^2(d+3+m_x^2) + L_x^3(d+3)h^2 + L_x^2(m_x^2 h^2 + (d+3)h) \tag{37}$$

$$\preceq \epsilon_{\theta,\mathrm{score}}^2 + L_x^2(d+3)h + L_x^2 h^2 m_x^2 \tag{38}$$

Analogous to (Chen et al., 2023), using properties of Brownian motions and local martingales, we can apply Girsanov's theorem and complete the stochastic integration. Since q_0 is the end of the reverse SDE, by the lower semicontinuity of the KL divergence and the data-processing inequality, we take the limit an+d obtain

$$\mathrm{KL}(q_0||p_0^{q_T}) \preceq (\epsilon_{\theta,\mathrm{score}}^2 + L_x^2(d+3)h + L_x^2 h^2 m_x^2)T \tag{39}$$

We finally conclude with Pinsker's inequality ($\mathrm{TV}^2 \le \mathrm{KL}$). □

This result holds for *general* representation-conditioned diffusion models, and to our best knowledge we are the first to provide theories for representation-conditioned generation, which is a general generation framework suitable for various domains such as images (Li et al., 2023) and graphs.

Lemma E.1 quantitatively characterizes the bound on generalization error in representation-conditioned diffusion. It directly suggests that the error of representation-conditioned diffusion will be no higher than that of its one-stage counterpart. This is because the first two components of the generalization error (i.e., the convergence of the forward process and the discretization error) of the representation-conditioned diffusion model align with those of traditional DDPM, provided that both are parameterized using the same diffusion processes. Furthermore, the third component (score estimation error) can be made identical if we simply set all representation-relevant parameters in s_θ to zero and disregard representation's impact. We therefore have the following conclusion,

Corollary E.1. *Self-representation-conditioned diffusion model can have the same or a lower generation distribution error than one-stage diffusion model.*

We now give a more *fine-grained error bound* analysis of representation-conditioned diffusion, given the relationship between r and x that enables our further qualitative analysis for the argubly better performance.

Assumption E.7. *(representation-conditioned score estimation error of second-stage diffusion). For all $t \in [0, T]$,*

$$\mathbb{E}_{r\sim p_\varphi(r), x_t\sim q_t(x_t|r)}[||s_\theta(x_t, t, r) - \nabla \ln q_t(x_t|r)||^2] \leq \epsilon^2_{\varphi,\theta,\text{cond}} \tag{40}$$

The following main theorem is novel and precise since it (i) deals with the generation error of first-stage representations explicitly; (ii) takes advantages of the conditional distribution $q(x|r)$ in the denoising network.

Theorem E.1. *(Theorem 3.1 in the main text) Suppose Assumption E.4, Assumption E.5, Assumption E.7 hold, and the step size $h := T/N$ satisfies $h \preceq 1/L_x$. Suppose we sample $x \sim p_\theta(x|r)$ from Gaussian noise where $r \sim p_\varphi(r)$, and denote the final distribution of x as $p_{\theta,\varphi}(x)$. Define $p_0^{q_{T|\varphi}}$, which is the end point of the reverse process starting from $q_{T|\varphi}$ instead of Gaussian. Here $q_{T|\varphi}$ is the T-th step in the forward process starting from $q_{0|\varphi} := \frac{1}{A}\int_r q(x_0|r)p_\varphi(r)dr$ where A is the normalization factor. Then the following holds,*

$$\text{TV}(p_{\theta,\varphi}(x), q(x)) \preceq \underbrace{\sqrt{\text{KL}(q_{0|\varphi}||\gamma^{d+3})}\exp(-T)}_{\textit{convergence of forward process}} + \underbrace{(L_x\sqrt{(d+3)h} + L_x m_x h)\sqrt{T}}_{\textit{discretization error}} \tag{41}$$

$$+ \underbrace{\epsilon_{\varphi,\theta,\text{cond}}\sqrt{T}}_{\textit{conditional score estimation error}} + \underbrace{\text{TV}(q_{0|\varphi}, q_0)}_{\textit{representation generation error}} \tag{42}$$

Proof. The proof sketch parallels that of Lemma E.1, except that in the first step we decompose the TV distance as follows,

$$\text{TV}(p_{\theta,\phi}, q(x)) \leq \text{TV}(p_0, p_0^{q_{T|\varphi}}) + \text{TV}(p_0^{q_{T|\varphi}}, q_{0|\varphi}) + \text{TV}(q_{0|\varphi}, q_0) \tag{43}$$

We complete the proof analogously to the proof of Lemma E.1. □

Remarkably, when $q_{0|\varphi}$, i.e., p_φ fully recovers the ground truth marginal distribution of representations $q(r)$, Theorem E.1 has the same format as Lemma E.1 but with $\epsilon_{\varphi,\theta,\text{cond}} < \epsilon_{\theta,\text{score}}$. This is because the former is the score estimation error based on explicit relationship between x and r while the latter learns implicitly. Thus, Theorem E.1 is a much tighter bound for representation-conditioned generation. *To the best of our knowledge, this is the first rigorous theoretical analysis on RCG* (Li et al., 2023). We now provide some qualitative discussions on why representations can arguably lead to better generalization error.

Typically, representations are powerful (and sometimes even complete) as they encode key information about x with potential additional knowledge via pretraining tasks (for example, coordinates denoising for molecules (Zaidi et al., 2022; Feng et al., 2023)). Therefore, it is reasonable to expect that score estimation conditioned on representations can be more accurate (i.e., $\epsilon_{\theta,\text{score}}$ could be significantly smaller than when estimating the score without representation conditioning). If the

representations are complete—where a special case would be $r = x$—this would greatly assist in predicting the noise. The same applies when r can be properly transformed back to x by a neural network. More generally, there are intermediate cases where r reflects partial information about x (e.g., a multiset of atoms and bonds), which would still aid in improving prediction.

To conclude this subsection, we provide a detailed characterization of the generalization error of representation-conditioned diffusion models. It is important to note that some parameters in our assumptions, such as Lipschitz scores and estimation errors, are not constants; they are functions of the SDE total time T and the number of diffusion steps N. Our conclusions also explain why representation-conditioned generation methods remain competitive even when the number of second-stage diffusion steps N is decreased for *faster generation*. This is because the score estimation error can remain small even when the number of diffusion steps is reduced, given the guidance from representations. As a result, reducing N causes a slower increase in $\epsilon_{\varphi,\theta,\mathrm{cond}}(N)$ compared to the score estimation error without representation conditioning, leading to representation-conditioned generative models' strong performance with fewer steps.

E.2 Conditional Generation

In this subsection, we aim to prove that conditional generation using our representation-conditioned generation have probable reward improvement. While we used c to denote conditions in the main text, we use the notation y instead for the targets or "reward" to keep coordinate with existing literature. Denote $q_a := q(\cdot|y = a)$ as the ground truth conditional distribution, and $\hat{p}_a := p(\cdot|y = a)$ the estimated distribution. Suppose the ground truth distribution satisfies $y := f^*(x, r)$ which can be decomposed as

$$f^*(x, r) = g^*(x_\parallel, r_\parallel) + h^*(x_\perp, r_\perp) \tag{44}$$

where we denote $x = x_\parallel$ when $x \sim q(x)$, $r = r_\parallel$ when $r \sim q(r)$, and $f^*(x, r) = g^*(x, r)$ when $x \sim q(x), r \sim q(r)$.

To start with, we assume a linear relationship between r and y, which is reasonable thanks to the powerful pretrained model (which makes the representations helpful in predicting molecule properties and even complete) if some noises are allowed. In detail, the reward is $f^*(x, r) = \hat{w}^\top r + \xi$ and $\xi \sim \mathcal{N}(0, \nu^2)$. In some cases, we may further make Gaussian assumptions on r (WLOG, $r \sim \mathcal{N}(\mu, \Sigma)$) but is not necessary.

E.2.1 Parametric Conditional Score Matching Error

First, we give a detailed form of the representation score estimation error presented in Assumption E.1 under the assumptions above.

Lemma E.2. *For $\delta \geq 0$, with probability $1 - \delta$, the score estimation error $\epsilon_r \simeq \epsilon_{\varphi,\mathrm{score}}$ is bounded by*

$$\frac{1}{T - t_0} \int_{t_0}^{T} \mathbb{E}_{(r_t,y)\sim q_t}[||\nabla \log q_t(r_t|y) - \hat{s}_\varphi(r_t, y, t)||_2^2]\mathrm{d}t \leq \epsilon_r^2 = \mathcal{O}\Big(\frac{1}{t_0}\sqrt{\frac{\mathcal{N}(\mathcal{S}, \frac{1}{n})d_r^2 \log \frac{1}{\delta}}{n}}\Big) \tag{45}$$

where t_0 is the early stopping time of the SDE, n is the number of samples, $\mathcal{S}$ is the parametric function class of denoising network, and $\mathcal{N}(\mathcal{S}, \frac{1}{n})$ is the log covering number of $\mathcal{S}$. When $\mathcal{S}$ is linearly parameterized, $\mathcal{N}(\mathcal{S}, \frac{1}{n}) = \mathcal{O}(d_r^2 \log(\frac{d_r n}{\nu^2}))$.

Proof. This is a direct extension of Lemma C.1 from Yuan et al. (2023). Note that we consider the special case where the low-dimensional subspace is the original space (i.e., $A = I_{d_r}$ and $d = D = d_r$ in their paper), and our noised linear assumption between r and y is identical to their pseudo labeling setting (i.e., $\hat{y} = \hat{w}^\top r + \xi$ where $\xi \sim \mathcal{N}(0, \nu^2)$). We only provide the proof sketch here.

When r follows the Gaussian design, some algebra gives

$$\nabla_r \log q_t(r, y) = \frac{\alpha(t)}{h(t)} B_t\Big(\alpha(t)r + \frac{h(t)}{\nu^2} yw\Big) - \frac{1}{h(t)} r \tag{46}$$

where $\alpha(t) = \exp(-t/2), h(t) = 1 - \exp(-t), B(t) = \Big(\alpha^2(t) I_{d_r} + \frac{h(t)}{\nu^2} ww^\top + h(t)\Sigma^{-1}\Big)^{-1}$. We then bound the estimation error with PAC-learning concentration argument by using Dudley's

entropy integral to bound the Rademacher complexity, and obtain

$$\epsilon_r^2 = \mathcal{O}\Big(\frac{1}{t_0}\sqrt{\frac{\mathcal{N}(\mathcal{S}, \frac{1}{n})d_r^2 \log \frac{1}{\delta}}{n}}\Big) \tag{47}$$

Further, the log covering number of $\mathcal{S}$ under Gaussian design satisfies

$$\mathcal{N}(\mathcal{S}, \frac{1}{n}) \leq d_r^2 \log\Big(1 + \frac{d_r n}{t_0 \lambda_{\min} \nu^2}\Big) \tag{48}$$

where $0 < \lambda_{\min} < 1$ is the smallest eigenvalue of Σ, and typically the early stopping time $t_0 = \mathcal{O}(1)$. In (Yuan et al., 2023) the authors assume $\nu^2 = 1/d_r$ which states that the variance ν^2 of regression residuals ξ reduces when we increase the representation dimensions, which is reasonable. □

Lemma E.2 provides a detailed score estimation error given the linear assumption between r and y, which serves as a special case of $\epsilon^2_{\varphi,\text{score}}$. Substituting it into Proposition E.1, we can obtain a quantitative result of representation generation error.

E.2.2 Reward Improvement via Conditional Generation

Next, we want to obtain the reward guarantees of the generated samples give the condition y. Define the suboptimality of distribution P as

$$\text{SubOpt}(P; y^*) = y^* - \mathbb{E}_{(x,r)\sim P}[f^*(x, r)] \tag{49}$$

where y^* is the target reward value (condition) and f^* is the ground truth reward function. We use the notation $\hat{p}_a := p_\varphi(r|y^* = a)$, then we have the following result for $\text{SubOpt}(\hat{p}_a; y^* = a)$, which can also be viewed as a form of off-policy regret.

Lemma E.3. (Theorem 4.6 in (Yuan et al., 2023).)

$$\text{SubOpt}(\hat{p}_a; y^* = a) \leq \underbrace{\mathbb{E}_{r\sim q_a}\Big[\big|(\hat{w} - w)^\top r\big|\Big]}_{\mathcal{E}_1} + \underbrace{\Big|\mathbb{E}_{r\sim q_a}[g^*(r_\parallel)] - \mathbb{E}_{r\sim \hat{p}_a}[g^*(r_\parallel)]\Big|}_{\mathcal{E}_2} + \underbrace{\mathbb{E}_{r\sim \hat{p}_a}[h^*(r_\perp)]}_{\mathcal{E}_3} \tag{50}$$

Proof. Recall the notation $q_a := q(\cdot|y = a)$, we have

$$\begin{aligned}
&\mathbb{E}_{r\sim \hat{p}_a}[f^*(r)] && (51)\\
\geq &\mathbb{E}_{r\sim q_a}[f^*(r)] - \big|\mathbb{E}_{r\sim \hat{p}_a}[f^*(r)] - \mathbb{E}_{r\sim q_a}[f^*(r)]\big| && (52)\\
\geq &\mathbb{E}_{r\sim q_a}[\hat{f}(r)] - \mathbb{E}_{r\sim q_a}\big[\big|\hat{f}(r) - f^*(r)\big|\big] - \big|\mathbb{E}_{r\sim \hat{p}_a}[f^*(r)] - \mathbb{E}_{r\sim q_a}[f^*(r)]\big| && (53)\\
\geq &\mathbb{E}_{r\sim q_a}[\hat{f}(r)] - \underbrace{\mathbb{E}_{r\sim q_a}\big[\big|\hat{f}(r) - g^*(r)\big|\big]}_{\mathcal{E}_1} - \underbrace{\big|\mathbb{E}_{r\sim q_a}[g^*(r_\parallel)] - \mathbb{E}_{r\sim \hat{p}_a}[g^*(r_\parallel)]\big|}_{\mathcal{E}_2} - \underbrace{\mathbb{E}_{r\sim \hat{p}_a}[h^*(r_\perp)]}_{\mathcal{E}_3} && (54)
\end{aligned}$$

where $\hat{w}$ is the estimated w by Ridge regression, $\mathbb{E}_{r\sim q_a}[\hat{f}(r)] = \mathbb{E}_{a\sim q}[a]$ and $r = r_\perp, f*(r) = g^*(r)$ when $r \sim q_a$. □

Here we give a brief interpretation of the decomposition. $\mathcal{E}_1$ is the prediction and generalization error coming from regression, which is independent from the diffusion error. Both $\mathcal{E}_2$ and $\mathcal{E}_3$ come from the diffusion process, where $\mathcal{E}_2$ reflects the disparity between $\hat{p}_a$ and q_a on the subspace support while $\mathcal{E}_3$ measures the off-subspace error in $\hat{p}_a$. The following results give concrete bounds for the terms in Lemma E.3.

Bounding Regression Error with Offline Bandits. By estimating w with Ridge regression, we have

$$\hat{w} = (R^\top R + \lambda I)^{-1} R^\top (Rw^* + \eta) \tag{55}$$

where $R^\top = (r_1, \ldots, r_n)$ and $\eta = (\xi_1, \ldots, \xi_n)$ where $\xi_i \sim \mathcal{N}(0, \nu^2)$. Define $V_\lambda := R^\top R + \lambda I$, $\hat{\Sigma}_\lambda := \frac{1}{n} V_\lambda$ and $\Sigma_{q_a} := \mathbb{E}_{r\sim q_a} r r^\top$ and take $\lambda = 1$, we have

Proposition E.3. *With high probability,*

$$\mathcal{E}_1 \leq \sqrt{\mathrm{Tr}(\hat{\Sigma}_\lambda^{-1}\Sigma_{q_a})} \cdot \frac{\mathcal{O}(||w^*|| + \nu^2\sqrt{d_r \log n})}{\sqrt{n}} \tag{56}$$

Further when r has Gaussian design $r \sim \mathcal{N}(\mu, \Sigma)$,

$$\mathrm{Tr}(\hat{\Sigma}_\lambda^{-1}\Sigma_{q_a}) \leq \mathcal{O}\big(\frac{a^2}{||w^*||_\Sigma} + d_r\big) \tag{57}$$

when $n = \Omega(\max\{\frac{1}{\lambda_{\min}}, \frac{d_r}{||w^*||_\Sigma^2}\})$.

Proof. First we have

$$\mathcal{E}_1 = \mathbb{E}_{\hat{p}_a}|r^\top(w^* - \hat{w})| \leq \mathbb{E}_{\hat{p}_a}||r||_{V_\lambda^{-1}} \cdot ||w^* - \hat{w}||_{V_\lambda} \tag{58}$$

where

$$\mathbb{E}_{\hat{p}_a}||r||_{V_\lambda^{-1}} \leq \sqrt{\mathbb{E}_{\hat{p}_a} r^\top V_\lambda^{-1} r} = \sqrt{\mathrm{Tr}(V_\lambda^{-1}\mathbb{E}_{\hat{p}_a} r r^\top)} \simeq \sqrt{\mathrm{Tr}(V_\lambda^{-1}\Sigma_{q_a})} \tag{59}$$

Hence we only need to prove

$$||w^* - \hat{w}||_{V_\lambda} \leq \mathcal{O}(||w^*|| + \nu^2\sqrt{d_r \log n}) \tag{60}$$

Using the closed form expression, we have

$$\hat{w} - w^* = V_\lambda^{-1} R^\top \eta - \lambda V_\lambda^{-1} w^* \tag{61}$$

Thus,

$$||w^* - \hat{w}||_{V_\lambda} \leq ||R^\top \eta||_{V_\lambda^{-1}} + \lambda ||w^*||_{V_\lambda^{-1}} \tag{62}$$

where $\lambda||w^*||_{V_\lambda^{-1}} \leq \sqrt{\lambda}||w^*||$, and according to (Abbasi-yadkori et al., 2011),

$$||R^\top \eta||_{V_\lambda^{-1}} = ||R^\top \eta||_{(R^\top R + \lambda I)^{-1}} \leq \mathcal{O}(\nu^2\sqrt{d_r \log n}) \tag{63}$$

with high probability. We hence conclude the first part of proof.

Further, when r has Gaussian design $r \sim \mathcal{N}(\mu, \Sigma)$, according to Lemma C.6 of (Yuan et al., 2023), we can prove the results. The key here is the conditional distribution follows the Gaussian below,

$$P_r\big(r|\hat{f}(r) = a\big) = \mathcal{N}\Big(\mu + \Sigma\hat{w}(\hat{w}^\top\Sigma\hat{w} + \nu^2)(a - \hat{w}^\top\mu), \Sigma - \Sigma\hat{w}(\hat{w}^\top\Sigma\hat{w} + \nu^2)^{-1}\hat{w}^\top\Sigma\Big) \tag{64}$$

Thus

$$\mathrm{trace}(\hat{\Sigma}_\lambda^{-1}\Sigma_{q_a}) = \mathrm{trace}\left(\frac{\Sigma^{1/2}\hat{w}\hat{w}^\top\Sigma\hat{\Sigma}_\lambda^{-1}\Sigma^{1/2}a^2}{(||\hat{w}||_\Sigma^2 + \nu^2)^2}\right) \leq (1 + \frac{1}{\sqrt{\lambda_{\min} n}}) \cdot \mathcal{O}(\frac{a^2}{||\hat{w}||_\Sigma^2} + d_r) \tag{65}$$

Notice that $||\hat{w}||_\Sigma \geq ||w^*||_\Sigma - ||\hat{w} - w^*||_\Sigma$. We have

$$||\hat{w} - w^*||_\Sigma = \mathcal{O}\left(\frac{||w^*|| + \nu^2\sqrt{d_r \log(n)}}{\sqrt{n}}\right) \tag{66}$$

where we can prove $||\hat{w}||_\Sigma \geq \frac{1}{2}||w^*||_\Sigma$ when $n = \Omega(\frac{d_r}{||w^*||_\Sigma^2})$. □

Remarkably, this is a more precise bound improving the results (Lemma C.5 and C.6) in (Yuan et al., 2023), where we make less assumptions on the relationship between y and r, explicitly taking $||w||$ and ν^2 into account.

Bounding Distribution Shift in Diffusion. We define the distribution shift between two arbitrary distributions p_1 and p_2 restricted under function class $\mathcal{L}$ as

$$\mathcal{T}(p_1, p_2; \mathcal{L}) := \sup_{l \in \mathcal{L}} \frac{\mathbb{E}_{x \sim p_1}[l(x)]}{\mathbb{E}_{x \sim p_2}[l(x)]} \tag{67}$$

We have the follow lemma.

Lemma E.4. *Under the assumption that r follows Gaussian design, then*

$$\mathrm{TV}(\hat{p}_a, q_a) = \mathcal{O}\bigg(\sqrt{\frac{\mathcal{T}(q(r, y=a), q_{ry}; \mathcal{S})}{\lambda_{\min}}} \cdot \epsilon_r\bigg) \tag{68}$$

where ϵ_r is defined in Lemma E.2. We can bound $\mathcal{E}_2$ with:

$$\mathcal{E}_2 = \mathcal{O}\Big((\mathrm{TV}(\hat{p}_a, q_a) + t_0)\sqrt{M(a)}\Big) \tag{69}$$

where $M(a) = \mathcal{O}(\frac{a^2}{||w^*||_\Sigma} + d)$. By plugging in $\epsilon_r^2 = \tilde{\mathcal{O}}(\frac{d_r^2}{t_0\sqrt{n}})$, when $t_0 = (d_r^4/n)^{1/6}$, $\mathcal{E}_2$ admits the best trade off with bound

$$\mathcal{E}_2 = \tilde{\mathcal{O}}\bigg(\sqrt{\frac{\mathcal{T}(q(r, y=a), q_{ry}; \mathcal{S})}{\lambda_{\min}}} \cdot (d_r^4/n)^{1/6} a\bigg) \tag{70}$$

Proof. The proof directly follows from Lemma C.4 and Lemma C.7 in (Yuan et al., 2023). However, the conclusion is slightly different as we do not assume a low dimensional subspace of r. □

One can also verify that when r follows Gaussian design, $T(q(r, y=a), q_{ry}; \mathcal{S}) = \mathcal{O}(a^2 \vee d_r)$.

E.2.3 Second Stage of Conditional Generation

Now that we have proved that: (i) the first-stage diffusion model can estimate the score function with provable error bound (Appendix E.2.1); and (ii) the generated representations have provable reward improvement (Appendix E.2.2). We continue to show that the ultimate generated samples also have distribution shift towards the desired target in the following contexts. Particularly, we want to answer the question: why utilizing the conditionally generated representations is enough for the second stage generation?

First, when we use the generated representations as the only condition of the second stage diffusion model, the generation process is identical to the second stage of unconditional generation. Therefore, the results in Appendix E.1.2 can be directly applied to analyze the second stage generation of conditional generation, which states that the generation conditioning on representations has small TV distance error compared with ground truth conditional distribution. Thus, when we have high-quality first-stage generation, the corresponding second stage generation would introduce almost no additional error, which implies provable reward improvements towards the desired target. In addition, the well-pretrained encode ensures that the correspondence between representations and data points is good, which makes it possible to rigorously construct the data points given the representations (a special case would be r is the complete representation of x).

We then partially answer this question from the information theoretic perspectives. We use $H(\cdot)$ to denote the information entropy, and $I(\cdot;\cdot)$ to denote the mutual information between two variables.

- $I(x; r) \geq I(x; y)$. On the one hand, r contains enough information to recover the targets y thanks to the results in Appendix E.2.2, thus we do not explicitly need y for the second stage. One the other hand, benefit from the pretraining task, the representations obviously contains more information in addition to y. This assumption is valid if w^* in previous analysis is sparse (there are components in r independent of y), i.e., $H(r) > H(y)$. Therefore, generating x conditioning on r is much easier than generating conditioning on y (traditional one stage generation), as the score estimation error of the former one would obviously be much smaller than the latter.

- $I(x, r; y) \geq I(x; y)$. Recall Equation (44) which states the target property y depends on both x and r. Hence, r may contain additional information of y obtained from pretrained tasks that is hard to (or cannot) be directly extracted from x - the complex pretrained model assists in extracting relevant information in our two-stage generation, while one-stage generation solely relies on the single denoising model to do so. Therefore, by leveraging representations with provable error bounds, we can better shift the distribution towards the target.

In summary, r is an ideal middle state connecting x and y - it is easy to recover r from y (Appendix E.2.1) and to recover x from y (Appendix E.1.2), and vice versa. In comparison, it is somewhat more difficult to directly recover x from y or predict y from x. Consequently, r may be a better indicator of y compared with x due to the aforementioned reasons.

Indeed, one-stage diffusion models generate x directly from conditions y need to optimize a highly complex score $\nabla_x \log p(x|y)$ where x and y are highly non-linearly mapped. As Theorem E.4 in (Yuan et al., 2023) points out, the nonparametric SubOpt of x generated by deep neural networks is larger than our results in Appendix E.2.2, which further validates the advantage of first generating r that can be well mapped from y.

F Visualization

F.1 Representation Visualization

To illustrate how well $p_\varphi(r)$ fits $q(r)$, we sample from both $q(r)$ (i.e., the representations produced by the pre-trained encoder for the QM9 and GEOM-DRUG datasets) and the trained representation generator $p_\varphi(r)$. We then visualize them in Appendix F.1, with colors indicating whether the samples are from $q(r)$ or $p_\varphi(r)$. We compute the Silhouette Score of the clustering results, scaled by $\mathbf{10^2}$ for clarity. A score close to zero suggests that the two clusters are difficult to distinguish, indicating a good fit between $p_\varphi(r)$ and $q(r)$. Similarly, we provide the visualization of conditionally generated representations in Figure 8

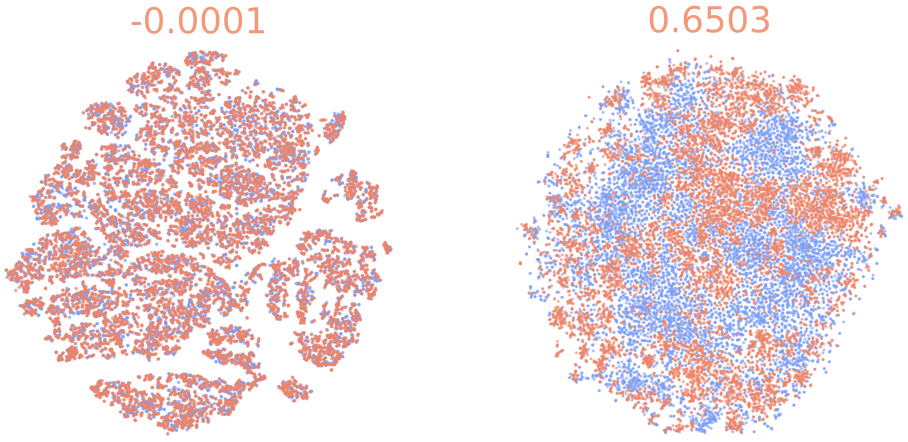


Figure 7: T-SNE visualizations of representations unconditionally generated by the representation generator ($\mathcal{T} = 1.0$) vs. those produced by the pre-trained encoder on the QM9 and DRUG datasets. The Silhouette Score is scaled by $\mathbf{10^2}$ for clarity.

F.2 Visualization of Molecule Samples

In this section, we provide additional random molecule samples to offer deeper insights into the performance of GeoRCG. Figure 9 and Figure 10 show unconditional random samples generated by GeoRCG trained on the QM9 and GEOM-DRUG datasets, respectively. Figure 11 presents random samples conditioned on the α property, along with their corresponding errors.

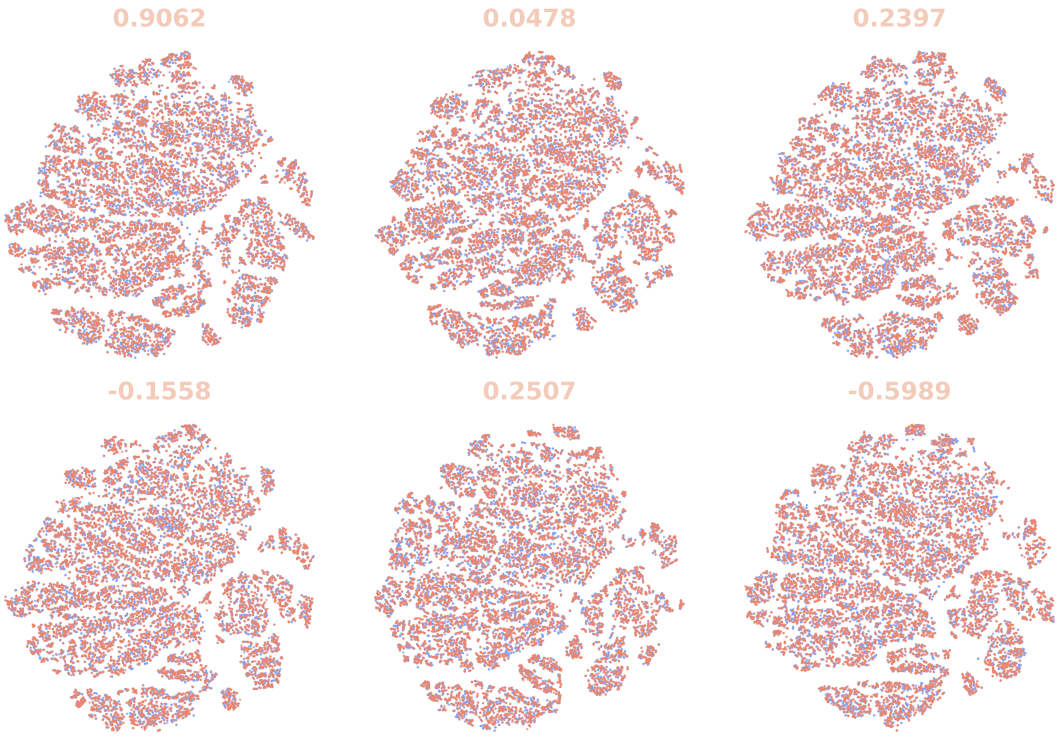


Figure 8: T-SNE visualization of representations conditionally generated by the representation generator vs. those produced by the pre-trained encoder on the QM9 dataset: (a) α, (b) $\Delta\epsilon$, (c) ϵ_{HOMO}, (d) ϵ_{LUMO}, (e) μ, and (f) C_v. The Silhouette Score is scaled by $\mathbf{10^2}$ for clarity.

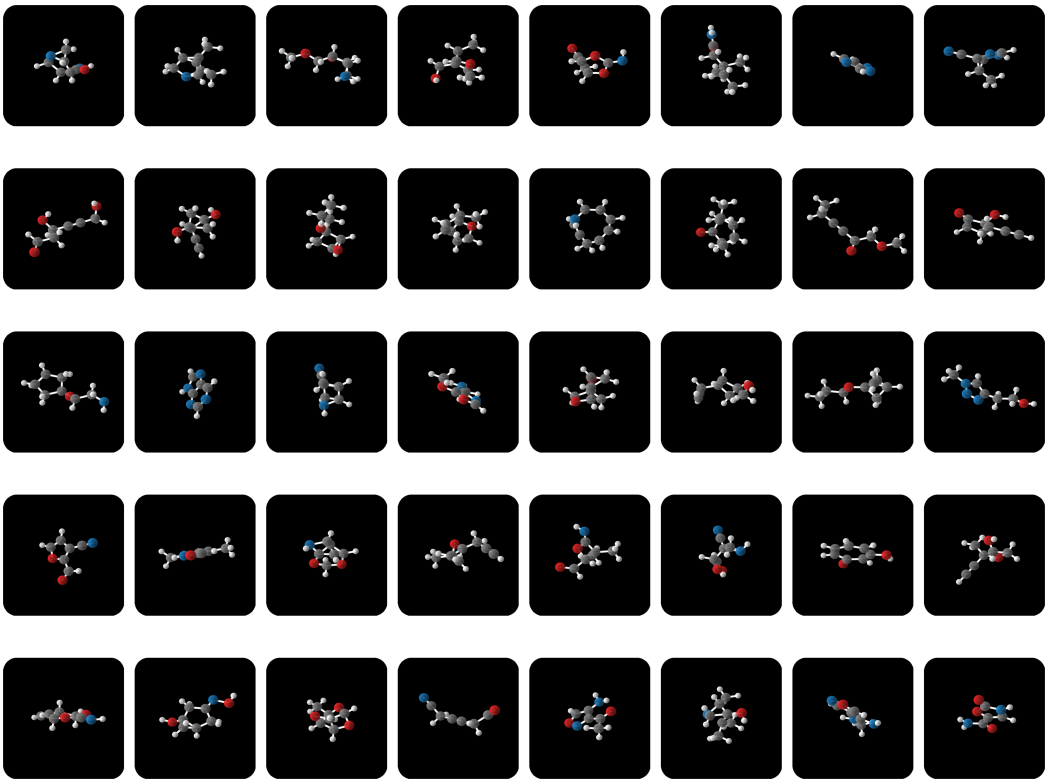


Figure 9: Unconditional random samples from GeoRCG trained on QM9. The number of nodes is randomly sampled from the node distribution $q(N)$.

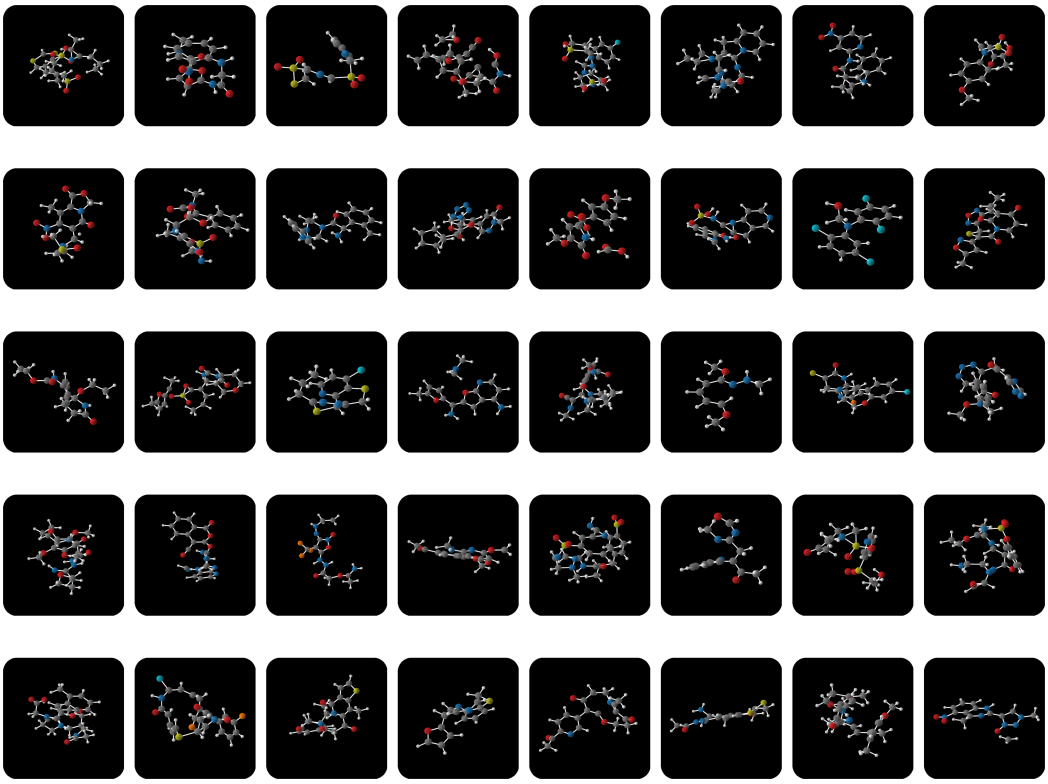


Figure 10: Unconditional random samples from GeoRCG trained on GEOM-DRUG. The number of nodes is randomly sampled from the node distribution $q(N)$.

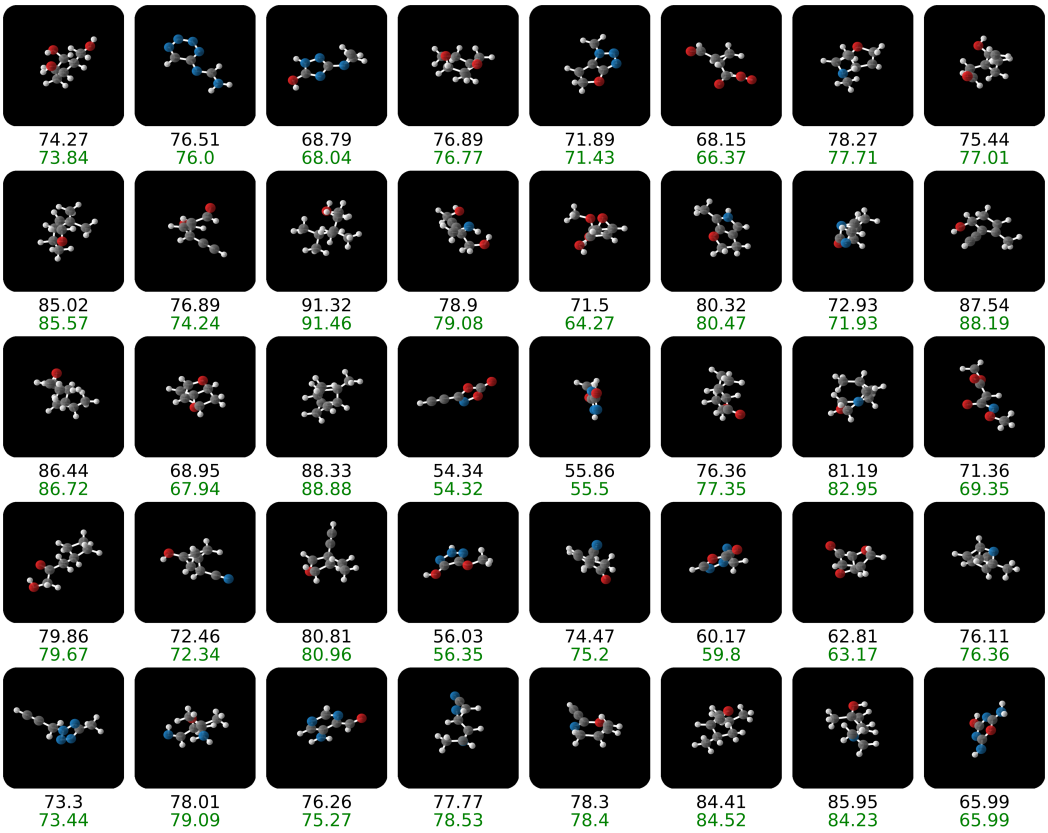


Figure 11: Conditional random samples from GeoRCG trained on QM9 dataset and α property. Black numbers indicate the specified property value condition, while green numbers represent the evaluated property value of the generated samples. The number of nodes and property value conditions are randomly sampled from the joint distribution $q(N, c)$.