

MOLRGEN: A BENCHMARK AND VERIFIER FOR LLMS ON DE NOVO MOLECULAR GENERATION

Anonymous authors
Paper under double-blind review

ABSTRACT

Recent large language models have shown strong performance on tasks with verifiable outcomes, but their use in de novo molecular generation remains limited by the lack of training environments where rewards can be computed without reference molecules. We introduce MOLRGEN, a benchmark and molecular verifier for post-training and evaluating LLMs on de novo molecular generation.[1] MOLRGEN contains approximately 4,500 protein-pocket targets, resulting in 50k multi-objective optimization prompts combining docking scores with molecular properties such as QED, synthetic accessibility, logP, and physicochemical descriptors. Unlike caption-based generation or molecule-editing benchmarks, MOLRGEN evaluates molecules proposed from scratch by computing rewards at generation time. We benchmark general-purpose and chemistry-specialized open-source LLMs and introduce a diversity-aware top-k metric to measure whether models can generate a diverse set of high-scoring molecules. Finally, we use the verifier to fine-tune a 128B LLM with GRPO, showing improved performance, at the cost of a diversity-exploitation trade-off.

1 INTRODUCTION

The difficulty of identifying molecules with the properties required to become viable drugs has motivated the development of a wide range of molecular generative models, enabling faster exploration of chemical space (Loeffler et al., 2024; Tripp & Hernández-Lobato, 2023). This task is particularly challenging because of its inherently multi-objective nature: generated candidates must simultaneously satisfy multiple criteria, such as modulating the expression of a specific protein, reducing toxicity, and having adequate pharmacokinetic properties (e.g., oral bioavailability or the ability to cross the intestinal barrier). These requirements are typically aggregated into a single reward or scoring function to facilitate optimization and decision-making.

In this work, we focus on *de novo* molecular generation, a setting in which no high-scoring molecule is known *a priori*, and the generative algorithm must identify promising molecules without prior knowledge of which structures may be viable. Several approaches have been explored to address this challenge, with reinforcement learning (RL) emerging as a natural paradigm: an agent generates molecular compounds, which are then evaluated using a scoring function, often based on computational tools such as docking software (Hassan et al., 2017; Verdonk et al., 2003), and the resulting scores guide future generations.

Recent work has successfully applied LLMs to various chemical tasks, including caption-based molecular generation and molecular optimization (editing existing molecules to alter properties), and reasoning-based LLMs to drug discovery (Li et al., 2025; Narayanan et al., 2025), motivated by advances achieved in other domains such as mathematical problem-solving and coding. These advances can notably be attributed to the development of reinforcement learning (RL) post-training pipelines (Zhang et al., 2026; He et al., 2026), where a model is trained to maximize the rewards associated to its completions, computed at training time.

[1] Code

Reviewers: please read the Reviewer Guidelines (iclr.cc/Conferences/2027/ReviewerGuidelines) and the AI Policy for Reviewers (iclr.cc/Conferences/2027/AIPolicyForReviewers). If you used AI to expand, edit, or polish your review, please provide the input text to the LLM. Better still, consider skipping the LLM and submitting your original text: we, and the authors, are much more interested in your unedited thoughts than in what an LLM has to say. AI-assisted or not, you are putting your name and reputation behind your review: LLM-generated falsehoods, hallucinations or misrepresentations are subject to disciplinary action, which may include desk-rejecting all papers you have authored.

These studies have demonstrated that LLMs can effectively learn molecular representations and reasoning patterns when reference data is available (e.g., a molecule corresponding to the input caption). Still, such reference data is unavailable in practice for *de novo* generation. Filling this gap, MOLRGEN provides a large-scale, reference-free, online verification environment for de novo molecular design: the model does not have to recover a known molecule, and the verifier can evaluate arbitrary newly generated candidates online. To the best of our knowledge, there is currently no large-scale benchmark including *de novo* molecular generation tasks to train and evaluate LLMs, using rewards computed at generation time.

Our aim is not to position LLMs as replacements for domain-specific computational tools, but rather to evaluate their potential relevance as components within a broader molecular design pipeline—understanding both their capabilities and limitations in this context. In this work, we aim at answering the two following questions: **Can current general-purpose reasoning LLMs perform open-ended molecular design under a computational verifier, and can that verifier be used to teach a general-purpose LLM to become a better at molecular generation?** We summarize our contribution as follows:

- We introduce MOLRGEN, a large-scale dataset for *de novo* molecular generation, leveraging 4.5k protein pocket targets, which we completed with property prediction auxiliary tasks.
- We release an open-source molecular verifier that computes rewards at generation time, enabling RL training without reference high-scoring molecules.
- We evaluate multiple open-source LLMs on our benchmark, and show that our verifier can be leveraged to train LLMs with RL, improving their performance on the generation task.

2 RELATED WORK

Molecular Generation. Molecular generation has been a long-standing challenge in computational chemistry and drug discovery. Traditional approaches include generative models, such as Variational Autoencoders (VAEs) (Gómez-Bombarelli et al., 2018) and genetic algorithms (Tripp & Hernández-Lobato, 2023), which have demonstrated strong empirical results in generating novel molecules. Reinforcement learning (RL) based methods have emerged as a natural paradigm for generating drug candidates while optimizing for a desirability score. Notably, Reinvent (Loeffler et al., 2024) generates SMILES representations of molecules (textual representation) by adjusting the log-likelihood of a prior model using this desirability score. Other approaches propose generating molecules in different modalities, such as graph representations (Koziarski et al., 2024; Gao et al., 2024; Haddad et al., 2025) or 3D point clouds (Zhou et al., 2025). BindGPT (Zholus et al., 2025) recently proposed to train a 100M parameter language model architecture, using GRPO to optimize its generation on 100 docking targets. While using a language model architecture, this model is not trained to generate or process natural language.

Large Language Models in Chemistry. The application of large language models (LLMs) to chemistry has rapidly expanded, with early work assessing LLM knowledge in chemistry (Mirza et al., 2024), showing that the majority of these models outperform human scores. To further advance the comprehension of chemistry by these models, various methods have been proposed to train and evaluate language models on fundamental tasks, such as SMILES-to-IUPAC translation, molecular captioning, and molecular generation from structural descriptions Yu et al. (2024); Kim et al. (2025). Recent work has leveraged 3D molecular representations (Li et al., 2024) or diffusion models (Gong et al., 2024) to perform generation tasks. A critical distinction exists between caption-based generation and our approach. Caption-based generation aims to find compounds matching a given natural language description (e.g., *"The molecule is a primary arylamine [...] and a member of anilines"* from Mol-Instructions Fang et al. (2024)), while our *de novo* molecular generation objective seeks to create novel compounds that optimize a desirability score, rather than match a specific description (e.g., *"Generate a molecule that inhibits the expression of ..."*).

Reasoning in Drug Discovery. With the emergence of LLM reasoning models and their demonstrated capabilities on tasks such as mathematical problem-solving and coding, several works have investigated chain-of-thought reasoning for chemical and biological tasks. BioReason (Fallahpour et al., 2025) incorporated a multimodal architecture for DNA sequence tasks, combining fine-tuning

on chain-of-thought reasoning from standard LLMs with GRPO optimization on ground-truth KEGG labels. For molecular applications, reasoning-enhanced models (Zhuang et al., 2025) leverage structured reasoning to improve molecular property and toxicity predictions. Mol-R1 (Li et al., 2025), Ether0 (Narayanan et al., 2025), and Chem-DFM (Zhao et al., 2025) combine methods such as rejection sampling, distillation, supervised fine-tuning, or reinforcement learning to perform multiple chemical tasks, including chemical reaction understanding or caption-based molecular generation. GeLLMO (Dey et al., 2025) tackles a different problem: modifying existing molecules to change their properties, by using at training time a set of molecular pairs labeled with their property modifications.

3 A Benchmark for De Novo Molecular Generation

3.1 Notations

We denote $\mathcal{X}$ the set of all possible molecules. For each task, we denote n_{props} the number of tasks to be performed simultaneously (in practice, $n_{\text{props}} > 1$ only for generation tasks, where we optimize multiple properties simultaneously). For $1 \leq i \leq n_{\text{props}}$, we will use the abusive notation $\gamma_i : \mathcal{X} \to \mathbb{R}$ a molecular property, such as the docking score to a target pocket for molecular generation, or the mapping to the label of a molecule for property prediction (such as the solubility of the molecule) disregarding possible metadata required to compute the property (e.g., the target pocket for docking scores) for clarity of notations. Additionally, we will assume that all properties are rescaled in $[0, 1]$ to simplify the notations. For property prediction tasks, we also denote $x_i \in \mathcal{X}$ the molecule the model is tasked to predict the property of.

A prompt q is then defined as $q = \{\nu_i, \gamma_i, x_i, \sigma_i\}_{1 \leq i \leq n_{\text{props}}}$, where: ν_i is the task objective (e.g., property maximization, class prediction), and σ_i is a numerical metadata used to compute the reward (e.g., property threshold or label standard deviation).

Given a model π_θ and prompt q, a completion $o \sim \pi_\theta(q)$ is generated, and we extract the final answer $\hat{o}$ from tags `<answer>` and `</answer>`. The notation $\gamma_i(\hat{o})$ denotes the property value resulting from computing a property γ_i to the extracted completion (rescaled in $[0, 1]$). Table 1 illustrates these notations and provides the corresponding reward functions for each task.

3.2 De Novo Molecular Generation.

3.2.1 Data Collection

The primary task in our dataset is the *de novo* molecular generation task: generating a compound while optimizing its "desirability score". We focus on optimizing the predicted binding affinity to a target protein, which we measure with docking scores. To enable the optimization of these docking scores, we need to build an extensive dataset of protein structures with defined pockets (i.e., regions of the protein where generated molecules should bind).

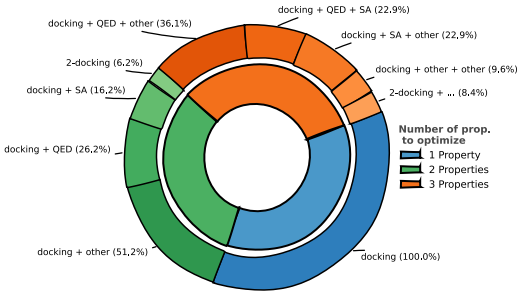


Figure 1: **Optimization tasks for de novo generation.** Optimization objectives in the dataset for de novo generation. The dataset is well-balanced between single-property (29%), 2-properties (36%), and 3-properties (35%) optimization.

Structure extraction. We extracted our receptors (protein pockets) from the SAIR dataset (Lemos et al., 2025), which consists of over 1M protein-ligand pairs from 5k unique protein structures. This dataset was built upon the structural prediction of Boltz1 (Wohlwend et al., 2024) (a cofolding model) and notably contains predicted protein structures for which no known experimental structure exists. To identify pockets for each structure, we performed a clustering of the protein's residues based on the predicted receptor-ligand structure (see details on the dataset creation in Appendix A) We completed this set with structures from the SIU dataset (Huang et al., 2024) (with labeled pockets). After cleaning all structures with Meeko (Santos-Martins et al., 2025), we obtained 4,500 structures with defined

Table 1: **Reward and prompt definition.** We propose different reward functions $r(q, \hat{o})$ depending on the objective ν_i of the task, that are designed to reflect how close the extracted answer $\hat{o}$ is to the desired outcome. All properties are normalized in $[0, 1]$.

Objective ν_i	Property γ_i	Reference x_i	Margin σ_i	Answer $\hat{o}$	Reward $r(q, \hat{o})$
***De Novo* Generation** (per-property reward)					**49k**
maximize	property	n/a	threshold	gen. SMILES	$\gamma_i(\hat{o})$
minimize	property	n/a	threshold	gen. SMILES	$1 - \gamma_i(\hat{o})$
below	property	n/a	threshold	gen. SMILES	$\mathbb{1}_{\gamma_i(\hat{o}) \leq \sigma_i}$
above	property	n/a	threshold	gen. SMILES	$\mathbb{1}_{\gamma_i(\hat{o}) \geq \sigma_i}$
Property Prediction					**55k**
regression	property	molecule	prop. std.	pred. value	$\max(0, 1 - \frac{(\hat{o} - \gamma_i(x_i))^2}{{\sigma_i}^2})$
classification	property	molecule	n/a	pred. value	$\mathbb{1}_{\hat{o} = \gamma_i(x_i)}$

pockets, each associated with a unique UniProt ID. We use AutoDock-GPU (Santos-Martins et al., 2021) to perform the docking simulations.

Objective Design. Once all protein structures have been extracted, we build our dataset by creating property-optimization combinations that will define the desirability score, or reward, to optimize. In addition to docking scores, we included 12 classical molecular properties computed using RDKit (Landrum et al., 2020), whose occurrence in our dataset is illustrated in Figure 1. All docking targets were sampled with equal probability, whereas classical molecular descriptors were sampled with varying probabilities. Specifically, we over-sampled the synthetic accessibility score (SA) (Ertl & Schuffenhauer, 2009) and the quantitative estimate of drug-likeness (QED) (Bickerton et al., 2012) to encourage the model to generate feasible, drug-like molecules. Finally, we created three splits: a training split of 49k prompts, and test and evaluation splits of 1k prompts, where the set of protein structures in the test, evaluation, and training splits is disjoint.

3.2.2 Evaluation of the Completions

Reward Design. Our next step is to define a reward function to evaluate how well a generated completion o answers a prompt $q = \{\nu_i, \gamma_i, x_i, \sigma_i\}_{1 \leq i \leq n_{\text{props}}}$. We provide in Table 1 a description of how this reward is computed depending on q for a single property optimization task: $\{\nu_i, \gamma_i, x_i, \sigma_i\} \quad 1 \leq i \leq n_{\text{props}}$. To aggregate multiple scores associated with multiple properties into a single reward $r(q, \hat{o})$, several strategies exist (Luukkonen et al., 2023), and we choose to rely on the geometric mean of the rewards associated with each property. Using the geometric mean compared to an arithmetic mean ensures that only completions effectively optimizing all properties are rewarded, compared to completions that would optimize only a subset of the properties. [2]

Top-k score. Our main objective is to evaluate the ability of a model to explore the chemical space (set of all possible molecules) to find high-reward molecules, which may be particularly challenging when high-reward molecules are rare and difficult to find. To evaluate this ability, we rely on the top_{k,n_r} score, which measures the average reward of the top-k best unique generated molecules when sampling $n_r \geq k$ completions (or rollouts). This score is simply computed by generating n_r completions $\{o_1, \ldots, o_{n_r}\} \sim \pi_\theta(q)$, extracting the final answers $\{\hat{o}_1, \ldots, \hat{o}_{n_r}\}$, and computing the average reward of the top-k best completions, when removing all completions associated to the same canonical SMILES string (padding with null values if not enough unique completions are obtained).

The top_{k,n_r} score accounts for duplicates and invalid answers, being assigned a reward of 0, effectively merging the evaluation of the model's ability to generate valid and unique molecules with its ability to generate high-reward molecules. However, it does not account for the diversity of the proposed molecules, which might be very similar to each other. This can prove problematic, indeed, in practice,

[2] Having a single property reward of 0 would lead the aggregated reward to be 0.

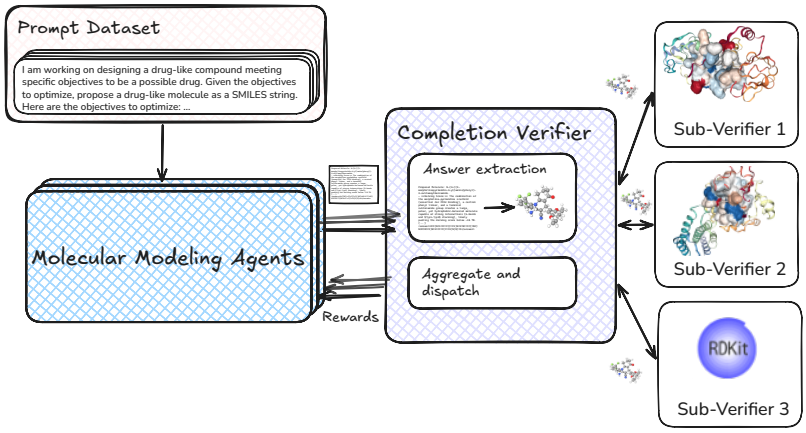


Figure 2: **Illustration of an inference/training using our MolecularVerifier.** Multiple models/clients are used to generate completions for a dataset of prompts. The MolecularVerifier, computes the associated reward for each prompt-completion pair by: extracting the SMILES string of the generated molecule, computing the per-property rewards and aggregating these per-property rewards into a single reward, and the computed rewards are then sent back to each model.

it is often desirable to generate a set of high-reward molecules that are also diverse, to obtain a range of candidates from different chemical series, and to mitigate the risk of failure of a specific chemical series in later stages of drug discovery.

Diversity-aware Top-k Score. To address this limitation, we propose to measure the top-k score on molecules with a constraint on their mutual similarity controlled by $s_{\max}$ ($0 < s_{\max} < 1$). We introduce the diversity-aware top-k score: $\text{top}^{\text{diversity}}_{k,n_r}(\pi_\theta, q, s_{\max})$.

To construct this score from a set of completions $\{o_1, \ldots, o_{n_r}\} \sim \pi_\theta(q)$, we perform the following procedure: (1) we select the highest reward molecule from the set of completion, (2) we remove all molecules that are too similar to the selected molecule (i.e., with a Tanimoto similarity above $s_{\max}$), and (3) we repeat this process until we have selected k molecules or until no more molecules can be selected. Finally, we return the average reward of the selected molecules, padded with null values if not enough molecules satisfy the similarity constraint.

This algorithm is a greedy approximation of the following combinatorial optimization problem:

$$\begin{aligned} &\max_{\{i_1,\ldots,i_k\}\subset\{1,\ldots,n_r\}} \frac{1}{k} \sum_{j\in\{i_1,\ldots,i_k\}} r(q,\hat{o}_j), \\ &\text{s.t.} \quad \forall j \in \{i_1,\ldots,i_k\}, \forall l \in \{1,\ldots k\}, j \neq l, \quad \text{sim}(\hat{o}_j, \hat{o}_l) < s_{\max} \end{aligned} \tag{1}$$

which corresponds to selecting k molecules such that their pairwise similarities are below $s_{\max}$, and the sum of their rewards is maximized.

Our diversity-aware top-k score measures the trade-off between the ability of the model to generate high-reward molecules and its ability to explore the chemical space. We will show in the next section that this metric provides additional insights into the performance of the models.

3.3 Training with the Molecular Verifier

Molecular Verifier. To compute the rewards defined in Table 1 at generation time, we developed a *Molecular Verifier*, which, given a prompt-completion pair, computes its associated reward by running docking simulations and computing the physico-chemical properties. This verifier can be used to evaluate pre-trained models, but also to fine-tune language models with reinforcement learning.

The *Molecular Verifier* enables the computation of the rewards asynchronously, hence enabling multiple queries from multiple clients to be processed in parallel. To compute the reward for a given prompt-completion pair, the verifier first parses the completion to extract the SMILES string of the generated molecule, or the predicted property value. Then the verifier routes the extracted information to the appropriate sub-verifier, which computes the per-property reward for the given prompt-completion pair. The sub-verifiers run domain specific tools if necessary such as docking simulations or prediction of physico-chemical properties with RDKIT (Landrum et al., 2020). Finally, the per-property-rewards are aggregated (as described in subsubsection 3.2.2) and the final is returned to the client, which can then use it to update the model. All of these steps are performed in parallel using Ray (Moritz et al., 2017), which also handles the distribution of the workload across multiple GPUs and CPUs if necessary.

Reinforcement Learning. We leveraged this verifier to fine-tune the Mistral-Small-4, containing 128B parameters using Group Relative Policy Optimization (GRPO) (Shao et al., 2024). GRPO aims at training a model to maximize the expected reward of its completions by generating a group of completions for each prompt, increasing the probability of completions that outperform the group average. Further details on the GRPO training procedure can be found in subsection D.1.

In the following section, we display results for 3 models trained for 250 steps, all trained with 32 prompts per step. We heavily focused the training of the models on the molecular generation task, with 66% of the prompts sampled from the molecular generation task, and 33% from the molecular property prediction task. These models are denoted as RL-Molstral-n_r where $n_r \in \{4, 8, 16\}$ is the group size used for training (i.e., the number of completions generated per prompt during training). We scale the batch size with the group size to maintain the same number of unique prompts per batch.

3.4 Molecular Property prediction.

To complete our trainig dataset, we include molecular property prediction tasks, where the model is tasked to predict a given property for a molecule. We extracted property prediction tasks from the Polaris platform (Wognum et al., 2024), aggregating property prediction benchmarks from academic and industrial research. For property prediction tasks, $n_{\text{props}} = 1$, and the task is fully defined by: $\nu_1, \gamma_1, x_1,$ and σ_1, where γ_i is the property to predict, ν_1 the type of task (regression or classification), x_1 is the molecule to predict the property of, and σ_1 is the standard deviation of the training labels, used to normalize the predicted error for regression tasks. The reward functions for property prediction tasks are defined in Table 1 (we present our results on the property prediction tasks in Appendix F).

4 Results

4.1 LLM Evaluation

Baseline models. We evaluate a diverse set of models across two categories: (1) **General-purpose LLMs** including Qwen3-30B-A3B-Thinking and Qwen3-Next-80B-A3B-Thinking (Team, 2025b), DeepSeek-R1 (distilled into Llama-70B and Qwen-32B backbones for reasonable experimental cost) (DeepSeek-AI, 2025), Gemma-3-27B (Team, 2025a), Gemma-4-31B (Team, 2026), MiniMax-M2 (MiniMax, 2025), GPT-OSS (OpenAI, 2025) and Llama-3.3-70B-Instruct (Team, 2024), and (2) **Chemically-specialized models** including ChemDFM-v2.0, ChemDFM-R (Zhao et al., 2025), and Ether0 (Narayanan et al., 2025).

Top-k performances of pretrained LLMs. Table 2 presents the top_{k,n_r} performances on the de novo molecular generation task. Among all evaluated models, Gemma-4 achieves the best overall performance in terms of top_{k,n_r} scores, outperforming both other general-purpose LLMs and chemically specialized models. In general, the most recent models tend to perform better than older ones, and we measured a Spearman correlation of 0.8 between the release date and the top_{k,n_r} score on average (correlation between the rankings induced by the two variables). This suggests that improvements in other domains, such as code generation and mathematical reasoning, do transfer to some extent to molecular generation. Conversely, we measured a lower Spearman correlation of 0.2

Table 2: ***De novo* generation results.** Evaluation of various LLMs on the molecular generation task. We report the average top-k scores among n_r generations per prompt and standard deviations, averaged over the unique prompts. Models are ordered by their release date.

Model	Size	n_r =16		n_r =24		n_r =32		n_r =48	
		top_1	top_{16}	top_1	top_{16}	top_1	top_{16}	top_1	top_{16}
General LLMs									
gemma-4	31B	**0.65** ±0.05	0.37 ±0.05	**0.67** ±0.04	0.48 ±0.03	**0.68** ±0.03	0.52 ±0.02	0.7 ±0.01	0.56 ±0.01
Mistral-Small-4	128B	0.6 ±0.06	0.32 ±0.04	0.62 ±0.05	0.44 ±0.03	0.64 ±0.04	0.48 ±0.02	0.66 ±0.02	0.52 ±0.01
MiniMax-M2	229B	0.61 ±0.06	0.31 ±0.05	0.64 ±0.05	0.43 ±0.03	0.66 ±0.04	0.48 ±0.02	0.68 ±0.02	0.52 ±0.01
Qwen3-Next	80B	0.58 ±0.05	0.3 ±0.04	0.6 ±0.04	0.41 ±0.03	0.61 ±0.03	0.45 ±0.02	0.63 ±0.02	0.49 ±0.01
gpt-oss	120B	0.57 ±0.05	0.23 ±0.05	0.59 ±0.04	0.34 ±0.05	0.6 ±0.03	0.42 ±0.04	0.62 ±0.02	0.49 ±0.01
Qwen3	30B	0.6 ±0.05	0.33 ±0.04	0.62 ±0.04	0.43 ±0.03	0.63 ±0.04	0.47 ±0.02	0.65 ±0.02	0.51 ±0.01
Llama-3.3	70B	0.56 ±0.03	0.16 ±0.03	0.57 ±0.02	0.2 ±0.03	0.57 ±0.02	0.23 ±0.03	0.59 ±0.01	0.27 ±0.02
gemma-3	27B	0.56 ±0.05	0.26 ±0.04	0.57 ±0.04	0.36 ±0.04	0.59 ±0.03	0.42 ±0.03	0.6 ±0.02	0.47 ±0.01
R1-Llama	70B	0.56 ±0.05	0.26 ±0.04	0.58 ±0.04	0.37 ±0.04	0.59 ±0.04	0.42 ±0.02	0.61 ±0.02	0.47 ±0.01
R1-Qwen	32B	0.51 ±0.06	0.2 ±0.04	0.53 ±0.05	0.29 ±0.04	0.55 ±0.04	0.36 ±0.03	0.57 ±0.03	0.41 ±0.01
Chemically specialized LLMs									
ether0	24B	0.55 ±0.07	0.22 ±0.04	0.58 ±0.06	0.31 ±0.04	0.59 ±0.05	0.36 ±0.03	0.62 ±0.03	0.42 ±0.02
ChemDFM-R	14B	0.62 ±0.07	0.31 ±0.04	0.65 ±0.05	0.41 ±0.03	0.66 ±0.05	0.46 ±0.03	0.69 ±0.02	0.5 ±0.02
ChemDFM-v2.0	14B	0.63 ±0.08	0.29 ±0.05	0.66 ±0.06	0.4 ±0.04	0.68 ±0.05	0.45 ±0.03	**0.7** ±0.03	0.5 ±0.02
Models trained on MolRGen									
RL-Molstral-g4	128B	0.59 ±0.05	0.37 ±0.04	0.62 ±0.04	0.47 ±0.02	0.63 ±0.03	0.49 ±0.02	0.65 ±0.02	0.53 ±0.01
RL-Molstral-g8	128B	0.59 ±0.04	0.41 ±0.03	0.61 ±0.04	0.48 ±0.02	0.62 ±0.03	0.51 ±0.01	0.64 ±0.02	0.53 ±0.01
RL-Molstral-g16	128B	0.64 ±0.05	**0.45** ±0.04	0.66 ±0.04	**0.53** ±0.02	0.67 ±0.03	**0.55** ±0.01	0.69 ±0.02	**0.57** ±0.01

between the number of parameters and the top_{k,n_r} score on average, and indeed, Gemma-4 with only 31B parameters outperforms significantly larger models exceeding 100B parameters.

Diversity-aware top-k performances. Figure 3 presents the diversity-aware top-k scores of various LLMs on the de novo molecular generation task, evaluated across a range of similarity thresholds s_{max}. As the similarity threshold increases, the diversity-aware score increases, as expected: high thresholds (e.g., $s_{\text{max}} = 0.9$) impose weak diversity constraints, as two molecules are only considered to be too similar if their Tanimoto similarity is very high. On the contrary, low thresholds (e.g., $s_{\text{max}} \approx 0$) enforce strong constraints where a molecule needs to be very different from all other selected molecules to be considered for the top-k score.

The diversity-aware evaluation provides additional insights compared to the raw top_{k,n_r} score. While Gemma-4 (the best model according to the top-k score) performs the best under most threshold values ($s_{\text{max}} \geq 0.3$), the chemically-specialized ChemDFM-R overtakes it under stronger constraints. This suggests a qualitative difference in generation strategies: while Gemma-4 performs best overall, ChemDFM-R demonstrates superior capacity when generating a set of highly diverse high-reward molecules (low pairwise similarity).

Effect of GRPO training. As explained in subsection 3.3, to show that our dataset and verifier can be leveraged to train LLMs for molecular generation, we fine-tuned Mistral-Small-4 with Group Relative Policy Optimization (GRPO) with 3 different group sizes $n_r \in \{4, 8, 16\}$. As explained in section 3, all docking targets in the test set are excluded from the training set of our models.

Figure 4 compares the performance of RL-Molstral against the base model using the top_{k,n_r} score across varying sampling budgets n_r where n_r is scaled proportionally to $k \in \{2, 4, 8\}$. The models trained with GRPO exhibit a significant improvement in top_{k,n_r} scores at low sampling regimes across all group sizes, and the larger the group size, the higher the improvement.

While the models trained with a group size of 4 and 8 see their improvements over the base model diminish as the sampling budget increases, the model trained with a group size of 16 maintains its performance advantage across all sampling budgets, at the cost of increased compute. For all models this performance improvement also comes at the cost of a decrease in the diversity of the generated molecules, as shown by the distribution of pairwise Tanimoto similarities between molecules generated from the same prompt in Figure 4, with larger group sizes leading to lower

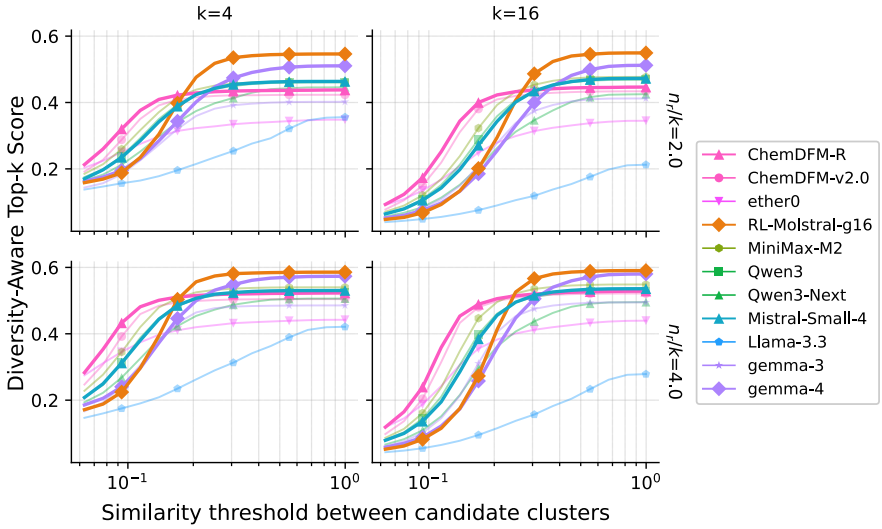


Figure 3: **Diversity-aware top-k score.** Evaluation of the diversity-aware top-k score (y-axis) against varying similarity thresholds (x-axis) between candidate clusters. As the similarity threshold increases, the diversity constraint relaxes, and the average score of the selected molecule pool increases.

diversity scores. This phenomenon aligns with recent findings (Yue et al., 2025), highlighting the following tradeoff: RL improves exploitation but harms exploration.

Despite the decrease in diversity, RL-Molstral-g16 still outperforms the base model for diversity constraints up to $s_{\max} \geq 0.3$, as shown in Figure 3, outperforming all other models in this regime, notably Gemma-4, outperforming this model over all similarity thresholds.

4.2 Comarison between LLMs and REINVENT

Finally, we compare the performances of the various LLMs with traditional models for molecular generation. We chose to compare our model with REINVENT (Loeffler et al., 2024; Xu et al., 2024), a commonly used model for multi-property molecular generation. REINVENT consists in training an autoregressive model with reinforcement learning for a specific optimization task. As such, one REINVENT model is trained for each prompt it is evaluated on (contrary to the zero-shot approach of the LLMs we are evaluating), making the comparison particularly computationally expensive, and we hence only report the results for the optimization of 100 tasks from our test set, and selected the best hyperparameters across all tasks.

More details and discussion on the comparison with domain-specific baselines can be found in subsection E.3.

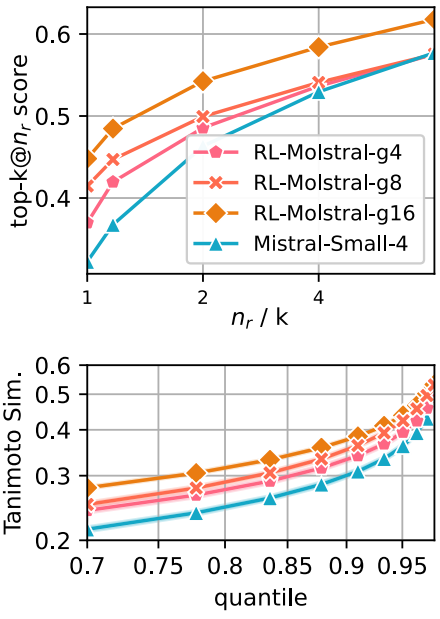


Figure 4: **GRPO Training Effect. (Top)** top_{k,n_r} scores across varying sampling scales n_r/k. **(Bottom)** Distribution of pairwise Tanimoto similarities for molecules generated for the same prompt.

The results presented in Figure 5 demonstrate that LLMs significantly underperform compared to this domain-specific baseline. It therefore seems that there is still a significant gap to bridge between LLMs and traditional domain-specific models for molecular generation, and that LLMs are not yet competitive with traditional baselines for this task.

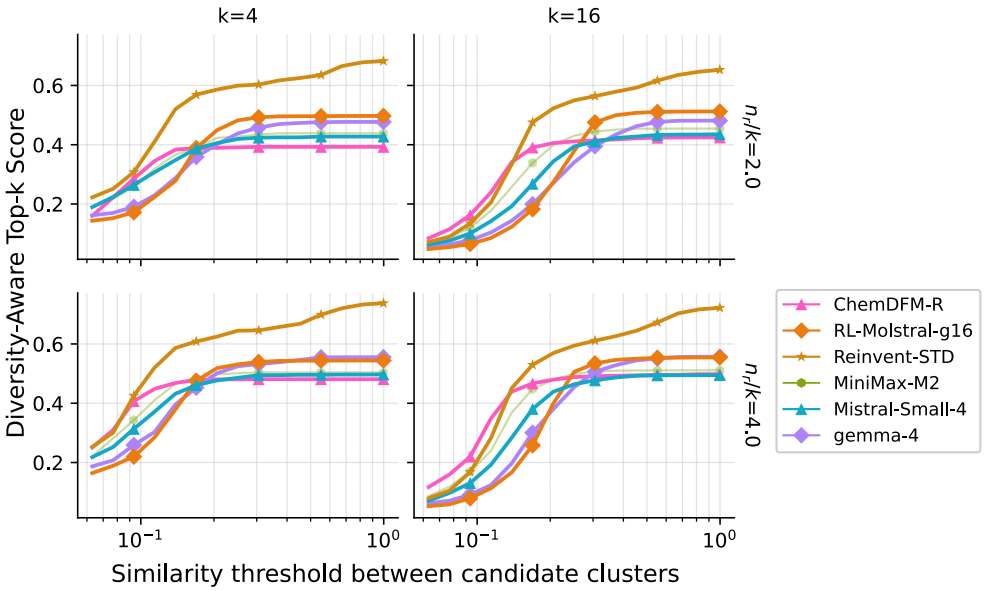


Figure 5: **Diversity-Aware scores with Reinvent.** We compare the performance of LLMs to a traditional generation algorithm baseline: REINVENT. REINVENT consistently outperforms the LLMs, highlighting the gap between LLMs and traditional generation algorithms for molecular generation.

5 Summary and Concluding Remarks

In this work, we introduced MolRGen, an open-source benchmark and verifier for training and evaluating reasoning-based LLMs on *de novo* molecular generation, covering approximately 4,500 protein targets and molecular property prediction tasks. Our evaluation of 13 open-source LLMs reveals that relying solely on the raw top_{k,n_r} score to compare models does not capture the full picture of their performance, and while Gemma-4 achieves the best top_{k,n_r} scores among pretrained LLMs, it is outperformed by chemically specialized models under strict diversity constraints.

Fine-tuning Mistral-Small-4 with GRPO demonstrates that MolRGen can serve as an effective training signal: RL-Molstral consistently outperforms the base model in terms of top_{k,n_r} scores and with mild diversity constraints when evaluating the diversity-aware top-k score. However, RL training consistently reduces chemical diversity, motivating the development and evaluation of RL training strategies that explicitly balance exploitation and exploration.

Limitations and future work. Despite our efforts to design a realistic and informative benchmark, there are still several limitations that could be explored in future work. First, our benchmark relies on docking scores as a proxy for binding affinity, which is a common practice to evaluate molecular generation methods, which are known to be imperfect predictors of experimental binding affinities. Additionally, the LLMs in our experiments receive limited information about target protein structures: only textual descriptions, and they do not have direct access to 3D structural coordinates, binding pocket geometries, or other detailed biophysical information. Providing richer structural context (e.g., via encoded 3D representations or direct access to the PDB files) could substantially improve model performance and is an important direction for future work.

We hope MolRGen will serve as a starting point for addressing these limitations and advancing the development of reasoning-based LLMs for molecular generation, to potentially help accelerate pre-clinical drug discovery pipelines.

6 ETHICS STATEMENT - REPRODUCIBILITY STATEMENT

With this work, we release MOLRGEN, a comprehensive dataset designed to advance scientific research in the field of molecular generation and drug discovery. We are committed to ensuring that this resource is used responsibly and ethically. Accelerated molecular generation has the potential to significantly impact drug discovery and development. MOLRGEN is intended for research purposes only, and not for any malicious or harmful activities, and we will monitor its usage to ensure that it is not misused.

Our work releases a large-scale dataset of tasks for LLMs in molecular modeling, which will be accessible through HuggingFace and GitHub, with detailed documentation and usage guidelines. Through this code repository, researchers will be able to access our verifiers which are fully disclosed, allowing any researcher to reproduce our results or continue the research in this field. All our methodology for processing our dataset, training our models, and the implementation of our verifiers are available in the supplementary material and the documentation of our code repository.

7 AI DISCLOSURE

In this work, we used generative AI tools for writing (aid or polish), edit a research paper to improve readability, format references, assist with translation. We have also used generative AI tools for code generation, and code reviews.

We have not used generative AI tools to generate synthetic data sets, help develop theoretical models or conceptual frameworks, design or provide feedback on research methodology or experiments, implement methods, clean and reformat dataset, support qualitative and thematic data analysis, interpret results, and formulating mathematical claims, providing critical ingredients for proving mathematical claims, assisting in the writing of proofs, proposing or refining hypotheses are not applicable to this work.

We have reviewed all AI-assisted work. For the help on writing the paper, we have reviewed and edited the proposed text to fit our tone. For the code generation and review, we systematically reviewed all propositions, tested the code (using notably, but not limited to, unit tests, and integration tests), and verified that the code is correct and does not contain any security vulnerabilities.

REFERENCES

G. Richard Bickerton, Gaia V. Paolini, Jérémy Besnard, Sorel Muresan, and Andrew L. Hopkins. Quantifying the chemical beauty of drugs. *Nature Chemistry*, 4(2):90–98, January 2012. ISSN 1755-4349. doi: 10.1038/nchem.1243.

DeepSeek-AI. Deepseek-r1: Incentivizing reasoning capability in llms via reinforcement learning, 2025. URL https://arxiv.org/abs/2501.12948.

Vishal Dey, Xiao Hu, and Xia Ning. Gellmo: Generalizing large language models for multi-property molecule optimization, 2025. URL https://arxiv.org/abs/2502.13398.

Discngine. fpocket. URL https://github.com/Discngine/fpocket. GitHub repository.

Peter Ertl and Ansgar Schuffenhauer. Estimation of synthetic accessibility score of drug-like molecules based on molecular complexity and fragment contributions. *Journal of Cheminformatics*, 1:8, June 2009. ISSN 1758-2946. doi: 10.1186/1758-2946-1-8. URL https://pmc.ncbi.nlm.nih.gov/articles/PMC3225829/.

Adibvafa Fallahpour, Andrew Magnuson, Purav Gupta, Shihao Ma, Jack Naimer, Arnav Shah, Haonan Duan, Omar Ibrahim, Hani Goodarzi, Chris J. Maddison, and Bo Wang. Bioreason: Incentivizing multimodal biological reasoning within a dna-llm model, 2025. URL https://arxiv.org/abs/2505.23579.

Yin Fang, Xiaozhuan Liang, Ningyu Zhang, Kangwei Liu, Rui Huang, Zhuo Chen, Xiaohui Fan, and Huajun Chen. Mol-instructions: A large-scale biomolecular instruction dataset for large language models, 2024. URL https://arxiv.org/abs/2306.08018.

Wenhao Gao, Shitong Luo, and Connor W. Coley. Generative artificial intelligence for navigating synthesizable chemical space, 2024. URL https://arxiv.org/abs/2410.03494.

Haisong Gong, Qiang Liu, Shu Wu, and Liang Wang. Text-guided molecule generation with diffusion language model, 2024. URL https://arxiv.org/abs/2402.13040.

Jiaqi Guan, Wesley Wei Qian, Xingang Peng, Yufeng Su, Jian Peng, and Jianzhu Ma. 3d equivariant diffusion for target-aware molecule generation and affinity prediction. In *International Conference on Learning Representations*, 2023.

Rafael Gómez-Bombarelli, Jennifer N. Wei, David Duvenaud, José Miguel Hernández-Lobato, Benjamín Sánchez-Lengeling, Dennis Sheberla, Jorge Aguilera-Iparraguirre, Timothy D. Hirzel, Ryan P. Adams, and Alán Aspuru-Guzik. Automatic Chemical Design Using a Data-Driven Continuous Representation of Molecules. *ACS Central Science*, 4(2):268–276, February 2018. ISSN 2374-7943. doi: 10.1021/acscentsci.7b00572. URL https://doi.org/10.1021/acscentsci.7b00572. Publisher: American Chemical Society.

Ragy Haddad, Eleni E. Litsa, Zhen Liu, Xin Yu, Daniel Burkhardt, and Govinda Bhisetti. Targeted molecular generation with latent reinforcement learning. *Scientific Reports*, 15(1):15202, April 2025. ISSN 2045-2322. doi: 10.1038/s41598-025-99785-0. URL https://www.nature.com/articles/s41598-025-99785-0. Publisher: Nature Publishing Group.

Nafisa M. Hassan, Amr A. Alhossary, Yuguang Mu, and Chee-Keong Kwoh. Protein-Ligand Blind Docking Using QuickVina-W With Inter-Process Spatio-Temporal Integration. *Scientific Reports*, 7(1):15451, November 2017. ISSN 2045-2322. doi: 10.1038/s41598-017-15571-7. URL https://www.nature.com/articles/s41598-017-15571-7. Publisher: Nature Publishing Group.

Zhongmou He, Yee Man Choi, Kexun Zhang, Ivan Bercovich, Jiabao Ji, Junting Zhou, Dejia Xu, Aidan Zhang, Yixiao Zeng, and Lei Li. Hardtestgen: A high-quality rl verifier generation pipeline for llm algorithimic coding. In C. Vondrick, B. Hariharan, C. Raffel, L. Pinto, D. Yang, and A. Faust (eds.), *International Conference on Learning Representations*, volume 2026, pp. 35671–35715, 2026. URL https://proceedings.iclr.cc/paper_files/paper/2026/file/3c11cea35aa2e4797b2f239da378927b-Paper-Conference.pdf.

Yanwen Huang, Bowen Gao, Yinjun Jia, Hongbo Ma, Wei-Ying Ma, Ya-Qin Zhang, and Yanyan Lan. Siu: A million-scale structural small molecule-protein interaction dataset for unbiased bioactivity prediction, 2024. URL https://arxiv.org/abs/2406.08961.

Dongki Kim, Wonbin Lee, and Sung Ju Hwang. Mol-llama: Towards general understanding of molecules in large molecular language model, 2025. URL https://arxiv.org/abs/2502.13449.

Michał Koziarski, Andrei Rekesh, Dmytro Shevchuk, Almer van der Sloot, Piotr Gaiński, Yoshua Bengio, Cheng-Hao Liu, Mike Tyers, and Robert A. Batey. Rgfn: Synthesizable molecular generation using gflownets, 2024. URL https://arxiv.org/abs/2406.08506.

Greg Landrum, Paolo Tosco, Brian Kelley, sriniker, gedeck, NadineSchneider, Riccardo Vianello, Ric, Andrew Dalke, Brian Cole, AlexanderSavelyev, Matt Swain, Samo Turk, Dan N, Alain Vaucher, Eisuke Kawashima, Maciej Wójcikowski, Daniel Probst, guillaume godin, David Cosgrove, Axel Pahl, JP, Francois Berenger, strets123, JLVarjo, Noel O'Boyle, Patrick Fuller, Jan Holst Jensen, Gianluca Sforna, and DoliathGavid. rdkit/rdkit: 2020_03_1 (q1 2020) release, March 2020. URL https://doi.org/10.5281/zenodo.3732262.

Seul Lee, Karsten Kreis, Srimukh Prasad Veccham, Meng Liu, Danny Reidenbach, Saee Paliwal, Weili Nie, and Arash Vahdat. Rethinking molecule synthesizability with chain-of-reaction, 2025. URL https://arxiv.org/abs/2509.16084.

Pablo Lemos, Zane Beckwith, Sasaank Bandi, Maarten van Damme, Jordan Crivelli-Decker, Benjamin J. Shields, Thomas Merth, Punit K. Jha, Nicola De Mitri, Tiffany J. Callahan, AJ Nish, Paul Abruzzo, Romelia Salomon-Ferrer, and Martin Ganahl. Sair: Enabling deep learning for protein-ligand interactions with a synthetic structural dataset. 2025. doi: 10.1101/2025.06.17.660168.

Jiatong Li, Weida Wang, Qinggang Zhang, Junxian Li, Di Zhang, Changmeng Zheng, Shufei Zhang, Xiaoyong Wei, and Qing Li. Mol-r1: Towards explicit long-cot reasoning in molecule discovery, 2025. URL https://arxiv.org/abs/2508.08401.

Sihang Li, Zhiyuan Liu, Yanchen Luo, Xiang Wang, Xiangnan He, Kenji Kawaguchi, Tat-Seng Chua, and Qi Tian. Towards 3d molecule-text interpretation in language models, 2024. URL https://arxiv.org/abs/2401.13923.

Hannes H. Loeffler, Jiazhen He, Alessandro Tibo, Jon Paul Janet, Alexey Voronov, Lewis H. Mervin, and Ola Engkvist. Reinvent 4: Modern AI–driven generative molecule design. *Journal of Cheminformatics*, 16(1):20, February 2024. ISSN 1758-2946. doi: 10.1186/s13321-024-00812-5. URL https://doi.org/10.1186/s13321-024-00812-5.

Sohvi Luukkonen, Helle W. van den Maagdenberg, Michael T.M. Emmerich, and Gerard J.P. van Westen. Artificial intelligence in multi-objective drug design. *Current Opinion in Structural Biology*, 79:102537, 2023. ISSN 0959-440X. doi: https://doi.org/10.1016/j.sbi.2023.102537. URL https://www.sciencedirect.com/science/article/pii/S0959440X23000118.

MiniMax. Minimax-m1: Scaling test-time compute efficiently with lightning attention, 2025. URL https://arxiv.org/abs/2506.13585.

Adrian Mirza, Nawaf Alampara, Sreekanth Kunchapu, Martiño Ríos-García, Benedict Emoekabu, Aswanth Krishnan, Tanya Gupta, Mara Schilling-Wilhelmi, Macjonathan Okereke, Anagha Aneesh, Amir Mohammad Elahi, Mehrdad Asgari, Juliane Eberhardt, Hani M. Elbeheiry, María Victoria Gil, Maximilian Greiner, Caroline T. Holick, Christina Glaubitz, Tim Hoffmann, Abdelrahman Ibrahim, Lea C. Klepsch, Yannik Köster, Fabian Alexander Kreth, Jakob Meyer, Santiago Miret, Jan Matthias Peschel, Michael Ringleb, Nicole Roesner, Johanna Schreiber, Ulrich S. Schubert, Leanne M. Stafast, Dinga Wonanke, Michael Pieler, Philippe Schwaller, and Kevin Maik Jablonka. Are large language models superhuman chemists?, 2024. URL https://arxiv.org/abs/2404.01475.

Mistral-AI, :, Abhinav Rastogi, Albert Q. Jiang, Andy Lo, Gabrielle Berrada, Guillaume Lample, Jason Rute, Joep Barmentlo, Karmesh Yadav, Kartik Khandelwal, Khyathi Raghavi Chandu, Léonard Blier, Lucile Saulnier, Matthieu Dinot, Maxime Darrin, Neha Gupta, Roman Soletskyi, Sagar Vaze, Teven Le Scao, Yihan Wang, Adam Yang, Alexander H. Liu, Alexandre Sablayrolles, Amélie Héliou, Amélie Martin, Andy Ehrenberg, Anmol Agarwal, Antoine Roux, Arthur Darcet, Arthur Mensch, Baptiste Bout, Baptiste Rozière, Baudouin De Monicault, Chris Bamford, Christian Wallenwein, Christophe Renaudin, Clémence Lanfranchi, Darius Dabert, Devon Mizelle, Diego de las Casas, Elliot Chane-Sane, Emilien Fugier, Emma Bou Hanna, Gauthier Delerce, Gauthier Guinet, Georgii Novikov, Guillaume Martin, Himanshu Jaju, Jan Ludziejewski, Jean-Hadrien Chabran, Jean-Malo Delignon, Joachim Studnia, Jonas Amar, Josselin Somerville Roberts, Julien Denize, Karan Saxena, Kush Jain, Lingxiao Zhao, Louis Martin, Luyu Gao, Lélio Renard Lavaud, Marie Pellat, Mathilde Guillaumin, Mathis Felardos, Maximilian Augustin, Mickaël Seznec, Nikhil Raghuraman, Olivier Duchenne, Patricia Wang, Patrick von Platen, Patryk Saffer, Paul Jacob, Paul Wambergue, Paula Kurylowicz, Pavankumar Reddy Muddireddy, Philomène Chagniot, Pierre Stock, Pravesh Agrawal, Romain Sauvestre, Rémi Delacourt, Sanchit Gandhi, Sandeep Subramanian, Shashwat Dalal, Siddharth Gandhi, Soham Ghosh, Srijan Mishra, Sumukh Aithal, Szymon Antoniak, Thibault Schueller, Thibaut Lavril, Thomas Robert, Thomas Wang, Timothée Lacroix, Valeriia Nemychnikova, Victor Paltz, Virgile Richard, Wen-Ding Li, William Marshall, Xuanyu Zhang, and Yunhao Tang. Magistral, 2025. URL https://arxiv.org/abs/2506.10910.

Philipp Moritz, Robert Nishihara, Stephanie Wang, Alexey Tumanov, Richard Liaw, Eric Liang, William Paul, Michael I. Jordan, and Ion Stoica. Ray: A distributed framework for emerging AI applications. *CoRR*, abs/1712.05889, 2017. URL http://arxiv.org/abs/1712.05889.

Siddharth M. Narayanan, James D. Braza, Ryan-Rhys Griffiths, Albert Bou, Geemi Wellawatte, Mayk Caldas Ramos, Ludovico Mitchener, Samuel G. Rodriques, and Andrew D. White. Training a scientific reasoning model for chemistry, 2025. URL https://arxiv.org/abs/2506.17238.

OpenAI. gpt-oss-120b gpt-oss-20b model card, 2025. URL https://arxiv.org/abs/2508.10925.

Diogo Santos-Martins, Leonardo Solis-Vasquez, Andreas F Tillack, Michel F Sanner, Andreas Koch, and Stefano Forli. Accelerating AutoDock4 with GPUs and Gradient-Based Local Search. *Journal of Chemical Theory and Computation*, 17(2):1060–1073, February 2021. ISSN 1549-9618. doi: 10.1021/acs.jctc.0c01006. URL https://doi.org/10.1021/acs.jctc.0c01006. Publisher: American Chemical Society.

Diogo Santos-Martins, Yiran He, Jerome Eberhardt, Parnika Sharma, Niccolò Bruciaferri, Matthew Holcomb, Manuel A. Llanos, Althea Hansel-Harris, Allison Pearl Barkdull, Andreas Frank Tillack, Giulia Bianco, May-Linn Paulsen, Joani Mato, Ishan Taneja, and Stefano Forli. Meeko: Molecule Parametrization and Software Interoperability for Docking and Beyond. *Journal of Chemical Information and Modeling*, 65(24):13045–13050, December 2025. ISSN 1549-9596. doi: 10.1021/acs.jcim.5c02271. URL https://doi.org/10.1021/acs.jcim.5c02271. Publisher: American Chemical Society.

Zhihong Shao, Peiyi Wang, Qihao Zhu, Runxin Xu, Junxiao Song, Xiao Bi, Haowei Zhang, Mingchuan Zhang, Y. K. Li, Y. Wu, and Daya Guo. Deepseekmath: Pushing the limits of mathematical reasoning in open language models, 2024. URL https://arxiv.org/abs/2402.03300.

Gemma Team. Gemma 3. 2025a. URL https://goo.gle/Gemma3Report.

Gemma Team. Gemma 4. 2026. URL https://ai.google.dev/gemma/docs/core/model_card_4.

Llama Team. The llama 3 herd of models, 2024. URL https://arxiv.org/abs/2407.21783.

Qwen Team. Qwen3 technical report, 2025b. URL https://arxiv.org/abs/2505.09388.

Austin Tripp and José Miguel Hernández-Lobato. Genetic algorithms are strong baselines for molecule generation, 2023. URL https://arxiv.org/abs/2310.09267.

Marcel L. Verdonk, Jason C. Cole, Michael J. Hartshorn, Christopher W. Murray, and Richard D. Taylor. Improved protein–ligand docking using GOLD. *Proteins: Structure, Function, and Bioinformatics*, 52(4):609–623, 2003. ISSN 1097-0134. doi: 10.1002/prot.10465. URL https://onlinelibrary.wiley.com/doi/abs/10.1002/prot.10465. _eprint: https://onlinelibrary.wiley.com/doi/pdf/10.1002/prot.10465.

Cas Wognum, Jeremy R. Ash, Matteo Aldeghi, Raquel Rodríguez-Pérez, Cheng Fang, Alan C. Cheng, Daniel J. Price, Djork-Arné Clevert, Ola Engkvist, and W. Patrick Walters. A call for an industry-led initiative to critically assess machine learning for real-world drug discovery. *Nature Machine Intelligence*, 6(10):1120–1121, October 2024. ISSN 2522-5839. doi: 10.1038/s42256-024-00911-w. URL https://doi.org/10.1038/s42256-024-00911-w.

Jeremy Wohlwend, Gabriele Corso, Saro Passaro, Mateo Reveiz, Ken Leidal, Wojtek Swiderski, Tally Portnoi, Itamar Chinn, Jacob Silterra, Tommi Jaakkola, and Regina Barzilay. Boltz-1 democratizing biomolecular interaction modeling. *bioRxiv*, 2024. doi: 10.1101/2024.11.19.624167. URL https://www.biorxiv.org/content/early/2024/11/20/2024.11.19.624167.

Pengcheng Xu, Tao Feng, Tianfan Fu, Siddhartha Laghuvarapu, and Jimeng Sun. Molecular de novo design through transformer-based reinforcement learning, 2024. URL https://arxiv.org/abs/2310.05365.

Botao Yu, Frazier N. Baker, Ziqi Chen, Xia Ning, and Huan Sun. Llasmol: Advancing large language models for chemistry with a large-scale, comprehensive, high-quality instruction tuning dataset, 2024. URL https://arxiv.org/abs/2402.09391.

Yang Yue, Zhiqi Chen, Rui Lu, Andrew Zhao, Zhaokai Wang, Yang Yue, Shiji Song, and Gao Huang. Does reinforcement learning really incentivize reasoning capacity in llms beyond the base model? *arXiv preprint arXiv:2504.13837*, 2025.

Guibin Zhang, Hejia Geng, Xiaohang Yu, Zhenfei Yin, Zaibin Zhang, Zelin Tan, Heng Zhou, Zhongzhi Li, Xiangyuan Xue, Yijiang Li, Yifan Zhou, Yang Chen, Chen Zhang, Yutao Fan, Zihu Wang, Songtao Huang, Francisco Piedrahita-Velez, Yue Liao, Hongru Wang, Mengyue Yang, Heng Ji, Jun Wang, Shuicheng Yan, Philip Torr, and Lei Bai. The landscape of agentic reinforcement learning for llms: A survey, 2026. URL https://arxiv.org/abs/2509.02547.

Zihan Zhao, Da Ma, Lu Chen, Liangtai Sun, Zihao Li, Yi Xia, Bo Chen, Hongshen Xu, Zichen Zhu, Su Zhu, Shuai Fan, Guodong Shen, Kai Yu, and Xin Chen. Developing chemdfm as a large language foundation model for chemistry. *Cell Reports Physical Science*, 6(4):102523, April 2025. ISSN 2666-3864. doi: 10.1016/j.xcrp.2025.102523. URL http://dx.doi.org/10.1016/j.xcrp.2025.102523.

Artem Zholus, Maksim Kuznetsov, Roman Schutski, Rim Shayakhmetov, Daniil Polykovskiy, Sarath Chandar, and Alex Zhavoronkov. BindGPT: A Scalable Framework for 3D Molecular Design via Language Modeling and Reinforcement Learning. *Proceedings of the AAAI Conference on Artificial Intelligence*, 39(24):26083–26091, April 2025. ISSN 2374-3468. doi: 10.1609/aaai.v39i24.34804. URL https://ojs.aaai.org/index.php/AAAI/article/view/34804.

Zhijian Zhou, Junyi An, Zongkai Liu, Yunfei Shi, Xuan Zhang, Fenglei Cao, Chao Qu, and Yuan Qi. Guiding diffusion models with reinforcement learning for stable molecule generation, 2025. URL https://arxiv.org/abs/2508.16521.

Jiaxi Zhuang, Yaorui Shi, Jue Hou, Yunong He, Mingwei Ye, Mingjun Xu, Yuming Su, Linfeng Zhang, Ying Qian, Linfeng Zhang, Guolin Ke, and Hengxing Cai. Reasoning-enhanced large language models for molecular property prediction, 2025. URL https://arxiv.org/abs/2510.10248.