

# Can LLMs Generate Diverse Molecules? Towards Alignment with Structural Diversity

Hyosoon Jang<sup>1</sup>, Yunhui Jang<sup>2</sup>, Jaehyung Kim<sup>3</sup>, Sungsoo Ahn<sup>2</sup>

<sup>1</sup>POSTECH <sup>2</sup>KAIST <sup>3</sup>Yonsei University

hsjang1205@postech.ac.kr, jaehyungk@yonsei.ac.kr,  
{yunhuijang, sungsoo.ahn}@kaist.ac.kr

## Abstract

Recent advancements in large language models (LLMs) have demonstrated impressive performance in molecular generation, which offers potential to accelerate drug discovery. However, the current LLMs overlook a critical requirement for drug discovery: proposing a diverse set of molecules. This diversity is essential for improving the chances of finding a viable drug, as it provides alternative molecules that may succeed where others fail in real-world validations. Nevertheless, the LLMs often output structurally similar molecules. While decoding schemes like diverse beam search may enhance textual diversity, this often does not align with molecular structural diversity. In response, we propose a new method for fine-tuning molecular generative LLMs to *autoregressively generate a set of structurally diverse molecules*, where each molecule is generated by conditioning on the previously generated molecules. Our approach consists of two stages: (1) supervised fine-tuning to adapt LLMs to autoregressively generate molecules in a sequence and (2) reinforcement learning to maximize structural diversity within the generated molecules. Our experiments show that the proposed approach enables LLMs to generate diverse molecules better than existing approaches for diverse sequence generation.

## 1 Introduction

Recent advances in large language models (LLMs) have demonstrated the potential to accelerate scientific discovery by leveraging their language processing capabilities. This progress has been particularly impactful for candidate design problems such as drug discovery (Pei et al., 2024), protein design (Zhuo et al., 2024), and material design (Gruber et al., 2024). In particular, with biomolecular datasets and molecular string representations, e.g., SMILES (Weininger, 1988) or SELFIES (Krenn et al., 2020), LLMs have demonstrated impressive

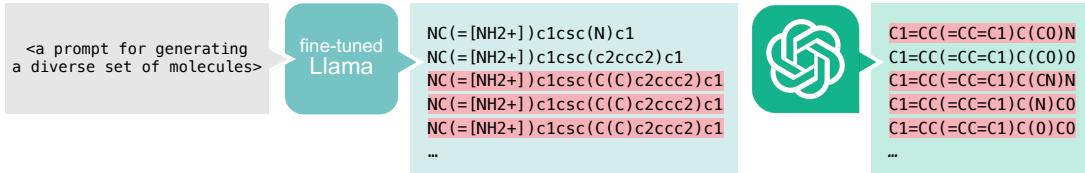
abilities to generate molecules from textual descriptions, e.g., molecular properties (Edwards et al., 2022; Ye et al., 2023; Pei et al., 2024).

However, current LLM-based molecular generation approaches (Edwards et al., 2022; Ye et al., 2023; Pei et al., 2024) often overlook a critical requirement for drug discovery: *proposing a diverse set of molecules*. In computer-aided drug discovery, identifying a single molecule with a desired property does not guarantee success in real-world pipelines that require additional cell-based studies and clinical trials (Vamathevan et al., 2019). Therefore, drug discovery requires a set of structurally diverse molecules. The generation of structurally diverse molecules increases the chances of finding a viable drug candidate (Xie et al., 2023), as different molecules may succeed where others fail. This diversity is essential to enhance the robustness and success of the drug discovery (Krantz, 1998; Hong et al., 2020; Sadybekov and Katritch, 2023).

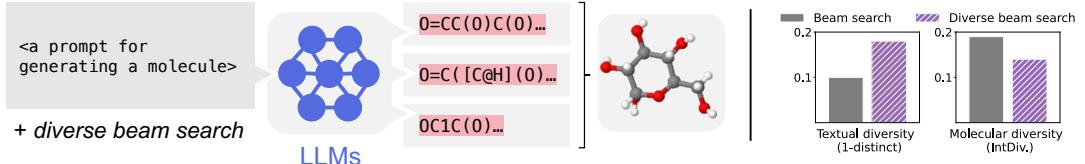
In response, we explore the use of LLMs for diverse molecular generation. We begin by identifying the limitations of recent LLMs (Ye et al., 2023; OpenAI, 2023) and decoding schemes (Vijayakumar et al., 2018; Su et al., 2022) in generating diverse molecules. Then, we present a new method for fine-tuning LLMs to generate diverse molecules. Our approach can be broadly applied to other LLM-based candidate design problems, e.g., computer-aided design (Wu et al., 2023a).

**Existing LLMs have limitations in generating diverse molecules.** To obtain diverse molecules, one may query the recent generalist LLMs, e.g., Llama (Touvron et al., 2023) or ChatGPT (OpenAI, 2023). However, our empirical observation in Figure 1(a) reveals that even the recent models produce structurally identical or highly similar molecules from the given prompt.<sup>1</sup> This observation aligns

<sup>1</sup>ChatGPT-4o (OpenAI, 2023) generates different SMILES strings that map to an identical molecule.



(a) Existing LLMs (Ye et al., 2023; OpenAI, 2023) lack the ability to generate a diverse set of molecules.



(b) (Left) Diverse output sequences (SMILES) induce the same molecular structures. (Right) Improved textual diversity via diverse beam search does not enhance molecular diversity in the experiments (Section 4.1).

Figure 1: **Existing works on LLMs fail to generate diverse molecules.** The existing decoding schemes (Vijayakumar et al., 2018) for diverse sequence generation and LLMs for chemical tasks fail to capture the molecular diversity, and may induce structurally identical molecules.

with previous observations that have shown LLMs may fail to generate diverse outputs (Kirk et al., 2024) for general text-based domains.

**Decoding schemes for diversified generation do not align with molecular diversity.** We also acknowledge the existence of decoding schemes, e.g., diverse beam search (Vijayakumar et al., 2018) or contrastive beam search (Su et al., 2022), which have been proposed to improve the diversity of output sequences generated by LLMs. However, these decoding schemes are limited to improving the textual diversity which often does not correspond to molecular structural diversity, e.g., there exist many SMILES or SELFIES strings that correspond to the same molecule, as illustrated in Figure 1(b).

**Our approach.** We repurpose existing molecular generative LLMs to autoregressively generate a diverse set of molecules from a single prompt. By enabling the LLMs to generate a new molecule conditioned on previously generated molecules, we expect the LLMs to learn to enhance the structural diversity between the generated molecules. To this end, we propose a two-stage approach to fine-tune LLMs: (a) a supervised fine-tuning stage to repurpose LLMs to autoregressively generate a sequence of multiple molecules and (b) a reinforcement learning stage to maximize the molecular structural diversity between the generated molecules.

In the supervised training stage, we train LLMs to autoregressively generate a set of molecules in a sequence. Note that the training data, i.e., a set of molecules, can be collected from LLMs themselves through iterative sampling, and then filtered to enhance the quality, e.g., removing invalid molecules.

However, this stage does not necessarily incorporate molecular diversity, as the training may not involve sufficiently distinct molecules (e.g., limitations in Figure 1(b)). To tackle this, we subsequently apply reinforcement learning with exploration towards discovering diverse molecules.

Next, in the reinforcement learning stage, we train LLMs to maximize the diversity of molecules within a generated sequence. However, for our task, conventional sequence-wise reinforcement learning (Ouyang et al., 2022) suffers from the credit assignment problem (Zhou et al., 2024): the challenges in identifying and promoting the generation of molecules responsible for increasing diversity, among a larger set of molecules in the sequence. To resolve this issue, we solve multi-stage molecule generation problems for a sequence of molecules, where the generation of each molecule aims to maximize the diversity with respect to the previously generated molecules. We train LLMs to maximize the associated rewards using proximal policy optimization (Schulman et al., 2017).

We compare our method with the decoding schemes for diversified generation (Vijayakumar et al., 2016; Su et al., 2022) and other representative LLMs, including chemical-task specialists (Edwards et al., 2022; Christofidellis et al., 2023; Pei et al., 2023, 2024), fine-tuned generalists on chemical domains (Fang et al., 2024; Yu et al., 2024), and the ChatGPT series (OpenAI, 2023, 2024). We observe that (1) our fine-tuning approach enables LLMs to better discover diverse molecules compared to existing decoding schemes and (2) our fine-tuned LLM outperforms other LLMs.

To conclude, our contributions can be summarized as follows:

- We are the first to explore the use of LLMs for generating diverse molecules.
- We first propose a fine-tuning approach for LLMs to generate diverse solutions, which presents a new direction distinct from existing approaches focused on the decoding scheme.
- Experimentally, our method outperforms the baselines in generating diverse molecules.

## 2 Related Work

**Large language models (LLMs) for molecular generation.** Recent advancements in LLMs have shown increasing promise in scientific applications, especially for molecular generation (Edwards et al., 2022; Pei et al., 2023; Fang et al., 2024; Pei et al., 2024). First, Edwards et al. (2022) proposed MolT5, a molecular generative LLM that translates between SMILES (Weininger, 1988) and molecular text descriptions. Next, Text+Chem T5 (Christofellis et al., 2023) and BioT5 (Pei et al., 2023) considered pre-training molecular generative LLMs on datasets that incorporate extensive chemical knowledge, e.g., scientific articles. Additionally, Ye et al. (2023), Fang et al. (2024), (Chen et al., 2023), and Yu et al. (2024) fine-tuned generalist LLMs, e.g., Llama (Touvron et al., 2023), through biological instructions, molecular modifications, and large-scale molecular datasets, respectively.

**Decoding schemes for generating diverse output sequences.** To generate diverse solution candidates from LLMs, existing literature on LLMs has studied improving decoding schemes. To acquire multiple distinct sequences with high likelihoods, one can consider employing beam search, which jointly decodes multiple distinct outputs (Och, 2003). To enhance diversity between sequences, Vijayakumar et al. (2016, 2018) incorporated token-wise differences between generated sequences in the beam search. Furthermore, Su et al. (2022) considered the contrast between the candidate sequences. In addition, Holtzman et al. (2020) proposed nucleus sampling, which enhances random sampling by balancing the quality and the diversity.

**Reinforcement learning (RL) for fine-tuning LLMs.** RL has been effectively applied to fine-tune LLMs, aligning them with desired behaviors expressed through reward signals. One notable example is RL from human feedback to align LLMs

with human preference (Ouyang et al., 2022). In addition, there has been a surge in research on devising RL for LLMs as well, such as addressing multi-turn settings (Shani et al., 2024) and incorporating multiple fine-grained reward signals (Wu et al., 2023b). For molecular generation, Ghugare et al. (2024) proposed RL-based fine-tuning to generate a molecule satisfying target properties. However, to the best of our knowledge, there exist no prior RL-based approaches that aim to increase the diversity of LLM-generated outputs.<sup>2</sup>

## 3 Method

In this section, we present our method for fine-tuning LLMs to generate diverse molecules. Specifically, we consider fine-tuning existing molecular generative LLMs that produce molecular representations such as SMILES or SELFIES. Importantly, our approach is versatile and can be applied to other domains, e.g., protein sequence (Zhuo et al., 2024) or computer-aided design (Wu et al., 2023a).

**Overview.** Our goal is to generate a sequence of structurally diverse molecules from a given prompt by producing them in a single concatenated output. To this end, we fine-tune the LLMs in two stages: (a) a supervised fine-tuning phase that repurposes the LLMs to generate a sequence of molecules rather than a single one, and (b) a reinforcement learning phase aimed at further enhancing the structural diversity among the generated molecules.

**Task details.** In detail, we consider generating molecules from a prompt  $p_{\text{desc}}$ , where the prompt describes a molecular property that the generated molecules should possess. In this setting, we aim to generate diverse molecules that satisfy the given description  $p_{\text{desc}}$ , where the diversity is evaluated using similarity measures between the structural features of the molecules, e.g., the presence of specific atoms, or substructures (Bajusz et al., 2015). We let  $\mathcal{P}$  denote the prompts used for training.

### 3.1 Supervised Fine-tuning

We first describe our supervised fine-tuning process for repurposing the pre-trained LLMs to autoregressively generate multiple molecules in a sequence. This involves collecting a dataset of molecules from a pre-trained LLM  $\pi_{\text{pre}}$ , and then fine-tuning the LLM  $\pi_{\text{SFT}}$  on the collected dataset. We describe the process in Figure 2(a) and Algorithm 1.

<sup>2</sup>More related works, e.g., RL for diverse molecular generation without using LLMs, are described in Appendix A.

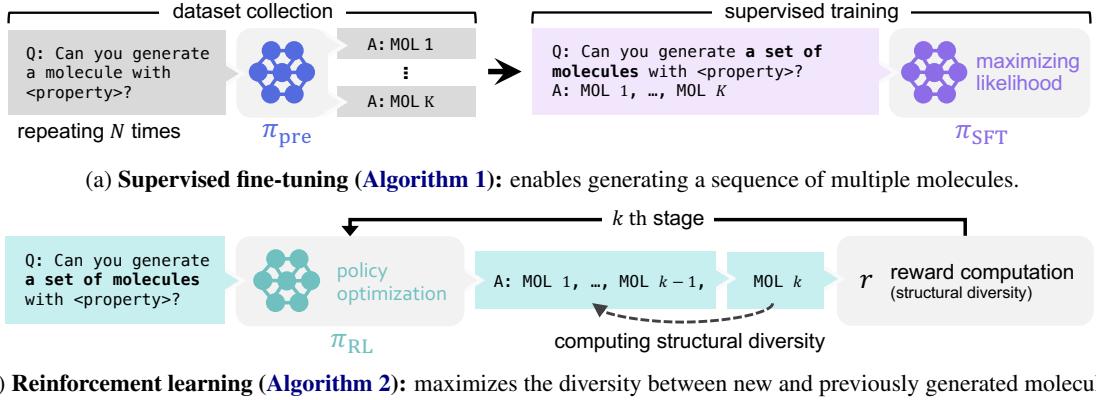


Figure 2: **Illustration of proposed fine-tuning approaches.** We consider two stages for fine-tuning LLMs: a supervised fine-tuning Figure 2(a) and a reinforcement learning Figure 2(b). The prompts are simplified for explanatory purposes, and the actual prompts are provided in Appendix D.

### Algorithm 1 Supervised fine-tuning

- 1: Initialize  $\pi_{\text{SFT}}$  with  $\pi_{\text{pre}}$
- 2: **repeat**
- 3: Get prompt  $p_{\text{desc}} \sim \mathcal{P}$
- 4: Get  $\{m_i\}_{i=1}^T$  from  $\pi_{\text{pre}}(m_i | p_{\text{desc}})$
- 5: Update  $\{m_i\}_{i=1}^K \leftarrow \text{Filter}(\{m_i\}_{i=1}^T)$
- 6: Maximize Equation (1) with  $\{m_i\}_{i=1}^K$
- 7: **until** Converged
- 8: **Output:** fine-tuned  $\pi_{\text{SFT}}$

**Dataset collection.** The supervised training process is conducted with a set of training prompts  $\mathcal{P}$ . Initially, the pre-trained LLM  $\pi_{\text{pre}}$  produces a set of molecules by iterative sampling molecules for a given prompt  $p_{\text{desc}} \in \mathcal{P}$  as follows:

$$m_i \sim \pi_{\text{pre}}(m_i | p_{\text{desc}}) \quad \text{for } i = 1, \dots, T,$$

where  $m_i$  denotes the string representation of the molecule. In practice, we employ beam search to collect the set of molecules  $\{m_i\}_{i=1}^T$ . Then, we filter out the invalid string representations, duplicate molecules, and molecules that do not satisfy the given prompt  $p_{\text{desc}}$ . This results in reducing the set of molecules from  $\{m_i\}_{i=1}^T$  to  $\{m_i\}_{i=1}^K$ . The details are described in Appendix B.

**Supervised training.** Given the filtered set of molecules  $\{m_i\}_{i=1}^K$ , we train the LLM  $\pi_{\text{SFT}}$ , which is initialized with  $\pi_{\text{pre}}$ , to generate them as a single concatenated sequence. We denote this sequence by  $\mathcal{M}_{1:K} = m_1 || \dots || m_K$ , where  $||$  denotes the concatenation of the molecular string representations. Specifically, given a modified prompt  $p_{\text{desc+div}}$  describing the target property with a request for generating diverse molecules, we train

the LLM to maximize the log-likelihood:

$$\log \pi_{\text{SFT}}(\mathcal{M}_{1:K} | p_{\text{desc+div}}). \quad (1)$$

However, the policy  $\pi_{\text{SFT}}$  does not necessarily incorporate a molecular structural diversity, as the set of molecules  $\{m_i\}_{i=1}^K$  collected from  $\pi_{\text{pre}}$  may insufficiently involve diverse molecular structures (e.g., due to limitations in Figure 1(b)). To tackle this, we next introduce an online reinforcement learning stage with exploration towards discovering diverse molecules.

### 3.2 Reinforcement Learning

#### Algorithm 2 Multi-stage RL fine-tuning

- 1: Sample  $p_{\text{desc}} \sim \mathcal{P}$
- 2: Sample  $m_1 \sim \pi_{\text{RL}}(m_1 | p_{\text{desc+div}})$
- 3: Update  $\pi_{\text{RL}}$  with PPO to maximize  $r(m_1)$
- 4: **for**  $k = 2, \dots, K$  **do**
- 5:     Get  $m_k \sim \pi_{\text{RL}}(m_k | \mathcal{M}_{1:k-1}, p_{\text{desc+div}})$
- 6:     Update  $\pi_{\text{RL}}$  with PPO to maximize  $r(m_k)$
- 7: **end for**

We apply reinforcement learning to maximize the diversity of the generated molecules within a sequence. However, when applied to a sequence of molecules  $\mathcal{M}_{1:K}$ , conventional sequence-wise reinforcement learning (Ouyang et al., 2022) suffers from the credit assignment problem (Zhou et al., 2024): the challenge in identifying and promoting the generation of molecules responsible for increasing diversity. To circumvent this, we introduce a molecule-wise reinforcement learning.<sup>3</sup>

<sup>3</sup>We make comparisons between the sequential- and the molecule-wise approaches in Table 5 of Section 4.4.

Table 1: **Comparison with existing decoding schemes.** **NCircles** represents both quality and diversity-related metric. **Accepted & unique** represent quality-related metrics. **IntDiv.** represents an average of pair-wise diversities. Our method (**Div-SFT+RL**) generates more diverse and high-quality molecules compared to the baselines. Notably, our method makes a larger gap over the baselines on **NCircles** related to capturing both quality and diversity.

Dataset	Method	$\text{NCircles}_{h=0.85}$	$\text{NCircles}_{h=0.75}$	Accepted & Unique	IntDiv.
L+M-24	Random	1.079	0.948	1.970	0.109
	Nucleus	1.006	0.918	1.623	0.090
	BS	4.562	2.603	33.406	0.176
	Diverse BS	2.395	1.743	9.915	0.234
	Contrastive BS	4.521	2.594	<b>33.568</b>	0.176
	Div-SFT	5.198	3.205	20.711	0.250
ChEBI-20	<b>Div-SFT+RL</b>	<b>10.51</b>	<b>6.278</b>	31.060	<b>0.287</b>
	Random	1.539	1.383	1.998	0.045
	Nucleus	0.897	0.876	0.972	0.014
	BS	5.576	4.171	11.734	0.194
	Diverse BS	4.413	3.637	5.905	0.140
	Contrastive BS	7.955	5.883	15.215	0.233
	Div-SFT	4.869	3.601	8.178	0.141
	<b>Div-SFT+RL</b>	<b>11.301</b>	<b>8.996</b>	<b>16.271</b>	<b>0.246</b>

Specifically, we consider reinforcement learning on a sequence of molecules  $\mathcal{M}_{1:K}$  as learning in  $K$  individual stages. Each stage corresponds to generating a molecule  $m_k$  conditioned on a sequence of previously generated molecules  $\mathcal{M}_{1:k-1}$ . Then, the LLM  $\pi_{\text{RL}}$  is trained to maximize the return of each stage, defined by the reward of the generated molecule. Here, the reward is the diversity between the previously generated molecules  $\{m_i\}_{i=1}^{k-1}$  and the new molecule  $m_k$ . We also incorporate an auxiliary reward to ensure that the generated molecule satisfies the description  $p_{\text{desc}}$ . We present our approach in Figure 2(b) and Algorithm 2.

**Reward.** The reward evaluates the molecule  $m_k$  with a diversity reward  $r_{\text{div}}(m_k, \{m_i\}_{i=1}^{k-1})$  and a description-matching reward  $r_{\text{match}}(m_k, p_{\text{desc}})$ :

$$r(m_k) = r_{\text{div}}(m_k, \{m_i\}_{i=1}^{k-1}) + r_{\text{match}}(m_k, p_{\text{desc}}),$$

where the diversity reward  $r_{\text{div}}$  evaluates structural differences between the molecule  $m_k$  and the previously generated molecules  $\{m_i\}_{i=1}^{k-1}$  by assessing their true molecular structures. Note that  $r_{\text{div}}(m_1)$  is zero. The description-matching reward  $r_{\text{match}}$  evaluates whether the molecule  $m_k$  satisfies the description  $p_{\text{desc}}$ . In practice, the final action completing the molecule yields the reward. The implementation is described in Section 4.

**Policy optimization.** We optimize the LLM  $\pi_{\text{RL}}$  to maximize the reward using proximal policy optimization (PPO; Schulman et al., 2017). Here,  $\pi_{\text{RL}}$  is initialized with  $\pi_{\text{SFT}}$ . We also combine per-token KL penalty from the supervised fine-tuned model following prior studies (Ouyang et al., 2022).

## 4 Experiment

In this section, we validate our supervised fine-tuning and reinforcement learning methods for diverse molecular generation, coined Div-SFT and Div-SFT+RL, respectively.

### 4.1 Comparison with Decoding Schemes

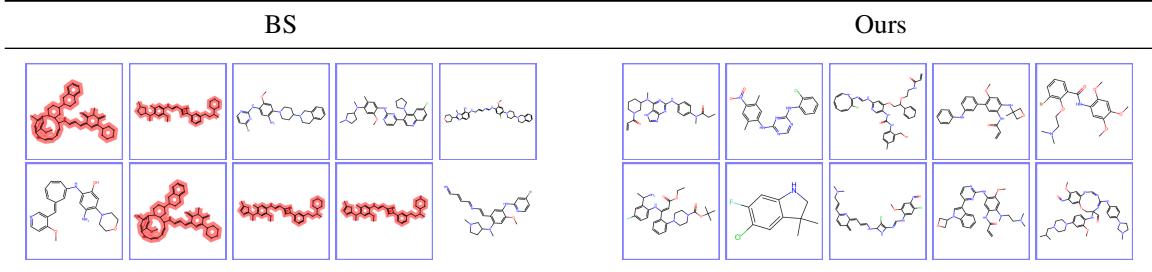
We first show that our fine-tuning method enables molecular generative LLMs to generate diverse molecules better than applying decoding schemes for diverse sequence generation. Here, we implement our method and decoding schemes on a recent description-guided molecular generative LLM: BioT5<sup>+</sup> (Pei et al., 2024).

**Tasks.** We consider diverse molecular generation with two existing datasets: L+M-24 (Edwards et al., 2024) and ChEBI-20 (Edwards et al., 2021). Each data point in these datasets provides a qualitative molecular property description, e.g., “The molecule is an EGFR inhibitor”, and some examples of target molecules that satisfy the description.<sup>4</sup> Both L+M-24 and ChEBI-20 datasets consist of training and test splits where the training splits have been used to pre-train the base LLM. Our fine-tuning method also uses their training splits. We generate 50 molecules for each description in the test splits for evaluation.<sup>5</sup>

<sup>4</sup>In L+M-24 training split, an average of 70 provided molecules corresponds to each molecular property description. We provide the data statistics in Appendix C.

<sup>5</sup>In Section 4.4, we further validate our method by generating 100, 150, and 200 molecules for each description.

Table 2: **Visualization of the generated molecules.** We generate ten molecules from: “The molecule is a egfr inhibitor and modulate, belonging to the cancer treatment class of molecules, and is treatment of disorder”. The blue line indicates the molecule that follows the given description, as evaluated using the external tool (Trott and Olson, 2010). BS generates structurally similar molecules. The details are described in Appendix E.



**Metrics.** We evaluate the structural diversity of the generated molecules that satisfy the given molecular description. However, evaluating whether the generated molecule satisfies some qualitative properties, e.g., a membrane stabilizer, is intractable due to the complexity of the required assessments. Therefore, we assume that the molecule satisfies the description if it shares a certain degree of structures with one of the target molecules provided in the dataset.<sup>6</sup> Here, the specific degree is described in the explanation of the accepted molecules.

To assess the similarity between two molecules, we compute Dice and Tanimoto similarities on their structural features (Bajusz et al., 2015), focusing on the degree of shared substructures and overall structural similarity, respectively. See Appendix A.2 for details. Based on these, we evaluate the following metrics for the generated molecules:

- The number of accepted and unique molecules (Accepted & Unique) counts unique molecules that share substructures with one of the provided examples of target molecules (as much as Dice similarity higher than 0.7).
- The number of circles (NCircles; Xie et al., 2023) considers both quality and diversity. Given the set of accepted molecules,  $\text{NCircles}_h$  computes the size of the largest subset of molecules in which no two molecules are structurally similar to each other (as much as Tanimoto similarity below a threshold  $h$ ).
- Internal diversity (IntDiv.; Polykovskiy et al., 2020) measures the average pair-wise structural distances, based on the complement of Tanimoto similarity, between the accepted molecules.

<sup>6</sup>In Table 2 and Section 4.3, we also evaluate whether the generated molecules actually satisfy the molecular description using an external tool, e.g., a docking tool.

**Implementations.** For supervised fine-tuning, we collect molecules for each description  $p_{\text{desc}}$  in the training splits (a maximum of 100 molecules per description). Then, the collected molecules are filtered to remove invalid, duplicated, and unaccepted molecules. The description-matching reward  $r_{\text{match}}(m_k)$  is defined as the maximum Dice similarity between the molecule  $m_k$  and the provided target molecules. Next, the diversity reward  $r_{\text{div}}(m_k, \{m_i\}_{i=1}^{k-1})$  is defined as the complement of the maximum Tanimoto similarity between the molecule  $m_k$  and the previously generated molecules  $\{m_i\}_{i=1}^{k-1}$ . We describe detailed implementations in Appendix B.

**Baselines.** We compare our method with various decoding schemes, including random sampling, nucleus sampling (Holtzman et al., 2020), and beam search (BS; Rosenberg and Baldwin, 1965). We also consider the variants of BS that promote sequence-level diversity: diverse BS (Vijayakumar et al., 2018) and contrastive BS (Su et al., 2022).

**Results.** In Table 1, we present experimental results on L+M-24 and ChEBI-20 datasets. One can see that applying our fine-tuning approach shows superior performance in discovering diverse accepted molecules, i.e., yields the highest NCircles, compared to applying existing decoding schemes on the molecular generative LLM. We also present qualitative results in Table 2, which evaluate whether the generated molecules actually satisfy the given description using an external tool (Trott and Olson, 2010). One can see that both BS and our method generate molecules that satisfy the given description, but our method generates more diverse molecules compared to BS.

## 4.2 Comparison with Existing LLMs

Additionally, we compare our fine-tuned BioT5<sup>+</sup> in Section 4.1 with baselines that apply existing

Table 3: **Comparison with results obtained by applying existing decoding schemes to each LLM.** The base LLM of our method is BioT5<sup>+</sup>. For each baseline, we report the best result obtained by one of the existing decoding schemes. Our fine-tuned model shows superiority in generating diverse molecules by yielding highest NCircles.

Dataset	Method	NCircles <sub><math>h=0.85</math></sub>	NCircles <sub><math>h=0.75</math></sub>	Accepted & Unique	IntDiv.
L+M-24	MolT5 (Diverse BS)	6.295	4.308	25.841	0.278
	BioT5 <sup>+</sup> (Contrastive BS)	4.562	2.603	<b>33.568</b>	0.176
	Meditron (Diverse BS)	4.693	3.037	27.837	0.263
	<b>Ours</b>	<b>10.511</b>	<b>6.278</b>	31.062	<b>0.287</b>
ChEBI-20	MolT5 (Diverse BS)	1.902	1.509	2.798	0.101
	Text+Chem T5 (Contrastive BS)	6.674	4.408	15.828	0.205
	BioT5 <sup>+</sup> (Contrastive BS)	7.955	5.883	15.215	0.233
	LlaSMol (BS)	7.371	5.492	<b>16.978</b>	0.217
	<b>Ours</b>	<b>11.301</b>	<b>8.966</b>	16.271	<b>0.246</b>

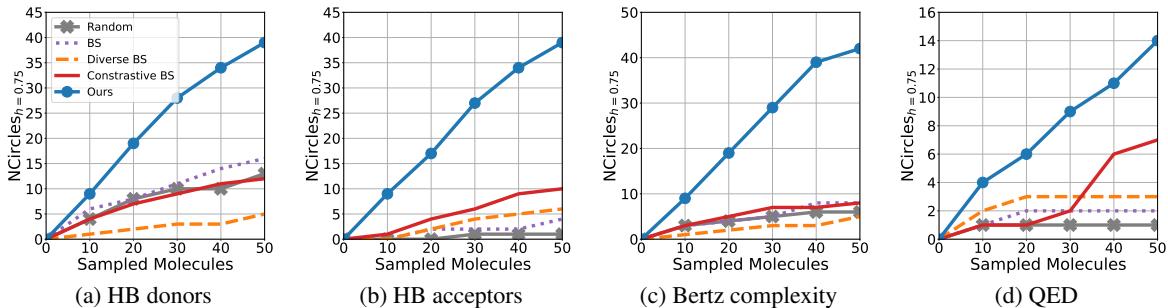


Figure 3: **Experiments with fine-tuned generalist LLMs (DrugAssist; Ye et al., 2023).** Our method consistently improves the performance for generating diverse and high-quality molecules.

decoding schemes to other LLMs, extending the experiment in Section 4.1. The purpose of this experiment is to highlight that other existing LLMs have limitations in diverse molecular generation when relying on existing decoding schemes for diverse sequence generation, compared to our fine-tuned model. The overall metrics and tasks are the same as settings in Section 4.1.

**Baselines.** For the comparison on L+M-24 dataset, we additionally consider LLMs trained on L+M-24 training split: MolT5 (Edwards et al., 2024, 2022) and Meditron (Edwards et al., 2024; Chen et al., 2023). For the comparison on ChEBI-20 dataset, we additionally consider LLMs trained on ChEBI-20 training split: MolT5 (Edwards et al., 2022), Text+Chem T5 (Christofidellis et al., 2023), and LlaSMol (Yu et al., 2024). For each baseline, we report the best results (highest NCircles) obtained using one of existing decoding schemes, i.e., random sampling, BS, diverse BS, and contrastive BS. We describe detailed settings in Appendix D.

**Results.** We present the results in Table 3. One can see that most baselines yield a low NCircles with respect to the number of accepted and unique molecules, indicating a lack of diversity, while our method yields a relatively high NCircles.

### 4.3 Fine-tuning Generalist LLMs

We further validate whether our fine-tuning method improves the generalist LLMs. Here, as a base LLM for implementing our method, we consider DrugAssist (Ye et al., 2023) which is based on the Llama-7B (Touvron et al., 2023). As baselines, we apply existing decoding schemes to DrugAssist.

**Tasks.** We consider molecular descriptions about quantitative molecular properties: hydrogen bond (HB) donors, HB acceptors, Bertz complexity (Bertz, 1981), and quantitative estimate of drug-likeness (QED) (Bickerton et al., 2012). Note that these properties can be evaluated using external tools like RDKit (Landrum et al., 2025).

**Implementations.** We consider fine-tuning the base LLM on prompts about three properties: HB donors, HB acceptors, and Bertz complexity. The prompts about QED are used to assess generalization to unseen properties. The overall implementations follow Section 4.1, but the description-matching reward  $r_{\text{match}}(m_k)$  is designed to yield a positive value when the given property is exactly satisfied. See Appendix B for the detailed settings.

**Results.** We present the results in Figure 3. One can see that our approach improves the performance in generating diverse molecules exactly sat-

Table 4: **Experiments with the large number of samples.** The base LLM is BioT5<sup>+</sup>. Our method discovers more diverse molecules with respect to the (1) the number of generations and (2) time costs.

Method <sub>num. of generations</sub>	BS <sub>300</sub>	BS <sub>400</sub>	BS <sub>500</sub>	Ours <sub>100</sub>	Ours <sub>150</sub>	Ours <sub>200</sub>
NCircles <sub>h=0.85</sub>	13.971	14.553	15.821	14.472	17.215	19.134
Time (sec.)	323	452	585	75	107	146

Table 5: **Comparison with variants of implementations.** Our multi-stage reinforcement learning shows superior performance compared to the single-stage reinforcement learning.

Method	NCircles <sub>h=0.85</sub>	NCircles <sub>h=0.75</sub>	Accepted & Unique	IntDiv.
Div-SFT+RL <sub>single</sub>	5.121	3.846	8.137	0.160
Div-SFT+RL (ours)	11.301	8.966	16.271	0.246

isfying the given molecular descriptions. Furthermore, our approach consistently demonstrates superior performance for the unseen prompt, i.e., the prompt about QED in Figure 3(d).

#### 4.4 Ablation Studies

**Large number of samples vs. performance.** We also analyze how well our method discovers diverse molecules with respect to the number of generations. Here, we extend beyond the settings in Section 4.1. We consider our method to generate 100, 150, and 200 molecules with a single NVIDIA A100 SXM4 40GB GPU. We also consider BS to generate a larger number of molecules with beam sizes of 300, 400, and 500, reaching our maximum computational budget (four NVIDIA A100 SXM4 40GB GPUs). This experiment uses 250 molecular descriptions in the ChEBI-20 test split. We present the results in Table 4. One can see that our method exhibits further performance improvements compared to the baseline, even though the baseline generates a larger number of molecules using our maximum computational budget.

**Time costs vs. performance.** In addition, we also analyze how well our method discovers diverse molecules with respect to the time costs. In Table 4, we present the time costs for each method. One can see that our method discovers more diverse molecules with respect to the time costs.

**Single-stage vs. multi-stage RL.** As mentioned in Section 3.2, we consider a multi-stage setting for generating multiple molecules. However, one may also consider a single-stage setting, where the return of a generated sequence is defined as the sum of the rewards from multiple generated molecules. In Table 5, we compare both approaches. One can see that the multi-stage setting significantly outperforms the single-stage setting. We hypothesize that

this result stems from credit assignment issues in the single-stage setting. Namely, the single-stage setting lacks signals to capture molecule-wise impacts on diversity among a large set of molecules and fails to promote the generation of molecules responsible for increasing diversity.

Table 6: **Coverage of the target molecules in L+M-24 dataset.** The base LLM is BioT5<sup>+</sup>.

Method	Target Coverage
Random	0.137
BS	0.420
Contrastive BS	0.420
Diverse BS	0.343
<b>Ours</b>	<b>0.558</b>

**Coverage of the target molecules.** We further validate how our method benefits capturing the wide range of target molecular space. Here, we compute how many target molecules in the L+M-24 dataset (an average of 17 molecules per description) can be captured within the generation space. The target molecule is considered captured if it is significantly similar to one of the generated molecules (Dice similarity  $> 0.95$ ). In Table 6, we present the average ratio of captured target molecules. One can see that our method yields the highest score.

## 5 Conclusion

In this paper, we identify the limitations of large language models (LLMs) for generating diverse molecules. In response, we present a new fine-tuning approach to adapt existing LLMs to generate diverse molecules. Experiments show that our approach enables LLMs to better discover diverse molecules compared to the existing approaches. This success highlights the potential of our method to advance LLM-driven drug discovery.

## 6 Limitations

Our method may require autoregressively generating a very long sequence to induce a huge set of molecules. In response, an interesting avenue for future work is to reduce the space and time complexity in generating the sequence of molecules, for example, introducing continuous tokens (Hao et al., 2025) that encode the set of previously generated molecules (Zaheer et al., 2017). Next, we validated whether the LLM-generated molecules satisfy the given molecular descriptions using computational approaches. However, this does not imply that the generated molecules will necessarily satisfy the given molecular descriptions in tricky real-world scenarios. To ensure the reliability of our method, the generated molecules should be validated through the real-world experiments. Additionally, although the molecules generated by each method are valid in computational terms, they may not be synthesizable in practice. Lastly, due to the limited computational budgets, we conducted experiments only on models with up to 7B parameters. The generalizability of our method to larger models (e.g., 70B) remains unexplored and is left for future work.

## 7 Ethical Considerations

Our fine-tuning method enables LLMs to generate diverse molecules from textual descriptions of molecular properties. However, these advancements also introduce potential risks, such as the generation of harmful drugs and the misuse of synthesized molecules.

## References

- Dávid Bajusz, Anita Rácz, and Károly Héberger. 2015. Why is tanimoto index an appropriate choice for fingerprint-based similarity calculations? *Journal of cheminformatics*, 7(1):1–13.
- Steven H Bertz. 1981. The first general index of molecular complexity. *Journal of the American Chemical Society*, 103(12):3599–3601.
- G Richard Bickerton, Gaia V Paolini, Jérémie Besnard, Sorel Muresan, and Andrew L Hopkins. 2012. Quantifying the chemical beauty of drugs. *Nature chemistry*, 4(2):90–98.
- Thomas Blaschke, Ola Engkvist, Jürgen Bajorath, and Hongming Chen. 2020. Memory-assisted reinforcement learning for diverse molecular de novo design. *Journal of cheminformatics*, 12(1):68.
- Zeming Chen, Alejandro Hernández Cano, Angelika Romanou, Antoine Bonnet, Kyle Matoba, Francesco Salvi, Matteo Pagliardini, Simin Fan, Andreas Köpf, Amirkeivan Mohtashami, et al. 2023. Meditron-70b: Scaling medical pretraining for large language models. *arXiv preprint arXiv:2311.16079*.
- Dimitrios Christofidellis, Giorgio Giannone, Jannis Born, Ole Winther, Teodoro Laino, and Matteo Manica. 2023. Unifying molecular and textual representations via multi-task language modelling. In *Proceedings of the 40th International Conference on Machine Learning*, volume 202 of *Proceedings of Machine Learning Research*, pages 6140–6157. PMLR.
- Carl Edwards, Tuan Lai, Kevin Ros, Garrett Honke, Kyunghyun Cho, and Heng Ji. 2022. Translation between molecules and natural language. In *Proceedings of the 2022 Conference on Empirical Methods in Natural Language Processing*, pages 375–413, Abu Dhabi, United Arab Emirates. Association for Computational Linguistics.
- Carl Edwards, Qingyun Wang, Lawrence Zhao, and Heng Ji. 2024. L+M-24: Building a dataset for Language+Molecules @ ACL 2024. In *Proceedings of the 1st Workshop on Language + Molecules (L+M 2024)*, pages 1–9, Bangkok, Thailand. Association for Computational Linguistics.
- Carl Edwards, ChengXiang Zhai, and Heng Ji. 2021. Text2Mol: Cross-modal molecule retrieval with natural language queries. In *Proceedings of the 2021 Conference on Empirical Methods in Natural Language Processing*, pages 595–607, Online and Punta Cana, Dominican Republic. Association for Computational Linguistics.
- Yin Fang, Xiaozhuan Liang, Ningyu Zhang, Kangwei Liu, Rui Huang, Zhuo Chen, Xiaohui Fan, and Hua-jun Chen. 2024. Mol-instructions: A large-scale biomolecular instruction dataset for large language models. In *The Twelfth International Conference on Learning Representations*.
- Miguel García-Ortegón, Gregor NC Simm, Austin J Tripp, José Miguel Hernández-Lobato, Andreas Bender, and Sergio Bacallado. 2022. Dockstring: easy molecular docking yields better benchmarks for ligand design. *Journal of chemical information and modeling*, 62(15):3486–3502.
- Raj Ghugare, Santiago Miret, Adriana Hugessen, Mariano Phielipp, and Glen Berseth. 2024. Searching for high-value molecules using reinforcement learning and transformers. In *The Twelfth International Conference on Learning Representations*.
- Nate Gruver, Anuroop Sriram, Andrea Madotto, Andrew Gordon Wilson, C. Lawrence Zitnick, and Zachary Ward Ulissi. 2024. Fine-tuned language models generate stable inorganic materials as text. In *The Twelfth International Conference on Learning Representations*.

- Shibo Hao, Sainbayar Sukhbaatar, DiJia Su, Xian Li, Zhiting Hu, Jason E Weston, and Yuandong Tian. 2025. [Training large language model to reason in a continuous latent space](#).
- Jiazhen He, Alessandro Tibo, Jon Paul Janet, Eva Nittinger, Christian Tyrchan, Werngard Czechitzky, and Ola Engkvist. 2024. Evaluation of reinforcement learning in transformer-based molecular design. *Journal of Cheminformatics*, 16(1):95.
- Ari Holtzman, Jan Buys, Li Du, Maxwell Forbes, and Yejin Choi. 2020. [The curious case of neural text degeneration](#). In *The Tenth International Conference on Learning Representations*.
- Benke Hong, Tuoping Luo, and Xiaoguang Lei. 2020. Late-stage diversification of natural products. *ACS central science*, 6(5):622–635.
- Edward J Hu, yelong shen, Phillip Wallis, Zeyuan Allen-Zhu, Yuanzhi Li, Shean Wang, Lu Wang, and Weizhu Chen. 2022. [LoRA: Low-rank adaptation of large language models](#). In *The Tenth International Conference on Learning Representations*.
- Xiuyuan Hu, Guoqing Liu, Yang Zhao, and Hao Zhang. 2024. De novo drug design using reinforcement learning with multiple gpt agents. *Advances in Neural Information Processing Systems*, 36.
- Robert Kirk, Ishita Mediratta, Christoforos Nalmpantis, Jelena Luketina, Eric Hambro, Edward Grefenstette, and Roberta Raileanu. 2024. Understanding the effects of rlhf on LLM generalisation and diversity. In *The Twelfth International Conference on Learning Representations*.
- Allen Krantz. 1998. Diversification of the drug discovery process. *Nature biotechnology*, 16(13):1294–1294.
- Mario Krenn, Florian Häse, AkshatKumar Nigam, Pascal Friederich, and Alan Aspuru-Guzik. 2020. Self-referencing embedded strings (selfies): A 100% robust molecular string representation. *Machine Learning: Science and Technology*, 1(4):045024.
- Greg Landrum. 2016. Rdkit: Open-source cheminformatics software. <http://www.rdkit.org>.
- Greg Landrum, Paolo Tosco, Brian Kelley, Ricardo Rodriguez, David Cosgrove, Riccardo Vianello, Sriniker, Peter Gedeck, Gareth Jones, Nadine Schneider, Eisuke Kawashima, Dan Nealschneider, Andrew Dalke, Matt Swain, Brian Cole, Samo Turk, Aleksandr Savelev, Tadhurst-Cdd, Alain Vaucher, and Jeremy Monat. 2025. rdkit/rdkit: 2024\_09\_5 (q3 2024) release (release\_2024\_09\_5). <https://doi.org/10.5281/zenodo.14779836>.
- Franz Josef Och. 2003. [Minimum error rate training in statistical machine translation](#). In *Proceedings of the 41st Annual Meeting of the Association for Computational Linguistics*, pages 160–167, Sapporo, Japan. Association for Computational Linguistics.
- OpenAI. 2023. Chatgpt: Openai language model. <https://openai.com/chatgpt>. Accessed: 2023-09-23.
- OpenAI. 2024. [Introducing openai o1 preview](#). Accessed: 2024-09-17.
- Long Ouyang, Jeffrey Wu, Xu Jiang, Diogo Almeida, Carroll Wainwright, Pamela Mishkin, Chong Zhang, Sandhini Agarwal, Katarina Slama, Alex Ray, et al. 2022. Training language models to follow instructions with human feedback. *Advances in neural information processing systems*, 35:27730–27744.
- Qizhi Pei, Lijun Wu, Kaiyuan Gao, Xiaozhuan Liang, Yin Fang, Jinhua Zhu, Shufang Xie, Tao Qin, and Rui Yan. 2024. [BiT5+: Towards generalized biological understanding with IUPAC integration and multi-task tuning](#). In *Findings of the Association for Computational Linguistics ACL 2024*, pages 1216–1240, Bangkok, Thailand and virtual meeting. Association for Computational Linguistics.
- Qizhi Pei, Wei Zhang, Jinhua Zhu, Kehan Wu, Kaiyuan Gao, Lijun Wu, Yingce Xia, and Rui Yan. 2023. [BiT5: Enriching cross-modal integration in biology with chemical knowledge and natural language associations](#). In *Proceedings of the 2023 Conference on Empirical Methods in Natural Language Processing*, pages 1102–1123. Association for Computational Linguistics.
- Tiago Pereira, Maryam Abbasi, Bernardete Ribeiro, and Joel P Arrais. 2021. Diversity oriented deep reinforcement learning for targeted molecule generation. *Journal of cheminformatics*, 13(1):21.
- Daniil Polykovskiy, Alexander Zhebrak, Benjamin Sanchez-Lengeling, Sergey Golovanov, Oktai Tatanov, Stanislav Belyaev, Rauf Kurbanov, Alexsey Artamonov, Vladimir Aladinskiy, Mark Veselov, Artur Kadurin, Simon Johansson, Hongming Chen, Sergey Nikolenko, Alan Aspuru-Guzik, and Alex Zhavoronkov. 2020. Molecular Sets (MOSES): A Benchmarking Platform for Molecular Generation Models. *Frontiers in Pharmacology*.
- Philipp Renz, Sohvi Luukkonen, and Günter Klambauer. 2024. Diverse hits in de novo molecule design: Diversity-based comparison of goal-directed generators. *Journal of Chemical Information and Modeling*, 64(15):5756–5761.
- David Rogers and Mathew Hahn. 2010. Extended-connectivity fingerprints. *Journal of chemical information and modeling*, 50(5):742–754.
- Jacob L. Rosenberg and David G. Baldwin. 1965. A stochastic difference equation for beam search. In *Proceedings of the International Conference on Machine Learning*, pages 275–283. ACM.
- Anastasiia V Sadybekov and Vsevolod Katritch. 2023. Computational approaches streamlining drug discovery. *Nature*, 616(7958):673–685.

- John Schulman, Filip Wolski, Prafulla Dhariwal, Alec Radford, and Oleg Klimov. 2017. Proximal policy optimization algorithms. *arXiv preprint arXiv:1707.06347*.
- Lior Shani, Aviv Rosenberg, Asaf Cassel, Oran Lang, Daniele Calandriello, Avital Zipori, Hila Noga, Or-gad Keller, Bilal Piot, Idan Szpektor, et al. 2024. Multi-turn reinforcement learning from preference human feedback. *arXiv preprint arXiv:2405.14655*.
- Yixuan Su, Tian Lan, Yan Wang, Dani Yogatama, Ling-peng Kong, and Nigel Collier. 2022. A contrastive framework for neural text generation. *Advances in Neural Information Processing Systems*, 35:21548–21561.
- Hugo Touvron, Thibaut Lavril, Gautier Izacard, Xavier Martinet, Marie-Anne Lachaux, Timothée Lacroix, Baptiste Rozière, Naman Goyal, Eric Hambro, Faisal Azhar, Aurelien Rodriguez, Armand Joulin, Edouard Grave, and Guillaume Lample. 2023. [Llama: Open and efficient foundation language models](#). Preprint, arXiv:2302.13971.
- Oleg Trott and Arthur J. Olson. 2010. [Autodock vina: improving the speed and accuracy of docking with a new scoring function, efficient optimization, and multithreading](#). *Journal of Computational Chemistry*, 31(2):455–461.
- Jessica Vamathevan, Dominic Clark, Paul Czodrowski, Ian Dunham, Edgardo Ferran, George Lee, Bin Li, Anant Madabhushi, Parantu Shah, Michaela Spitzer, et al. 2019. Applications of machine learning in drug discovery and development. *Nature reviews Drug discovery*, 18(6):463–477.
- Ashwin Vijayakumar, Michael Cogswell, Ramprasaath Selvaraju, Qing Sun, Stefan Lee, David Crandall, and Dhruv Batra. 2018. [Diverse beam search for improved description of complex scenes](#). *Proceedings of the AAAI Conference on Artificial Intelligence*, 32(1).
- Ashwin K Vijayakumar, Michael Cogswell, Ram-prasath R Selvaraju, Qing Sun, Stefan Lee, David Crandall, and Dhruv Batra. 2016. Diverse beam search: Decoding diverse solutions from neural sequence models. *arXiv preprint arXiv:1610.02424*.
- David Weininger. 1988. Smiles, a chemical language and information system. 1. introduction to methodology and encoding rules. *Journal of chemical information and computer sciences*, 28(1):31–36.
- Sifan Wu, Amir Khasahmadi, Mor Katz, Pradeep Kumar Jayaraman, Yewen Pu, Karl Willis, and Bang Liu. 2023a. Cad-LLM: Large language model for cad generation. In *NeurIPS 2023 Workshop on Machine Learning for Creativity and Design*. NeurIPS.
- Zequi Wu, Yushi Hu, Weijia Shi, Nouha Dziri, Alane Suhr, Prithviraj Ammanabrolu, Noah A. Smith, Mari Ostendorf, and Hannaneh Hajishirzi. 2023b. [Fine-grained human feedback gives better rewards for language model training](#). In *Thirty-seventh Conference on Neural Information Processing Systems*.
- Yutong Xie, Ziqiao Xu, Jiaqi Ma, and Qiaozhu Mei. 2023. [How much space has been explored? measuring the chemical space covered by databases and machine-generated molecules](#). In *The Eleventh International Conference on Learning Representations*.
- Geyan Ye, Xibao Cai, Houtim Lai, Xing Wang, Jun-hong Huang, Longyue Wang, Wei Liu, and Xiangxiang Zeng. 2023. Drugassist: A large language model for molecule optimization. *arXiv preprint arXiv:2401.10334*.
- Botao Yu, Frazier N. Baker, Ziqi Chen, Xia Ning, and Huan Sun. 2024. [LlaSMol: Advancing large language models for chemistry with a large-scale, comprehensive, high-quality instruction tuning dataset](#). In *First Conference on Language Modeling*.
- Manzil Zaheer, Satwik Kottur, Siamak Ravanbakhsh, Barnabas Poczos, Russ R Salakhutdinov, and Alexander J Smola. 2017. [Deep sets](#). In *Advances in Neural Information Processing Systems*, volume 30. Curran Associates, Inc.
- Yifei Zhou, Andrea Zanette, Jiayi Pan, Sergey Levine, and Aviral Kumar. 2024. Archer: Training language model agents via hierarchical multi-turn rl. *arXiv preprint arXiv:2402.19446*.
- Le Zhuo, Zewen Chi, Minghao Xu, Heyan Huang, Heqi Zheng, Conghui He, Xian-Ling Mao, and Wentao Zhang. 2024. ProtLLM: An interleaved protein-language LLM with protein-as-word pre-training. *arXiv preprint arXiv:2403.07920*.

## A Additional Related Works

### A.1 Reinforcement Learning (RL) for Diverse Molecular Generation

Existing literature has studied RL-based methods to generate molecules with desired properties while enhancing their diversity. First, Blaschke et al. (2020); Pereira et al. (2021) introduced memory-assisted RL, which penalizes the reward of a molecule when it is highly similar to the molecules stored in the memory unit. He et al. (2024) also incorporated RL with a diversity penalty in transformer-based architectures for molecular generation. In addition, Hu et al. (2024) leveraged multiple GPT-based agents trained with RL to encourage these agents to explore diverse directions for discovering diverse molecules. Their algorithms are designed to consider a fixed target property. In contrast, our work fine-tunes LLMs to generate diverse molecules given a prompt that is flexible to describe various target properties.

### A.2 Molecular Similarity Measures

In this section, we explain the structural similarity between two molecules. These measures are used to define the reward of reinforcement learning (Appendix B) and diversity metrics (Appendix A.3).

**Molecular structural features.** First, molecular structural features are expressed with Morgan fingerprint (Rogers and Hahn, 2010), which is a vector that characterizes the presence of specific atoms, bonds, or substructures in a given molecule. Note that we get this from RDkit package (Landrum, 2016). We denote this with  $f(m)$ , which maps the molecule  $m$  to its Morgan fingerprint. Next, we compute the molecular structural similarity based on the molecular fingerprints.

**Tanimoto similarity.** The overall similarity between two molecules is typically evaluated as follows:

$$T(m_i, m_j) = \frac{|f(m_i) \cap f(m_j)|}{|f(m_i) \cup f(m_j)|},$$

where  $f(m_i)$  maps the molecule  $m_i$  to its Morgan fingerprint. This similarity  $T(m_i, m_j)$  is referred to as Tanimoto similarity (Bajusz et al., 2015), evaluating overall structural similarity.

**Dice similarity.** In this paper, we also consider Dice similarity (Bajusz et al., 2015) which is sensitive to the degree of shared structural features between two molecules. This is defined as follows:

$$D(m_i, m_j) = \frac{2|f(m_i) \cap f(m_j)|}{|f(m_i)| + |f(m_j)|},$$

where  $\langle \cdot, \cdot \rangle$  denotes a dot product between two vectors.

### A.3 Molecular Diversity Metrics

In this section, we provide a detailed explanation of the diversity metrics for evaluating the given set of molecules. Specifically, we explain two diversity metrics: the number of circles (Xie et al., 2023) and the internal diversity (Polykovskiy et al., 2020).

**The number of circles (NCircles; Xie et al., 2023).** To evaluate the diversity of a given set of molecules  $\mathcal{M}$ , this computes the size of the largest subset of molecules in which no two molecules are similar to each other. This metric is defined with a Tanimoto similarity  $T(\cdot, \cdot)$  and a similarity threshold  $h$  as follows:

$$\text{NCircles}_h = \max_{\mathcal{C} \subseteq \mathcal{M}} |\mathcal{C}| \quad \text{s.t. } T(x, y) < h, \forall x \neq y \in \mathcal{C}, \quad (2)$$

where  $\mathcal{C}$  is a subset of molecules. Every pair of molecules in  $\mathcal{C}$  should have a similarity lower than  $h$ . The high NCircles value implies that the given set of molecules  $\mathcal{M}$  is diverse and covers a wide range of molecular space (Xie et al., 2023). Recent work (Ren et al., 2024) haven shown that this metric is relatively exact to measure the molecular diversity, compared to the other metrics, e.g., internal diversity. **Internal diversity (IntDiv; Polykovskiy et al., 2020).** Given a set of molecules  $\mathcal{M}$ , this metric measures the average of pair-wise Tanimoto similarities:

$$\text{Intdiv.} = \frac{1}{|\mathcal{M}| \cdot (|\mathcal{M}| - 1)} \sum_{i=1}^{|\mathcal{M}|} \sum_{j=i+1}^{|\mathcal{M}|} (1 - T(m_i, m_j)), \quad (3)$$

where  $m_i$  is  $i$ -th molecule in the given set of molecules  $\mathcal{M}$ .

## B Detailed Implementations and Training

### B.1 Supervised Fine-tuning

**Dataset collection.** To fine-tune BioT5<sup>+</sup> (Section 4.1), we collect  $T = 100$  molecules using contrastive beam search for each training molecular description in ChEBI-20 training split. In the case of molecular descriptions from the L+M-24 training split, we use only the provided set of target molecules for each description, without additional data collection. To fine-tune DrugAssist (Section 4.3), we collect  $T = 300$  molecules using contrastive beam search for each training prompt. Note that the collected molecules were filtered to remove invalid string representations, duplicated molecules, and unaccepted molecules. The invalid string representations are evaluated with RDKit package (Landrum, 2016). Additionally, the collected molecules are concatenated into a single sequence  $m_1 || \dots || m_K$ .

**Supervised learning.** We consider four NVIDIA A100 SXM4 40GB GPUs for supervised fine-tuning.

- For the supervised fine-tuning of BioT5<sup>+</sup> (Section 4.1), we consider 80 epochs, 8-batch size, 5e – 4 learning rate, 0.05 warm-up ratio, and apply a cosine learning scheduler. The maximum sequence length in supervised training is limited to 2560 due to memory limitations.
- For the supervised fine-tuning of DrugAssist (Section 4.3), we consider 80 epochs, 4-batch size, 3e – 5 learning rate, 0.05 warm-up ratio, and apply a cosine learning scheduler. The maximum sequence length in supervised training is limited to 1024 due to memory limitations. We also apply LoRA (Hu et al., 2022), where the rank and alpha are 64 and 128, respectively.

### B.2 Reinforcement Learning

**Reward design.** In experiments on L+M-24 and ChEBI-20 datasets (Section 4.1), we define the description-matching reward using the Dice similarity as follows:

$$r_{\text{match}}(m_k, p_{\text{desc}}) = \max_{m_{\text{target}} \in \mathcal{M}_{\mathcal{D}}(p_{\text{desc}})} D(m_{\text{target}}, m_k)^{\alpha}, \quad (4)$$

where  $\mathcal{M}_{\mathcal{D}}(p_{\text{desc}})$  is a set of target molecules provided in the datasets for each molecular description  $p_{\text{desc}}$ .<sup>7</sup> As described in Appendix A.2,  $D(m_i, m_j)$  is Dice similarity capturing the degree of shared structural features between two molecules. Note that  $\alpha$  is a hyper-parameter. In experiments with DrugAssist (Section 4.3), the description-matching reward yields 1 if the molecule satisfies the quantitative properties described in  $p_{\text{desc}}$  (evaluated using RDkit (Landrum, 2016)) and 0 otherwise.

Next, the diversity reward,  $r_{\text{div}}$ , is defined to consider molecular structural diversity between the new molecule  $m_k$  and the previously generated molecules  $\{m_i\}_{i=1}^{k-1}$  within a sequence:

$$r_{\text{div}}(m_k, \{m_i\}_{i=1}^{k-1}) = 1 - \max_{m \in \{m_i\}_{i=1}^{k-1}} T(m_k, m)^{\beta}, \quad (5)$$

where  $T(m_i, m_j)$  is Tanimoto similarity (Appendix A.2) between  $m_i$  and  $m_j$  and  $\beta$  is a hyper-parameter.

**Policy optimization.** We consider four NVIDIA A100 SXM4 40GB for reinforcement learning implemented with proximal policy optimization.

- For the reinforcement learning of BioT5<sup>+</sup> (Section 4.1), we consider 200 PPO iterations, 8 mini-batch size, 128 batch size, and 5e – 5 learning rate. We also consider 0.01 KL penalty. Note that  $\alpha$  and  $\beta$  in Equations (4) and (5) are 0.5 and 1.0, respectively. We increase  $\beta$  to 2.0 during training on ChEBI-20 dataset. The reward signal is amplified by multiplying by a value of 8.0. The maximum sequence length in reinforcement learning is limited to 2560 due to memory limitations. Here, we also apply LoRA (Hu et al., 2022) where the rank and alpha are 16 and 32, respectively.
- For the reinforcement learning of DrugAssist (Section 4.3), we consider 200 PPO iterations, 4 mini-batch size, 64 batch size, and 3e – 6 learning rate. We also consider 0.1 KL penalty. Note that  $\beta$  in Equation (5) is 2.0. The reward signal is amplified by multiplying by a value of 8.0. The maximum sequence length in reinforcement learning is limited to 1024 due to memory limitations. We also apply LoRA (Hu et al., 2022) where the rank and alpha are 64 and 128, respectively.

<sup>7</sup>As described in Section 4.1, we assume that the molecule satisfies the description if it shares a certain degree of structures with one of the target molecules provided in the dataset.

## C Dataset Details

In this section, we describe detailed data statistics of ChEBI-20 ([Edwards et al., 2021](#)) and L+M-24 datasets ([Edwards et al., 2024](#)). Both datasets serve to validate description-guided molecular generation capability. Note that both datasets consider a molecular description expressed in English and a molecule represented by SMILES ([Weininger, 1988](#)). In [Table 7](#), we provide the number of molecular descriptions.

To be specific, L+M-24 dataset provides an average of 70 and 17 target molecules for each description in training and test splits, respectively. Note that the original L+M-24 dataset provides a single target molecule for each description. However, L+M-24 dataset involves identical molecular properties for different molecules. Moreover, some molecular descriptions, e.g., an inhibitor of both BCL2 and BTK proteins, include the molecular properties of other descriptions, e.g., an inhibitor of BCL2 protein. Thus, we associate multiple molecules with a single molecular description. Next, ChEBI-20 dataset provides a single example of target molecule for each description.

In addition, since the original BioT5<sup>+</sup> is not pre-trained on L+M-24 dataset, we further train it on the original L+M-24 training split. Note that we use the first 1,000 molecular descriptions in the test split of the L+M-24 dataset for evaluation, as evaluating the entire test split with 50 generated molecules per description takes too much time (five to six days).

**Table 7: Data statistics.** The number of molecular descriptions.

Dataset	# of training	# of test
L+M-24	160,492	21,839
ChEBI-20	26,408	3,301

## D Experiments Setup

Our experiments consider various molecular generative LLMs: MolT5 (Edwards et al., 2022), BioT5<sup>+</sup> (Pei et al., 2024), Text+Chem T5 (Christofidellis et al., 2023), LlasMol (Yu et al., 2024), DrugAssist (Ye et al., 2023), and Meditron (Chen et al., 2023).<sup>89</sup> They consider a molecular description expressed in English. They also consider a molecule represented as SMILES (Weininger, 1988), except for BioT5<sup>+</sup>, which uses SELFIES (Krenn et al., 2020). T5 models have 220M parameters, and others have 7B parameters. We use four NVIDIA A100 SXM4 40GB GPUs for the experiments. We consider a single run.

**Comparison with decoding schemes.** In this experiment, we first consider random sampling with different temperatures {0.7, 1.0, 1.5}. For the other decoding schemes, we consider conventional configurations: nucleus sampling with top-p 0.8, beam search, diverse beam search with a diversity penalty of 0.5, and contrastive beam search with a penalty alpha of 0.5. We apply greedy decoding for our approach. The prompts for BioT5<sup>+</sup> are described in Table 8.

Table 8: **Prompts for BioT5<sup>+</sup> (Pei et al., 2024).**

Prompt	Contents
$p_{desc}$	“Definition: You are given a molecule description in English. Your job is to generate the molecule SELFIES that fits the description. Now complete the following example - Input: <molecular description> Output: ”
$p_{desc+div}$ (fine-tuning)	“Definition: You are given a molecule description in English. Your job is to generate the molecule SELFIES that fits the description. Now provide a set of molecules - Input: <molecular description> Output: ”

**Comparison with existing LLMs.** For all existing LLMs, we apply random sampling, beam search, diverse beam search with a diversity penalty of 0.5, and contrastive beam search with a penalty alpha of 0.5. The prompts for existing LLMs, which are molecular generative LLMs based on a given molecular description, are described in Table 9.

Table 9: **Prompts for various existing LLMs**

Method	$p_{desc}$
MolT5	“<molecular description>”
Meditron	“Below is an instruction that describes a task, paired with an input that provides further context. Write a response that appropriately completes the request. ### Instruction: You are a researcher. You can come up molecule smile strings based on your existing knowledge. Molecule smile strings are given against the following input. You should be as detailed as possible. Input: <molecular description> In that caption, could you generate a molecule smile string? ### Response: ”
Text+Chem T5	“<molecular description>”
LlasMol	“Give me a molecule that satisfies the conditions outlined in the description: <molecular description>”

<sup>8</sup>MolT5, BioT5<sup>+</sup>, Text+Chem T5, LlasMol, and DrugAssist are licensed under the MIT License.

<sup>9</sup>Meditron is licensed under the Apache 2.0 License.

Table 10: **Prompts for DrugAssist (Ye et al., 2023).**

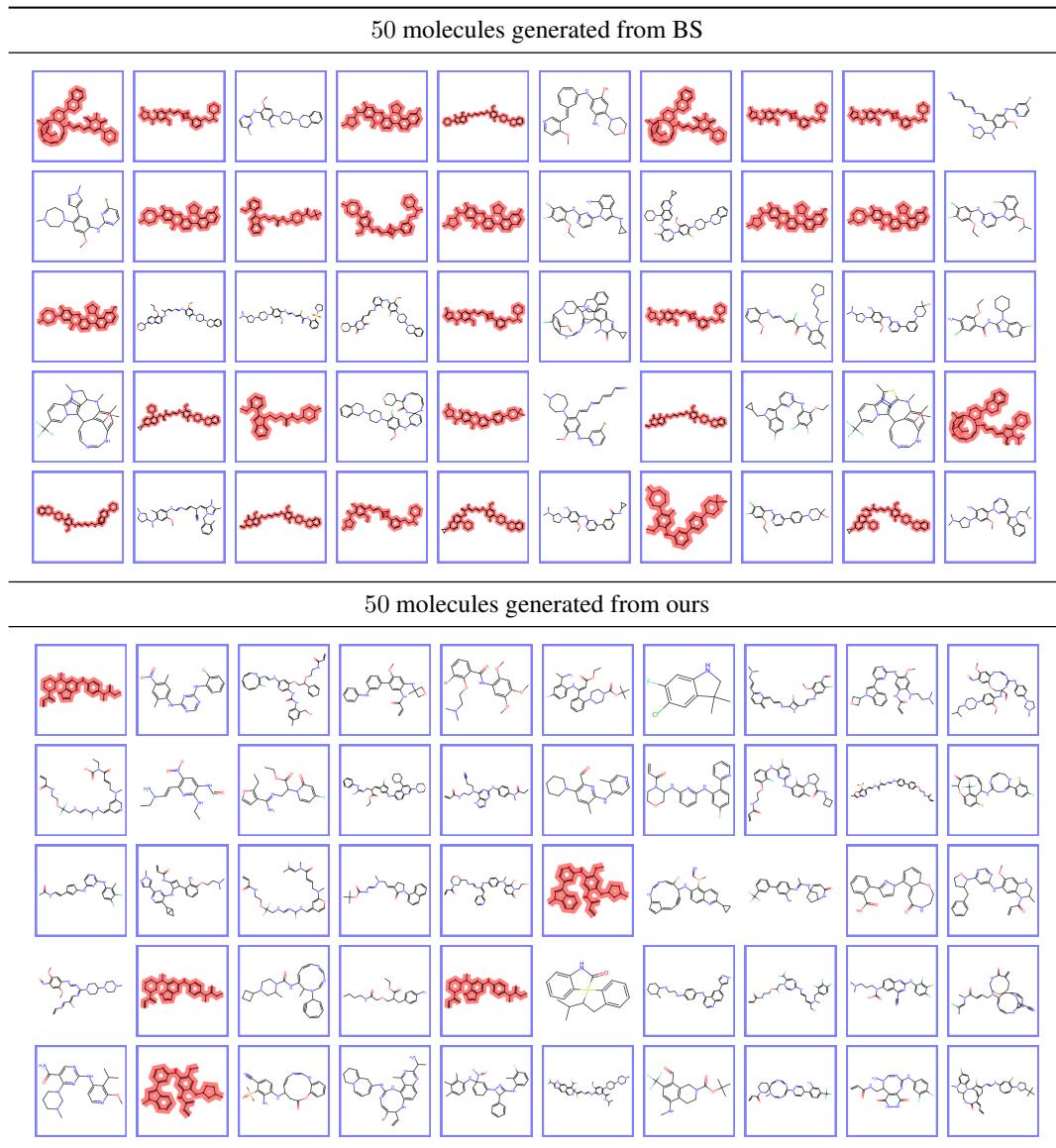
Prompt	Contents
$p_{desc}$	Hydrogen bond donors and acceptors: “Can you generate a molecule with <value> <property>? Print it in SMILES format.”
	QED and Bertz complexity: “Can you generate a molecule with <property> below <value1> but at least <value2>? Print it in SMILES format.”
$p_{desc+div}$ (fine-tuning)	Hydrogen bond donors and acceptors: “Can you generate a set of molecules that have <value> <property>? Print each of them in SMILES format.”
	QED and Bertz complexity: “Can you generate a set of molecules that have <property> below <value1> but at least <value2>? Print each of them in SMILES format.”

Table 11: **Prompts for DrugAssist (without fine-tuning, Figure 1).**

$p_{desc+div}$	“Can you generate a set of molecules? Each molecule has <value> <property>. Print each of them in SMILES format.”
	“Can you generate a diverse set of molecules? Each molecule has <value> <property>. Print each of them in SMILES format.”
	“Can you generate a structurally diverse set of molecules? Each molecule has <value> <property>. Print each of them in SMILES format.”
	“Can you generate a set of molecules with <value> <property>? Print each of them in SMILES format.”
	“Can you generate a diverse set of molecules with <value> <property>? Print each of them in SMILES format.”
	“Can you generate a structurally diverse set of molecules with <value> <property>? Print each of them in SMILES format.”

**Fine-tuning generalist LLMs.** We synthesize 600 pairs of training prompts and corresponding sets of molecules. The prompts specify hydrogen bond donors and acceptors ranging from one to four, and a Bertz complexity ranging from 0 to 300. The sets of molecules are collected by applying beam search on DrugAssist and then perturbed by shuffling their order. We use the prompts described in Table 10. The system prompt follows the default settings from (Ye et al., 2023). For evaluation, we use four prompts specifying three hydrogen bond donors, three hydrogen bond acceptors, a Bertz complexity between 100 and 200, and a QED value between 0.4 and 0.6. Additionally, as shown in Figure 1, we try to generate diverse molecules by designing prompts (Table 11). However, these prompts show lower performance compared to applying beam search.

**Table 12: Visualization of the generated molecules.** We generate 50 molecules from: “The molecule is a egfr inhibitor and modulate, belonging to the cancer treatment class of molecules, and is treatment of disorder”. The blue line indicates the molecule that follows the given description, as evaluated using the external tool (docking tool; Trott and Olson, 2010). We also evaluate structurally similar molecules (Tanimoto similarity > 0.7).



## E Additional Results

In this section, we present qualitative results by evaluating whether the generated molecules satisfy the given molecular description using an external tool. Here, we consider the generation of a set of molecules from: “The molecule is a egfr inhibitor and modulate, belonging to the cancer treatment class of molecules, and is treatment of disorder” obtained from L+M-24 test split. Note that the molecule with sufficient binding affinity to EGFR can satisfy this description. Thus, we evaluate the generated molecule using a docking tool (García-Ortegón et al., 2022). We accept the generated molecule if it yields a Vina score (Trott and Olson, 2010) below  $-7.0$  when docked with an EGFR protein, which is the threshold for a potential inhibitor of the protein. We present the results in Table 12. One can see that BS generates structurally similar molecules while our method generates structurally diverse molecules.

## **F Use of AI Assistants**

We used AI-based writing assistants to improve the sentence structure and grammar. These tools were used only for editorial improvements. The technical content, methodology, and experimental results were entirely authored by the researchers.