CHAIN-OF-MOLECULES: AGENTIC REASONING IN CHEMICAL SPACE FOR CONSTRAINED MOLECULAR OPTIMIZATION

Anonymous authors
Paper under double-blind review

ABSTRACT

Chain-of-thought reasoning has enabled language models to solve complex problems through explicit intermediate steps. We ask whether this paradigm can be transferred from language into chemical space. We introduce *Chain-of-Molecules* (CoM), a molecular-generation paradigm in which intermediate reasoning states are molecules and transitions between them are explicit chemical edits. In Agentic CoM, a symbolic environment executes valid edits, rejects infeasible ones, and returns the resulting molecule, allowing the model to explore alternatives, backtrack, and adapt its subsequent actions. We evaluate CoM in two complementary settings. First, a controlled property-targeting benchmark compares four generation paradigms across in-distribution, interpolation, and extrapolation regimes, disentangling the contributions of intermediate molecular states and interleaved environment execution. Second, we ask which generation paradigm yields the most adaptable base model for molecular optimization: starting directly from the respective pretrained base models, we use reinforcement learning to adapt them to three downstream bioactivity objectives without task-specific supervised finetuning. Agentic CoM performs strongest across both settings. Together, these results support explicit molecular trajectories and environment feedback not only as an effective optimization paradigm, but as a basis for reusable pretrained molecular optimizers. Code, data, and model checkpoints are publicly available.

1 INTRODUCTION

Step-by-step reasoning enables language models to solve difficult problems by turning a single prediction into a structured search process. Intermediate traces support decomposition, exploration of alternative solutions, recovery from dead ends, backtracking, and verification before a final answer is produced (Wei et al., 2022; Nye et al., 2021; Gandhi et al., 2025). Recent reasoning models further show that such behavior can be acquired through supervised and reinforcement-learning (RL)-based post-training rather than elicited only through prompting (Guo et al., 2025).

These reasoning dynamics are not inherently tied to natural language. Language provides a convenient representation of intermediate states, but the underlying computation only requires a space in which partial solutions are meaningful and a set of operations for moving between them. This perspective has motivated extensions of step-by-step generation beyond text, including progressive visual refinement in image generation (Zhang et al., 2025) and explicit scene plans or keyframes in video understanding (Fei et al., 2024).

Molecular design provides a natural setting for reasoning outside language. Molecules can be modified incrementally, alternative structures can be explored, unsuccessful modifications can be reversed, and promising candidates can be refined over several steps (Westhead et al., 1995). Most molecular language models, however, generate a complete molecular representation directly, without exposing or controlling the search process that leads to it (Tang et al., 2024; Özçelik et al., 2025).

This observation motivates representing molecular generation as an explicit sequence of task-directed transformations, with molecules as intermediate reasoning states and chemical edits connecting them. Such a trajectory makes the proposed search explicit providing a structured trajectory through chemical space, but a self-generated sequence still requires the model to predict both the transformations

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.

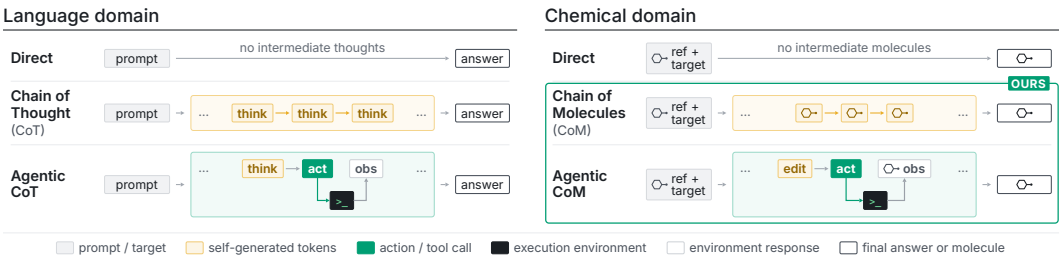

Figure 1: **Reasoning paradigms in language and chemical space.** Direct generation contains only the final output. Self-generated reasoning adds intermediate states; agentic reasoning interleaves actions with environment execution and observations.

and their resulting molecular states. These predictions need not be chemically valid or consistent with the stated edits. Recent advances in language models integrate tool execution into reasoning, allowing subsequent decisions to draw on the observed consequences of earlier actions (Yao et al., 2023; Team et al., 2026; Liu et al., 2025). We extend this principle to *agentic molecular reasoning*: a model searches chemical space by choosing transformations, observing their execution in a chemical environment, and deciding how to proceed. The resulting search can include among others continued editing, exploration of alternative branches, and backtracking.

We introduce *Chain-of-Molecules* (CoM), a representation whose intermediate reasoning states are molecules connected by edits. In *self-generated CoM*, the model predicts both the edits and the resulting molecular states. *Agentic CoM*, our instantiation of agentic molecular reasoning, instead constructs the trajectory through interaction with a chemical environment. We implement these paradigms with autoregressive molecular models and study both supervised and RL-based task adaptation. Figure 1 contrasts direct generation, self-generated intermediate reasoning, and agentic reasoning in language and chemical space.

We evaluate CoM in two complementary settings. First, we use controlled property targeting across five deterministic molecular objectives to compare four generation paradigms along two design choices: whether trajectories contain intermediate molecular states and whether proposed edits are executed by a chemical environment or the model itself. Evaluation across in-distribution, interpolation gap, and extrapolation regimes provides a controlled setting for isolating the contributions of explicit molecular trajectories and interleaved environment execution. Second, we investigate whether the same pretrained models provide useful starting points for downstream bioactivity optimization. Rather than constructing task-specific supervised trajectories, we start directly from the respective base models and adapt them through RL to three bioactivity objectives. This tests adaptation to new molecular objectives without task-specific supervised trajectories. Together, these experiments test Agentic CoM both as a molecular reasoning paradigm and as a basis for reusable pretrained molecular optimizers.

2 Problem Setting and Generation Paradigms

2.1 Constrained Molecular Optimization

In the following, we consider autoregressive SMILES (Weininger, 1988) generators, as established by Segler et al. (2018). We formulate constrained molecular optimization as a conditional molecular generation task. Given a reference molecule and a molecular objective, the model generates a new molecule that satisfies the objective while remaining within a specified structural-distance interval from the reference. The objective is either to match a requested property value or to maximize a predicted activity. We call the setting *optimization* because the generated molecule is obtained under property and structural objectives relative to a given starting molecule.

Let $\mathcal{M}$ denote the set of valid molecules, let $\mathcal{J}$ denote the set of considered molecular objectives, and let $\{f_j : \mathcal{M} \to \mathbb{R}\}_{j\in\mathcal{J}}$ denote the set of considered molecular-property functions, including calculated descriptors and predicted activities. A task is specified by $\mathcal{T} = \big(m_{\mathrm{ref}}, j, v^\star, \ell, h\big)$, where $m_{\mathrm{ref}} \in \mathcal{M}$ is the reference molecule, $j \in \mathcal{J}$ identifies the molecular objective, $v^\star \in \mathbb{R}$ is the requested objective value, and $[\ell, h]$, with $0 \le \ell \le h$, is the permitted structural-distance interval.

Let $d_{\mathrm{GED}} : \mathcal{M} \times \mathcal{M} \to \mathbb{N}_0$ denote the graph edit distance; the structural success component s_{struct} is then given by $s_{\mathrm{struct}}(\hat{m}; \mathcal{T}) = \mathbb{I}[\ell \le d_{\mathrm{GED}}(m_{\mathrm{ref}}, \hat{m}) \le h]$. For a valid generated molecule $\hat{m} \in \mathcal{M}$, the property component of success is

$$s_{\mathrm{prop}}(\hat{m}; \mathcal{T}) = \begin{cases} \mathbb{I}[|f_j(\hat{m}) - v^\star| \le \varepsilon_j], & \text{property targeting,} \\ \mathbb{I}[f_j(\hat{m}) \ge \kappa], & \text{bioactivity.} \end{cases} \tag{1}$$

Here $\mathbb{I}[\cdot]$ denotes the indicator function, $\varepsilon_j > 0$ is the property-specific tolerance, and κ is the chosen bioactivity evaluation threshold. The joint-success indicator combines the structural and property success components:

$$s_{\mathrm{joint}}(\hat{m}; \mathcal{T}) = s_{\mathrm{prop}}(\hat{m}; \mathcal{T}) \cdot s_{\mathrm{struct}}(\hat{m}; \mathcal{T}). \tag{2}$$

An output is jointly successful when $s_{\mathrm{joint}} = 1$. Outputs without a valid committed molecule, denoted $\perp$, receive zero for all three indicators. The upper distance bound limits structural change from the reference; a positive lower bound requires a nontrivial modification.

2.2 Direct molecular generation

Let $\mathcal{V}$ denote the token vocabulary and $\mathbf{x}_{\mathcal{T}}$ the serialized, tokenized prompt for task $\mathcal{T}$. A direct molecular generator produces a token sequence $\mathbf{y} = (y_1, \ldots, y_L) \in \mathcal{V}^L$ of length L according to

$$p_\theta(\mathbf{y} \mid \mathbf{x}_{\mathcal{T}}) = \prod_{i=1}^{L} p_\theta(y_i \mid \mathbf{x}_{\mathcal{T}}, y_{<i}). \tag{3}$$

A decoder $\mathrm{dec} : \mathcal{V}^* \to \mathcal{M} \cup \{\perp\}$ maps the generated sequence to a molecule, where $\perp$ denotes an invalid representation. In our experiments, $\mathbf{y}$ is a SMILES sequence and $\hat{m} = \mathrm{dec}(\mathbf{y})$.

A direct molecular generator produces the final molecular representation without exposing an explicit reasoning trajectory of intermediate molecular states, molecular-edit actions, or environment observations. At inference time, validity, property success, and structural success can be evaluated only after the complete SMILES sequence has been generated and decoded. This is our first paradigm, which we refer to as *direct SMILES*.

2.3 Structured molecular reasoning

Structured molecular reasoning produces an explicit trajectory through chemical space rather than only a final molecular string. We write the trajectory as

$$\boldsymbol{\tau} = (m_0, a_1, m_1, \ldots, a_T, m_T), \qquad m_0 = m_{\mathrm{ref}}, \tag{4}$$

where T is the number of trajectory decisions, $m_t \in \mathcal{M}$ is a molecular state, and $a_t \in \mathcal{A}$ records an edit or control decision. We write $\boldsymbol{\tau}_{<t}$ for the trajectory prefix ending at m_{t-1}. In the agentic formats, a_t includes the arguments needed for execution; in self-generated CoM, it is an action label accompanying the self-predicted molecular state. This representation specifies the contents of an explicit molecular reasoning trajectory, but not how its intermediate molecular states and observations are produced. They may be generated by the model or supplied through interleaved interaction with an external environment; we distinguish these cases in Section 2.4.

The action space $\mathcal{A}$ contains molecular-edit and control actions. The four molecular-edit actions are add, delete, replace, and bridge. For agentic execution, their arguments specify the application sites and fragments as required. The control actions allow the model to backtrack to an earlier molecular state or commit a molecule as its final output. At step t, the model generates an action according to $a_t \sim p_\theta(\cdot \mid \mathbf{x}_{\mathcal{T}}, \boldsymbol{\tau}_{<t})$. The trajectory terminates when the model issues a commit action, and the selected output molecule is denoted by $\hat{m} = \mathrm{commit}(\boldsymbol{\tau})$. The actions are illustrated in Figure 2a; the action grammar and interaction protocol are described in Appendices B.2 and B.3.

2.4 Agentic molecular reasoning

A structured molecular trajectory may be produced in two different ways. In *non-agentic molecular reasoning*, the model predicts action labels and intermediate molecular states in a single uninterrupted output:

$$(a_t, m_t) \sim p_\theta(\cdot \mid \mathbf{x}_{\mathcal{T}}, \boldsymbol{\tau}_{<t}). \tag{5}$$

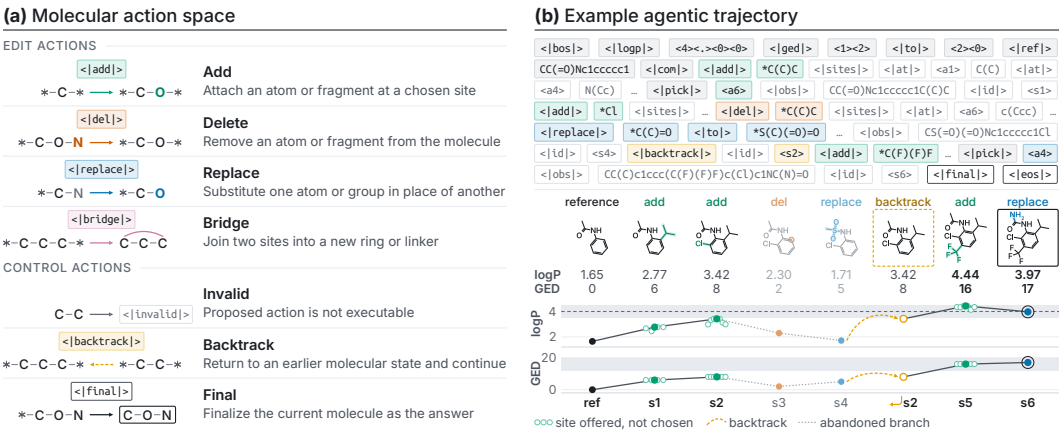

Figure 2: **(a)** Molecular-edit and search-control actions. **(b)** Example optimization trajectory with exploration, backtracking, and commit steps.

The action labels do not specify executable fragment arguments. Because transitions are predicted rather than executed by an environment, the change from m_{t-1} to m_t need not match the stated action a_t. This is our second paradigm, which we refer to as *self-generated CoM*.

In *agentic molecular reasoning*, the model generates an action, while an external environment supplies the corresponding molecular state before the next action is generated:

$$a_t \sim p_\theta\big(\cdot \mid \mathbf{x}_{\mathcal{T}}, \boldsymbol{\tau}_{<t}\big), \qquad m_t = \mathcal{E}\big(\boldsymbol{\tau}_{<t}, a_t\big). \tag{6}$$

The next model decision therefore conditions on the environment-returned consequences of the preceding action. This interleaving of model generation and environment execution forms a closed feedback loop that prevents hallucinated molecular states. *Agentic CoM*, our neuro-symbolic realization of molecular reasoning, is our third paradigm. A learned neural model selects molecular edits, while an RDKit-based symbolic environment validates and executes them. When needed, the environment enumerates legal application sites and returns the resulting molecular state or an error observation. This yields an inspectable, interruptible, and verifiable trajectory in which the final molecule is causally determined by the executed sequence of actions. Our fourth paradigm, *agentic edit program*, is an agentic counterpart to direct prediction: it specifies the answer through a short executable program with a maximum two-edit budget, rather than an extended molecular reasoning trajectory. As in Agentic CoM, the model receives site-selection menus and the environment executes each selected edit.

2.5 Evaluation metrics

We report the property component s_{prop} as *property success* for property targeting and *activity hit* for bioactivity. These test target matching and a predicted-activity threshold, respectively (Equation (1)). *Structural success* tests the requested GED interval, and *joint success* requires both components (Equation (2)). Success rates average the corresponding indicators over all sampled outputs, including missing or invalid outputs as failures. We additionally introduce *Diverse-Hit AUC (DH-AUC)* to measure the diversity of successful solutions.

DH-AUC builds on the idea of measuring chemical-space coverage through thresholded counts of structurally distinct molecules (Xie et al., 2023; Renz et al., 2024). Rather than evaluating the number of diverse successful molecules at a single distance threshold, DH-AUC integrates this quantity over the full threshold range and normalizes by the number of sampled outputs. Let $H_{\mathcal{T}}$ denote the set of distinct jointly successful molecules for task $\mathcal{T}$, and let $d_{\text{Tan}}(\cdot,\cdot)$ denote the Tanimoto distance between Morgan fingerprints of radius 2 and 2048 bits. For threshold $D \in [0,1]$,

$$\mu_{\mathcal{T}}(D) = \max\big\{\, |U| \;:\; U \subseteq H_{\mathcal{T}},\;\; d_{\text{Tan}}(m,m') \geq D \;\text{ for all }\; m \neq m' \in U \,\big\}, \tag{7}$$

which we approximate using greedy sphere exclusion. We then define

$$\text{DH-AUC}(\mathcal{T}) = \frac{1}{K}\int_0^1 \mu_{\mathcal{T}}(D)\,\mathrm{d}D \in [0,1], \tag{8}$$

where K is the number of sampled outputs per task ($K = 128$ for property targeting and bioactivity). We evaluate the integral on a uniform grid with step size 0.05. DH-AUC rewards both the number of jointly successful solutions and their separation across distance thresholds. It attains one only if every sampled output is jointly successful and all successful molecules are mutually maximally dissimilar.

3 Related work

Molecular generators and optimizers. Early methods relied on heuristic search methods such as genetic algorithms (GAs) (Westhead et al., 1995), which remain competitive baselines for molecular optimization (Jensen, 2019; Yoshikawa et al., 2018). With the rise of deep learning, molecular generation has increasingly adopted neural generative models operating over molecular graphs (You et al., 2018; Zhou et al., 2019) or string representations such as SMILES (Weininger, 1988) and SELFIES (Krenn et al., 2022). Representative approaches include VAEs (Gómez-Bombarelli et al., 2018), GFlowNets (Bengio et al., 2021), masked (Mahmood et al., 2021) and autoregressive generators (Segler et al., 2018), diffusion models (Hoogeboom et al., 2022), and, more recently, large language models (LLMs). Our approach belongs to the family of autoregressive SMILES generators.

Reference-constrained molecular optimization additionally requires preserving structural similarity to a starting molecule. Reference-constrained generation has been approached through scaffold decoration (Arús-Pous et al., 2020; Fialková et al., 2022), graph-to-graph molecular transformation (Jin et al., 2019), and fingerprint-based similarity objectives (Brown et al., 2019). We instead impose a graph edit distance interval, aligning the structural constraint with the edit operations available to our structured model.

Goal-directed molecular optimization and conditional generation. Goal-directed molecular generation steers generation toward desired properties. Classical approaches include GAs, while neural generators are commonly adapted after pretraining through iterative goal-directed optimization, particularly RL (Olivecrona et al., 2017; You et al., 2018), or trained as conditional generators that receive desired molecular properties as input (Kotsias et al., 2020; Wang et al., 2021). More recently, general-purpose LLMs have generated molecules under explicit structural constraints (Bartmann et al., 2026) and served as proposal operators in goal-directed molecular optimization (Wang et al., 2025). Supervised reasoning and RL post-training have further achieved state-of-the-art performance on molecular-design tasks (Narayanan et al., 2026). Several works also combine conditional generation with subsequent RL optimization (Wang et al., 2021; Zholus et al., 2025; Zhu et al., 2026). Our models similarly combine conditional autoregressive generation with RL post-training but, like GAs, optimize molecules through iterative modifications.

Structured and agentic molecular reasoning. Chain-of-thought prompting and scratchpads introduced explicit intermediate computation into language-model generation (Wei et al., 2022; Nye et al., 2021), while reasoning models trained with verifiable rewards optimize such traces using outcome-based feedback (Guo et al., 2025). ReAct and Toolformer further connect language-model generation to external actions and observations (Yao et al., 2023; Schick et al., 2023). Recent molecular reasoning models use linguistic or structured traces for molecular generation and optimization, including ether0 (Narayanan et al., 2026), MolReasoner (mol), DrugImproverGPT (Liu et al., 2026), and MolRGen (Formont et al., 2026). RePO studies reference-constrained molecular optimization using textual reasoning traces (Li et al., 2026b), while STRIDE generates executable edit trajectories for biological sequences (Zhang et al., 2026). Our work instead makes molecules themselves the intermediate reasoning states. We compare direct answer generation with CoM, each in non-agentic and agentic forms, to distinguish the contribution of molecular reasoning from that of environment interaction. This distinction is relevant because self-generated reasoning traces need not faithfully reflect the computation producing the final answer (Turpin et al., 2023; Lanham et al., 2023).

Molecular optimization benchmarks and comparisons. MOSES (Polykovskiy et al., 2020), GuacaMol (Brown et al., 2019), and PMO (Gao et al., 2022) provide standardized benchmarks for molecular generation and goal-directed optimization. These benchmarks generally do not condition each task on a specific starting molecule. GuacaMol and PMO include similarity-, scaffold-, and rediscovery-based objectives, but these are defined relative to fixed benchmark targets rather than per-instance reference molecules. They are therefore not directly suited to our reference-constrained setting, where both the desired property and permitted structural change are specified relative to each starting molecule. A recLLM-based optimizer (Li et al., 2026b) considers related reference-

constrained tasks, but differs substantially in model scale, pretraining corpora, and finetuning tasks. Direct performance comparisons are therefore difficult to interpret fairly, so we treat this method primarily as external context rather than controlled baseline.

4 CONTROLLED COMPARISON OF GENERATION PARADIGMS

We design this experiment to compare the generation paradigms under controlled and exactly verifiable optimization objectives. Across five deterministic molecular properties, we compare four paradigms that vary in their use of intermediate molecular states and interleaved environment execution. We evaluate them in in-distribution, interpolation gap, and extrapolation regimes to test whether these differences persist under target-value generalization.

4.1 OPTIMIZATION TASKS AND EVALUATION REGIMES

Tasks. We evaluate the constrained molecular optimization problem from Section 2.1 across five molecular-property objectives: the octanol–water partition coefficient ($\log P$), topological polar surface area (TPSA), the fraction of sp^3-hybridized carbon atoms (Fsp^3), the quantitative estimate of drug-likeness (QED), and synthetic accessibility (SA). We compute these properties using RDKit (Landrum et al., 2006), with $\log P$, QED, and SA following the definitions of Wildman & Crippen (1999), Bickerton et al. (2012), and Ertl & Schuffenhauer (2009), respectively. Their deterministic and efficiently verifiable nature allows fast comparison without confounding noise from learned or approximate evaluators.

Evaluation regimes. For each property j, training targets are sampled from a property-specific range $\mathcal{R}_j$, excluding an interior interval $\mathcal{G}_j$ held out to evaluate interpolation to unseen target values. Thus, we evaluate three target-value regimes: (a) **in-distribution:** $v^\star \in \mathcal{R}_j \setminus \mathcal{G}_j$, (b) **interpolation gap:** $v^\star \in \mathcal{G}_j$, corresponding to target values inside the training range but excluded during training, and (c) **extrapolation:** $v^\star \notin \mathcal{R}_j$. Figure A1 illustrates these ranges, and Table A1 lists the intervals for all properties. In all regimes, evaluation reference molecules are scaffold-disjoint from training, as described in Section 4.2.

Structural constraints. For the structural constraint, we use graph edit distance based on atom and bond differences relative to the maximum common substructure (Dalke & Hastings, 2013). Each task specifies an interval $[\ell, h]$ bounding the permitted structural change from the reference; the exact metric is given in Appendix A.4.

4.2 CORPUS CONSTRUCTION, DATA SPLITS, AND GENERATION FORMATS

Molecular pool and scaffold split. We begin from a curated pool of approximately 3.4 million single-component organic molecules. Every reference in this shared pool lies within the training range and outside the held-out gap for *all five properties simultaneously*, regardless of which property a task requests. We reserve 15% of Bemis–Murcko scaffolds for evaluation and construct the training corpora from the remaining scaffolds. Complete filtering, splitting, and leakage-prevention criteria are provided in Appendix A.

Trajectory construction and validation. To construct each optimization problem, we sample a reference molecule, property, target value, and edit-distance interval. A genetic algorithm (Section A.5.2) then searches for a molecule satisfying both constraints in Eq. (2). For successful searches, we retain the selected solution and the sequence of molecular states connecting it to the reference molecule. The retained solutions are subsequently normalized and validated before being added to the corpus. Following hindsight relabelling (Andrychowicz et al., 2017), we replace each target with the objective value attained by the selected final molecule: the distorted property value for pretraining and the original property value for SFT. Each retained solution therefore matches its relabelled target exactly and satisfies the requested edit-distance interval. We additionally replay structured trajectories in the chemical environment and discard any trajectory whose actions fail to reproduce the selected solution.

Leakage prevention. We apply additional exclusions to prevent information leakage across training and evaluation conditions. For each property j, the interpolation interval $\mathcal{G}_j$ is excluded from the

training reference pool, target sampling, and hindsight-relabeled solutions. References reserved for RL are likewise excluded from the supervised training corpus.

Generation formats. For each optimization task, this procedure yields a shared solution record consisting of a reference molecule, an exact target, a selected solution, and its transformation path. From each record, we derive one generation format for each of the four paradigms, ensuring that comparisons do not confound the model interface with differences in the underlying optimization problems or solution molecules. Each format defines how the shared record is serialized during pretraining and supervised finetuning, including which tokens are predicted by the model and which are provided as environment context. At inference, each model follows the corresponding generation protocol. Full serialization, supervision, and interaction protocols are provided in Appendix B.

4.3 Model configuration, training and evaluation

We use models with approximately 1M and 20M parameters for both property targeting and downstream bioactivity adaptation. All models first undergo relation-distorted pretraining on approximately 2.5 million examples, learning molecular syntax, format-specific generation protocols, and search behavior without a fixed relationship between molecular structure and the target properties. For property targeting, we apply supervised finetuning on 50,000 true-property examples followed by RL, and also compare RL directly from pretraining. Bioactivity adaptation follows the direct route: property-relation-distorted pretrained models are adapted through RL without bioactivity-specific pretraining or SFT. Both settings use Group Relative Policy Optimization (GRPO) (Shao et al., 2024). Full model and training details are provided in Appendix C.

For evaluation, we sample 128 outputs per task at temperature 1.0 with a total sequence limit of 6,144 tokens. Stop tokens and interaction budgets are format-specific and match those used in training. No intermediate property, bioactivity, or GED scores are supplied during generation in our experiments. Property targeting uses 500 tasks per property–regime pair across five properties and three evaluation regimes, giving 7,500 tasks in total (Section 4.1). Bioactivity evaluation uses 750 tasks per target–reference-band pair across three targets and two reference-activity bands, giving 4,500 tasks in total (Section 5). Missing or invalid final molecules count as failures for all task constraints. We report the metrics defined in Section 2.5, with headline results macro-averaged across properties or bioactivity targets within each evaluation regime or reference-activity band. Standard errors are computed over evaluation tasks.

4.4 Results

Agentic CoM leads joint success in distribution and in the gap, and DH-AUC across regimes. After SFT and RL, the 1M Agentic CoM reaches joint success of 0.460 in distribution and 0.495 on interpolation gap targets, compared with 0.397 and 0.379 for the next-best formats (Figure 3). At 20M, the corresponding Agentic CoM scores are 0.553 and 0.576 (Figure D1). Agentic CoM has the highest DH-AUC among the four current formats in all three regimes at both sizes. This combines successful-output yield and diversity; it does not isolate diversity conditional on success. Extrapolation remains difficult: the 20M Agentic CoM scores 0.155, while at 1M the edit program scores 0.134 versus 0.121 for Agentic CoM. Following Li et al. (2026b), we evaluate self-generated CoM and Agentic CoM at 1M and 20M parameters on TOMG-Bench's QED and logP tasks (Li et al., 2026a), without further training. The 20M Agentic CoM checkpoint trained with RL directly from pretraining achieves the highest success–similarity product on both tasks among the compared systems, including substantially larger LLM-based baselines. Training data and budgets differ across systems, so this comparison provides external context rather than a controlled test. Appendix D.5 provides the evaluation protocol and full results.

Intermediate molecular states support property control, while environment execution supports structural control. Figure 3 also shows that the generation paradigms differ substantially in the balance between property and structural success. For the 1M models in distribution, the edit program attains structural success of 0.903 but property success of 0.358. Agentic CoM attains 0.885 and 0.505, respectively, and the highest joint success. Direct SMILES and self-generated CoM have lower structural success (0.626 and 0.341). These comparisons are consistent with complementary benefits of executed edits and observable molecular states, rather than two independently beneficial additions: at 1M, neither control exceeds Direct SMILES in joint success. Under extrapolation, property success

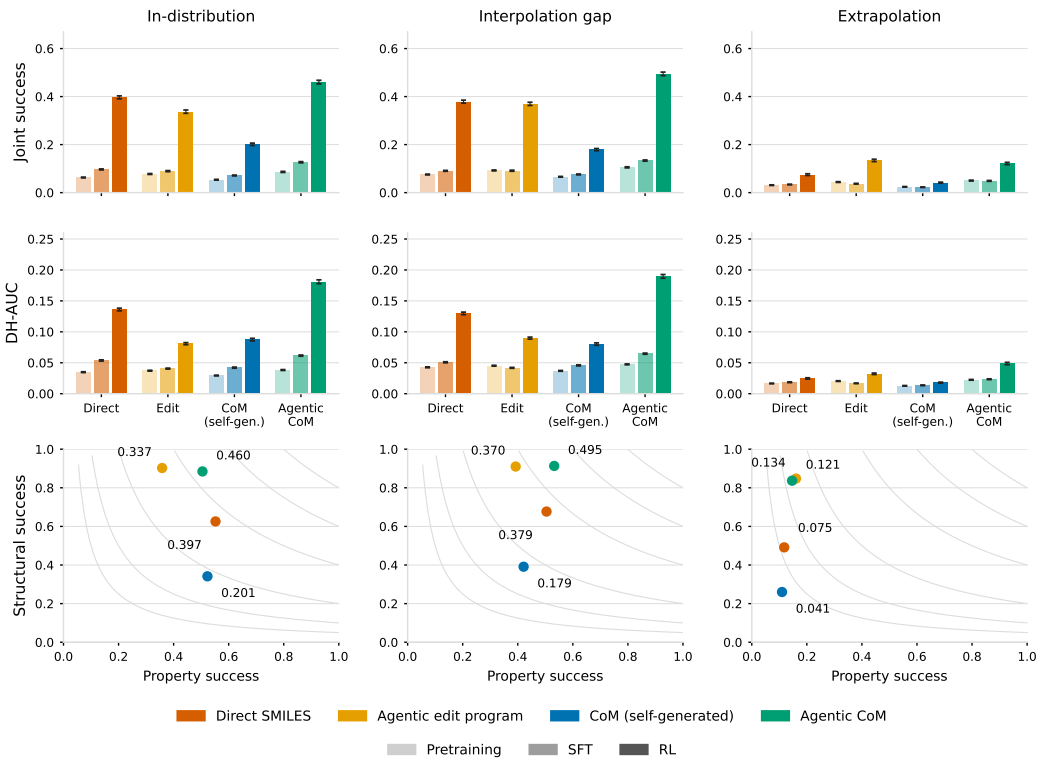

Figure 3: **Optimization performance for 1M models. Top:** Joint success and DH-AUC across four generation paradigms, three training stages, and three target regimes. Error bars are ± 1 SE over tasks. **Bottom:** Property versus structural success after GRPO; points show joint success and grey curves show contours of constant product.

drops sharply for every format, even when structural success remains comparatively high for the executed formats.

Agentic execution makes molecular trajectories verifiable. Optimizing the final molecule does not ensure that the trajectory faithfully represents its transformations. For 1M self-generated CoM, SFT-to-RL increases property-improving trajectories from 51% to 76%, while backtrack-label inconsistency rises from 26% to 65%. Agentic CoM closes this gap by having the environment execute accepted edits and supply each resulting state. Its trajectory is therefore a replayable record of the transformations that produced the committed molecule, with state-transition consistency enforced by construction. This guarantee concerns execution; it does not establish a mechanistic explanation of the model's internal computation (Turpin et al., 2023; Lanham et al., 2023). Table D11 reports these checks for both model sizes and also covers the bioactivity tasks introduced in Section 5.

Scale, supervised initialization, and RL all matter. RL yields the largest gains across training stages, and the benefit of SFT initialization becomes smaller at larger model scale. Direct RL from pretraining is already effective (Table D7; Figure D3), motivating its application to objectives outside the supervised property-targeting tasks.

5 Adaptation to Downstream Bioactivity Objectives

5.1 Experimental Design

Adapting pretrained models to new objectives through RL could broaden their use in drug discovery and chemical design without requiring task-specific supervised trajectories. We investigate how effectively this adaptation works across the four generation paradigms and both model sizes. Specifically, we adapt the pretrained base models directly through RL to three bioactivity objectives: DRD2,

Table 1: **Adaptation to downstream bioactivity objectives.** Results for 1M models, macro-averaged across DRD2, GSK3β, and JNK3. Moderate and Low denote reference-molecule activity in $[0.3, 0.5)$ and $[0, 0.3)$, respectively. Error bars indicate SE.

	Moderate		Low	
Generation paradigm	Joint ↑	DH-AUC ↑	Joint ↑	DH-AUC ↑
Direct SMILES	0.305 ± 0.007	0.075 ± 0.002	$\mathbf{0.023 \pm 0.002}$	$\mathbf{0.007 \pm 0.001}$
Agentic edit program	0.405 ± 0.009	0.076 ± 0.002	0.021 ± 0.002	0.004 ± 0.000
CoM (self-generated)	0.211 ± 0.006	0.055 ± 0.002	0.002 ± 0.001	0.001 ± 0.000
Agentic CoM	$\mathbf{0.479 \pm 0.009}$	$\mathbf{0.122 \pm 0.002}$	$\mathbf{0.025 \pm 0.003}$	$\mathbf{0.006 \pm 0.001}$

GSK3β, and JNK3. These objectives are absent from pretraining, and no intermediate SFT stage is used.

We use the pretrained activity predictors distributed with the Therapeutics Data Commons (Huang et al., 2021) as optimization oracles, each returning a predicted activity score in $[0, 1]$. Unlike the property-targeting setting, the objective is directional: models are rewarded for increasing predicted activity while satisfying the requested structural-distance constraint. Bioactivity references are selected from a separate subset of the cleaned molecular library using activity scores, without applying the five property-range and gap filters. We evaluate two reference-activity bands: moderate activity in $[0.3, 0.5)$ and low activity in $[0.0, 0.3)$. An activity hit requires predicted activity of at least 0.5; joint success additionally requires the requested GED interval. Appendix D.4 also reports the stricter activity threshold of 0.8. Evaluation otherwise follows the sampling and validity conventions as described in Section 4. Details are provided in Appendix A.3.2.

5.2 Results

Table 1 presents downstream bioactivity results for the 1M models (20M results: Table D15). The pretrained base models can be adapted directly to new bioactivity objectives through RL, without task-specific supervised fine-tuning. Among the four generation paradigms, Agentic CoM yields the strongest adaptation to the bioactivity tasks, particularly for moderate-activity references. Performance from low-activity references remains limited across all paradigms.

6 Conclusion and Limitations

We introduced CoM as a representation for molecular reasoning and Agentic CoM as its environment-executed realization. In a controlled property-targeting experiment, Agentic CoM achieves the strongest balance of property satisfaction, structural control, and solution diversity, while comparisons with the other generation paradigms reveals the complementary roles of intermediate molecular states and environment execution. We further find that the Agentic CoM base model adapts most effectively through reinforcement learning to downstream bioactivity objectives without task-specific supervised finetuning. Together, these results support explicit molecular trajectories and environment feedback both as an effective optimization paradigm and as a basis for reusable pretrained models for molecular optimization.

Our experiments remain computational: the property objectives are deterministic functions, while the bioactivity objectives are predictions from pretrained oracles rather than experimental measurements. We furthermore study relatively small models, and our external comparison on TOMG-Bench is not matched in pretraining, task-specific supervision, or compute. Supervised trajectories are constructed from a fixed fragment library, although post-trained policies generate fragments autoregressively and can move beyond this catalogue subject to the edit grammar and validity filters. Future work should study larger models and adaptation to noisy experimental, delayed, and multi-objective feedback.

AI USE STATEMENT

Large Language Models (LLMs), including Anthropic's Claude Code and OpenAI's Codex, were used during the preparation of this work for writing assistance, literature retrieval and discovery, research ideation, implementation support, analysis, and drafting. These tools assisted with locating and summarizing related literature; proposing and refining experimental comparisons; implementing and debugging data-processing, training, evaluation, and analysis code; interpreting results; preparing tables and figures; and improving the wording, clarity, and organization of the manuscript. All AI-assisted code, analyses, references, and manuscript content were reviewed and verified by the authors. The authors made the scientific and methodological decisions and take full responsibility for the correctness and originality of the final text, methods, experiments, results, and conclusions.

ETHICS AND IMPACT STATEMENT

This work studies computational methods for constrained molecular optimization. More controllable molecular search and explicit records of the edits made during generation could support drug-discovery research and make computational comparisons easier to inspect. However, our experiments do not establish synthetic feasibility, experimentally confirmed activity, safety, or therapeutic utility. Chemical validity and success on property, structural, or predicted-bioactivity objectives are not safety guarantees; optimizing these objectives can also exploit limitations of the scoring functions.

As with other molecular-design methods, responsible use should account for both potential benefits and possible misuse. Applications in drug discovery would require expert review and experimental validation of activity and safety, including toxicity and off-target effects.

REPRODUCIBILITY STATEMENT

Code, fixed evaluation benchmarks, experiment configurations, and instructions for data generation, training, and evaluation are available in our anonymous repository. The reference molecule pools, model checkpoints, and tokenizers are provided on Figshare. Appendices A–C detail data construction, interaction protocols, and model training. Section 4.3 and Appendix D specify evaluation and uncertainty estimates, and Appendix D.5 documents the external RePO comparison.

REFERENCES

MolReasoner: Toward effective and interpretable reasoning for molecular llms, author=Zhao, Guojiang and Lu, Zixiang and Ge, Yutang and Li, Sihang and Cheng, Zheng and Lin, Haitao and Wu, Lirong and Xia, Hanchen and Cai, Hengxing and Guo, Wentao and others, journal=arXiv preprint, year=2025, url=https://arxiv.org/abs/2508.02066, doi=,.

Andrychowicz, M., Wolski, F., Ray, A., Schneider, J., Fong, R., Welinder, P., McGrew, B., Tobin, J., Pieter Abbeel, O., and Zaremba, W. Hindsight experience replay. *Advances in Neural Information Processing Systems*, 30, 2017. URL.

Arús-Pous, J., Patronov, A., Bjerrum, E. J., Tyrchan, C., Reymond, J.-L., Chen, H., and Engkvist, O. SMILES-based deep generative scaffold decorator for de-novo drug design. *Journal of Cheminformatics*, 12(1):38, 2020. 10.1186/s13321-020-00441-8.

Bartmann, C., Schimunek, J., Ielanskyi, M., Seidl, P., Klambauer, G., and Luukkonen, S. MolecularIQ: Characterizing chemical reasoning capabilities through symbolic verification on molecular graphs. In *International Conference on Learning Representations*, 2026. URL.

Bemis, G. W. and Murcko, M. A. The properties of known drugs. 1. molecular frameworks. *Journal of Medicinal Chemistry*, 39(15):2887–2893, 1996. 10.1021/jm9602928.

Bengio, E., Jain, M., Korablyov, M., Precup, D., and Bengio, Y. Flow network based generative models for non-iterative diverse candidate generation. *Advances in Neural Information Processing Systems*, 34:27381–27394, 2021. URL.

Bickerton, G. R., Paolini, G. V., Besnard, J., Muresan, S., and Hopkins, A. L. Quantifying the chemical beauty of drugs. *Nature Chemistry*, 4(2):90–98, 2012. 10.1038/nchem.1243.

Brown, N., Fiscato, M., Segler, M. H., and Vaucher, A. C. GuacaMol: benchmarking models for de novo molecular design. *Journal of Chemical Information and Modeling*, 59(3):1096–1108, 2019. 10.1021/acs.jcim.8b00839.

Dalke, A. and Hastings, J. FMCS: a novel algorithm for the multiple MCS problem. *Journal of Cheminformatics*, 5(Suppl 1):O6, 2013. 10.1186/1758-2946-5-S1-O6.

Dubey, A., Jauhri, A., Pandey, A., Kadian, A., et al. The Llama 3 herd of models. *arXiv preprint*, 2024. URL.

Edwards, C., Lai, T., Ros, K., Honke, G., Cho, K., and Ji, H. Translation between molecules and natural language. In *Proceedings of the 2022 Conference on Empirical Methods in Natural Language Processing*, pp. 375–413, 2022. 10.18653/v1/2022.emnlp-main.26.

Ertl, P. and Schuffenhauer, A. Estimation of synthetic accessibility score of drug-like molecules based on molecular complexity and fragment contributions. *Journal of Cheminformatics*, 1(1):8, 2009. 10.1186/1758-2946-1-8.

Fei, H., Wu, S., Ji, W., Zhang, H., Zhang, M., Lee, M.-L., and Hsu, W. Video-of-thought: Step-by-step video reasoning from perception to cognition. In *Proceedings of the 41st International Conference on Machine Learning*, pp. 13109–13125, 2024. URL.

Fialková, V., Zhao, J., Papadopoulos, K., Engkvist, O., Bjerrum, E. J., Kogej, T., and Patronov, A. LibINVENT: reaction-based generative scaffold decoration for in silico library design. *Journal of Chemical Information and Modeling*, 62(9):2046–2063, 2022. 10.1021/acs.jcim.1c00469.

Formont, P., Darrin, M., Ayed, I. B., and Piantanida, P. MolRGen: A training and evaluation setting for de novo molecular generation with reasonning models. *arXiv preprint*, 2026. 10.48550/arXiv.2603.18256.

Gandhi, K., Lee, D. H., Grand, G., Liu, M., Cheng, W., Sharma, A., and Goodman, N. D. Stream of Search (SoS): Learning to search in language. In *First Conference on Language Modeling*, 2024. URL.

Gandhi, K., Chakravarthy, A. K., Singh, A., Lile, N., and Goodman, N. Cognitive behaviors that enable self-improving reasoners, or, four habits of highly effective STars. In *Second Conference on Language Modeling*, 2025. URL.

Gao, W., Fu, T., Sun, J., and Coley, C. Sample efficiency matters: a benchmark for practical molecular optimization. *Advances in Neural Information Processing Systems*, 35:21342–21357, 2022. URL.

Gómez-Bombarelli, R., Wei, J. N., Duvenaud, D., Hernández-Lobato, J. M., Sánchez-Lengeling, B., Sheberla, D., Aguilera-Iparraguirre, J., Hirzel, T. D., Adams, R. P., and Aspuru-Guzik, A. Automatic chemical design using a data-driven continuous representation of molecules. *ACS Central Science*, 4(2):268, 2018. 10.1021/acscentsci.7b00572.

Guo, D., Yang, D., Zhang, H., Song, J., Wang, P., Zhu, Q., Xu, R., Zhang, R., Ma, S., Bi, X., et al. DeepSeek-R1 incentivizes reasoning in LLMs through reinforcement learning. *Nature*, 645: 633–638, 2025. 10.1038/s41586-025-09422-z.

Hoogeboom, E., Satorras, V. G., Vignac, C., and Welling, M. Equivariant diffusion for molecule generation in 3D. In *International Conference on Machine Learning*, pp. 8867–8887, 2022. URL.

Huang, K., Fu, T., Gao, W., Zhao, Y., Roohani, Y., Leskovec, J., Coley, C. W., Xiao, C., Sun, J., and Zitnik, M. Therapeutics data commons: Machine learning datasets and tasks for drug discovery and development. *Proceedings of Neural Information Processing Systems, NeurIPS Datasets and Benchmarks*, 2021. URL.

Irwin, J. J., Sterling, T., Mysinger, M. M., Bolstad, E. S., and Coleman, R. G. ZINC: a free tool to discover chemistry for biology. *Journal of Chemical Information and Modeling*, 52(7):1757–1768, 2012. 10.1021/ci3001277.

Jensen, J. H. A graph-based genetic algorithm and generative model/Monte Carlo tree search for the exploration of chemical space. *Chemical Science*, 10(12):3567–3572, 2019. 10.1039/c8sc05372c.

Jin, W., Yang, K., Barzilay, R., and Jaakkola, T. Learning multimodal graph-to-graph translation for molecular optimization. In *International Conference on Learning Representations*, 2019. URL.

Jin, W., Barzilay, R., and Jaakkola, T. Multi-objective molecule generation using interpretable substructures. In *International Conference on Machine Learning*, pp. 4849–4859. PMLR, 2020. URL.

Jordan, K., Jin, Y., Boza, V., You, J., Cesista, F., Newhouse, L., and Bernstein, J. Muon: An optimizer for hidden layers in neural networks, 2024. URL.

Kim, S., Chen, J., Cheng, T., Gindulyte, A., He, J., He, S., Li, Q., Shoemaker, B. A., Thiessen, P. A., Yu, B., et al. Pubchem 2023 update. *Nucleic Acids Research*, 51(D1):D1373–D1380, 2023. 10.1093/nar/gkac956.

Kotsias, P.-C., Arús-Pous, J., Chen, H., Engkvist, O., Tyrchan, C., and Bjerrum, E. J. Direct steering of de novo molecular generation with descriptor conditional recurrent neural networks. *Nature Machine Intelligence*, 2(5):254–265, 2020. 10.1038/s42256-020-0174-5.

Krenn, M., Ai, Q., Barthel, S., Carson, N., Frei, A., Frey, N. C., Friederich, P., Gaudin, T., Gayle, A. A., Jablonka, K. M., et al. Selfies and the future of molecular string representations. *Patterns*, 3 (10), 2022. 10.1016/j.patter.2022.100588.

Landrum, G. et al. RDKit: Open-source cheminformatics, 2006. URL.

Lanham, T., Chen, A., Radhakrishnan, A., Steiner, B., Denison, C., Hernandez, D., Li, D., Durmus, E., Hubinger, E., Kernion, J., et al. Measuring faithfulness in chain-of-thought reasoning. *arXiv preprint*, 2023. 10.48550/arXiv.2307.13702.

Lehnert, L., Sukhbaatar, S., Su, D., Zheng, Q., McVay, P., Rabbat, M., and Tian, Y. Beyond A*: Better planning with transformers via search dynamics bootstrapping. In *First Conference on Language Modeling*, 2024. URL.

Li, J., Li, J., Wang, W., Liu, Y., Zheng, C., Bian, Y., Zhou, D., Wei, X.-Y., and Li, Q. Speak-to-Structure: Evaluating LLMs in open-domain natural language-driven molecule generation. In *Proceedings of the 32nd ACM SIGKDD Conference on Knowledge Discovery and Data Mining*, 2026a. 10.1145/3770855.3817473.

Li, X., Zhou, Z., Li, Z., Yao, J., Rong, Y., Zhang, L., and Han, B. Reference-guided policy optimization for molecular optimization via LLM reasoning. In *International Conference on Learning Representations*, 2026b. URL.

Liu, A., Mei, A., Lin, B., Xue, B., Wang, B., Xu, B., Wu, B., Zhang, B., Lin, C., Dong, C., et al. DeepSeek-V3.2: Pushing the frontier of open large language models. *arXiv preprint*, 2025. 10.48550/arXiv.2512.02556.

Liu, X., Jiang, S., Chen, S., Yang, Z., Chen, Y., Foster, I. T., and Stevens L., R. Gpt-driven drug optimization with structured policy optimization post-training. In *Proceedings of the 11th Machine Learning for Healthcare Conference*, pp. 1208–1242, 2026. URL.

Mahmood, O., Mansimov, E., Bonneau, R., and Cho, K. Masked graph modeling for molecule generation. *Nature Communications*, 12(1):3156, 2021. 10.1038/s41467-021-23415-2.

Muennighoff, N., Yang, Z., Shi, W., Li, X. L., Fei-Fei, L., Hajishirzi, H., Zettlemoyer, L., Liang, P., Candès, E., and Hashimoto, T. s1: Simple test-time scaling. In *Proceedings of the 2025 Conference on Empirical Methods in Natural Language Processing*, 2025. 10.18653/v1/2025.emnlp-main.1025.

Narayanan, S., Braza, J., Griffiths, R.-R., Bou, A., Wellawatte, G., Caldas Ramos, M., Mitchener, L., Pieler, M., Rodriques, S., and White, A. Training a scientific reasoning model for chemistry. *Advances in Neural Information Processing Systems*, 38:157671–157710, 2026. URL.

NeMo, R. A scalable and efficient post-training library, 2025. URL.

Nye, M., Andreassen, A. J., Gur-Ari, G., Michalewski, H., Austin, J., Bieber, D., Dohan, D., Lewkowycz, A., Bosma, M., Luan, D., et al. Show your work: Scratchpads for intermediate computation with language models. *arXiv preprint*, 2021. 10.48550/arXiv.2112.00114.

Olivecrona, M., Blaschke, T., Engkvist, O., and Chen, H. Molecular de-novo design through deep reinforcement learning. *Journal of Cheminformatics*, 9(1):48, 2017. 10.1186/s13321-017-0235-x.

OpenAI. GPT-4o mini: Advancing cost-efficient intelligence, 2024. URL.

Özçelik, R., Brinkmann, H., Criscuolo, E., and Grisoni, F. Generative Deep Learning for de Novo Drug DesignA Chemical Space Odyssey. *Journal of Chemical Information and Modeling*, 65(14): 7352–7372, 2025. 10.1021/acs.jcim.5c00641.

Pei, Q., Zhang, W., Zhu, J., Wu, K., Gao, K., Wu, L., Xia, Y., and Yan, R. BioT5: 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*, pp. 1102–1123, 2023. 10.18653/v1/2023.emnlp-main.70.

Polykovskiy, D., Zhebrak, A., Sanchez-Lengeling, B., Golovanov, S., Tatanov, O., Belyaev, S., Kurbanov, R., Artamonov, A., Aladinskiy, V., Veselov, M., Kadurin, A., Johansson, S., Chen, H., Nikolenko, S., Aspuru-Guzik, A., and Zhavoronkov, A. Molecular Sets (MOSES): A Benchmarking Platform for Molecular Generation Models. *Frontiers in Pharmacology*, 11:565644, 2020. 10.3389/fphar.2020.565644.

Qwen, Yang, A., Yang, B., Zhang, B., Hui, B., et al. Qwen2.5 technical report. *arXiv preprint*, 2025. 10.48550/arXiv.2412.15115.

Renz, P., Luukkonen, S., and Klambauer, G. Diverse hits in de novo molecule design: Diversity-based comparison of goal-directed generators. *Journal of Chemical Information and Modeling*, 64(15): 5756–5761, 2024. 10.1021/acs.jcim.4c00519.

Schick, T., Dwivedi-Yu, J., Dessì, R., Raileanu, R., Lomeli, M., Hambro, E., Zettlemoyer, L., Cancedda, N., and Scialom, T. Toolformer: Language models can teach themselves to use tools. In *Advances in Neural Information Processing Systems*, volume 36, pp. 68539–68551, 2023. URL.

Segler, M. H., Kogej, T., Tyrchan, C., and Waller, M. P. Generating focused molecule libraries for drug discovery with recurrent neural networks. *ACS Central Science*, 4(1):120–131, 2018. 10.1021/acscentsci.7b00512.

Shao, Z., Wang, P., Zhu, Q., Xu, R., Song, J., Bi, X., Zhang, H., Zhang, M., Li, Y., Wu, Y., et al. DeepSeekMath: Pushing the limits of mathematical reasoning in open language models. *arXiv preprint*, 2024. 10.48550/arXiv.2402.03300.

Sun, J., Jeliazkova, N., Chupakhin, V., Golib-Dzib, J.-F., Engkvist, O., Carlsson, L., Wegner, J., Ceulemans, H., Georgiev, I., Jeliazkov, V., et al. ExCAPE-DB: an integrated large scale dataset facilitating big data analysis in chemogenomics. *Journal of Cheminformatics*, 9(1):17, 2017. 10.1186/s13321-017-0203-5.

Tang, X., Dai, H., Knight, E., Wu, F., Li, Y., Li, T., and Gerstein, M. A survey of generative AI for de novo drug design: new frontiers in molecule and protein generation. *Briefings in Bioinformatics*, 25(4):bbae338, 2024. 10.1093/bib/bbae338.

Team, K., Bai, T., Bai, Y., Bao, Y., Cai, S., Cao, Y., Chai, Z., Charles, Y., Che, H., Chen, C., et al. Kimi K2.5: Visual agentic intelligence. *arXiv preprint*, 2026. 10.48550/arXiv.2602.02276.

Turpin, M., Michael, J., Perez, E., and Bowman, S. Language models don't always say what they think: Unfaithful explanations in chain-of-thought prompting. *Advances in Neural Information Processing Systems*, 36:74952–74965, 2023. URL.

Wang, H., Skreta, M., Ser, C.-T., Gao, W., Kong, L., Strieth-Kalthoff, F., Duan, C., Zhuang, Y., Yu, Y., Zhu, Y., et al. Efficient evolutionary search over chemical space with large language models. In *International Conference on Learning Representations*, volume 2025, pp. 51694–51727, 2025. URL.

Wang, J., Hsieh, C.-Y., Wang, M., Wang, X., Wu, Z., Jiang, D., Liao, B., Zhang, X., Yang, B., He, Q., et al. Multi-constraint molecular generation based on conditional transformer, knowledge distillation and reinforcement learning. *Nature Machine Intelligence*, 3(10):914–922, 2021. 10.1038/s42256-021-00403-1.

Wei, J., Wang, X., Schuurmans, D., Bosma, M., Xia, F., Chi, E., Le, Q. V., Zhou, D., et al. Chain-of-thought prompting elicits reasoning in large language models. *Advances in Neural Information Processing Systems*, 35:24824–24837, 2022. URL.

Weininger, D. 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, 1988. 10.1021/ci00057a005.

Westhead, D. R., Clark, D. E., Frenkel, D., Li, J., Murray, C. W., Robson, B., and Waszkowycz, B. PRO_LIGAND: An approach to de novo molecular design. 3. a genetic algorithm for structure refinement. *Journal of Computer-Aided Molecular Design*, 9(2):139–148, 1995. 10.1007/BF00124404.

Wildman, S. A. and Crippen, G. M. Prediction of physicochemical parameters by atomic contributions. *Journal of Chemical Information and Computer Sciences*, 39(5):868–873, 1999. 10.1021/ci9903071.

Xie, Y., Xu, Z., Ma, J., and Mei, Q. How much space has been explored? measuring the chemical space covered by databases and machine-generated molecules. In *International Conference on Learning Representations*, 2023. URL.

Yang, A., Li, A., Yang, B., Zhang, B., Hui, B., Zheng, B., Yu, B., Gao, C., Huang, C., Lv, C., et al. Qwen3 technical report. *arXiv preprint*, 2025. 10.48550/arXiv.2505.09388.

Yao, S., Zhao, J., Yu, D., Du, N., Shafran, I., Narasimhan, K., and Cao, Y. ReAct: Synergizing reasoning and acting in language models. In *International Conference on Learning Representations*, 2023. URL.

Yoshikawa, N., Terayama, K., Sumita, M., Homma, T., Oono, K., and Tsuda, K. Population-based de novo molecule generation, using grammatical evolution. *Chemistry Letters*, 47(11):1431–1434, 2018. 10.1246/cl.180665.

You, J., Liu, B., Ying, Z., Pande, V., and Leskovec, J. Graph convolutional policy network for goal-directed molecular graph generation. *Advances in Neural Information Processing Systems*, 31, 2018. URL.

Zdrazil, B., Felix, E., Hunter, F., Manners, E. J., Blackshaw, J., Corbett, S., De Veij, M., Ioannidis, H., Lopez, D. M., Mosquera, J. F., et al. The ChEMBL database in 2023: a drug discovery platform spanning multiple bioactivity data types and time periods. *Nucleic Acids Research*, 52 (D1):D1180–D1192, 2024. 10.1093/nar/gkad1004.

Zhang, D., Zhang, S., He, S., Zhang, Y., Rizvi, S. A., and van Dijk, D. STRIDE: Post-training llms to reason and refine bio-sequences via edit trajectories. *arXiv preprint*, 2026. URL.

Zhang, R., Tong, C., Zhao, Z., Guo, Z., Zhang, H., Zhang, M., Liu, J., Gao, P., and Li, H. Let's verify and reinforce image generation step by step. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition (CVPR)*, pp. 28662–28672, 2025. URL.

Zholus, A., Kuznetsov, M., Schutski, R., Shayakhmetov, R., Polykovskiy, D., Chandar, S., and Zhavoronkov, A. BindGPT: A scalable framework for 3d molecular design via language modeling and reinforcement learning. In *Proceedings of the AAAI Conference on Artificial Intelligence*, volume 39, pp. 26083–26091, 2025. 10.1609/aaai.v39i24.34804.

Zhou, Z., Kearnes, S., Li, L., Zare, R. N., and Riley, P. Optimization of molecules via deep reinforcement learning. *Scientific Reports*, 9(1):10752, 2019. 10.1038/s41598-019-47148-x.

Zhu, Y., Liu, Y., Li, W., Luo, T., and Jiang, M. Controllable molecular generative foundation models. *arXiv preprint*, 2026. 10.48550/arXiv.2605.15354.

Chain-of-Molecules - Appendix

CONTENTS

A Data, experimental design, and corpus construction **18**
A.1 Molecular pool curation . . . 18
A.2 Scaffold and reference splits . . . 18
A.3 Experimental design . . . 18
A.4 Success criteria and evaluation metrics . . . 21
A.5 Training corpora and prompt sets . . . 22

B Generation formats and interaction protocols **25**
B.1 Generation formats . . . 25
B.2 Molecular actions and addressing . . . 26
B.3 Interaction protocol . . . 26
B.4 Matched SFT examples . . . 27

C Model and training details **27**
C.1 Model and tokenizer . . . 28
C.2 Supervised pretraining and finetuning . . . 28
C.3 Reinforcement learning . . . 29

D Extended results **30**
D.1 Property-targeting performance . . . 31
D.2 RL initialization and group size . . . 40
D.3 Trajectory behaviour and consistency . . . 44
D.4 Downstream bioactivity adaptation . . . 47
D.5 Directional property optimization . . . 52

E Extended limitations **54**

F Computational resources **54**

NOTATION

We collect the recurring paper-specific notation below. Quantities used only within one equation are defined alongside that equation.

Symbol	Meaning
Task definition	
$\mathcal{M}$	Set of chemically valid molecules considered by the task.
$\mathcal{T}$	Task $(m_{\mathrm{ref}}, j, v^\star, \ell, h)$: reference molecule, objective index, requested value, and lower and upper GED bounds.
$\mathcal{J}$, j, f_j	Objective set, selected objective index, and its descriptor or predicted-activity function.
ℓ, h, w	Lower and upper GED bounds and interval width, $w = h - \ell$.
ε_j	Property tolerance for evaluation and GA solution acceptance.
$\hat{m}$, $\perp$	Committed molecule, or failure to obtain a valid one.
d_{GED}	Graph edit distance used for the structural constraint.
Generation and trajectories	
p_θ	Autoregressive model distribution with trainable parameters θ.
$\mathcal{V}$, $\mathbf{x}_\mathcal{T}$, $\mathbf{y}$	Token vocabulary, task prompt, and generated token sequence.
L, T	Token-sequence length and number of trajectory decisions.
$\boldsymbol{\tau}$, $\boldsymbol{\tau}_{<t}$	Molecular trajectory and its prefix ending at m_{t-1}, with $m_0 = m_{\mathrm{ref}}$.
m_t, a_t	Molecular state and edit or control decision at step t; a_t is executable in agentic formats and an action label in self-generated CoM.
$\mathcal{A}$, $\mathcal{E}$	Action space and symbolic-environment transition function.
Evaluation	
s_{prop}, s_{struct}, s_{joint}	Objective, structural, and joint-success indicators; s_{prop} means property success or activity hit, depending on the experiment.
κ	Bioactivity evaluation threshold (0.5 or 0.8), distinct from the prompt target $v^\star = 1.0$.
K	Evaluation outputs per task ($K = 128$), not the RL group size G.
$H_\mathcal{T}$	Set of distinct jointly successful molecules for task $\mathcal{T}$.
d_{Tan}, D	Tanimoto distance between radius-2, 2048-bit Morgan fingerprints and its separation threshold $D \in [0, 1]$.
$\mu_\mathcal{T}(D)$	Largest D-separated subset size of $H_\mathcal{T}$, approximated by greedy sphere exclusion.
DH-AUC$(\mathcal{T})$	Area under $\mu_\mathcal{T}(D)$ over $D \in [0, 1]$, normalized by K.
Data and genetic search	
$\mathcal{R}_j$, $\mathcal{G}_j$	Training target range and held-out interpolation gap for property j.
F_j, ρ	Empirical property CDF and pretraining quantile-space distortion.
$\tilde{f}_j$	GA objective: distorted for pretraining data, true property for SFT.
B_j, δ_j	Fixed normalization span and penalty-free GA error tolerance, $\delta_j = 0.02B_j$.
F_{GA}, P_{GA}, S_{GA}	GA fitness, property penalty, and structural penalty (Equation (A.3)).
Supervised and reinforcement learning	
$\mathbf{s}^{(n)}$, $\mathcal{I}_c$, $\mathcal{C}_+$	Serialized example n, unmasked target positions in token class c, and classes with at least one such position in the batch.
$\mathcal{L}_{\mathrm{sup}}$	Class-balanced negative log-likelihood for pretraining and SFT.
P, G	RL prompts per iteration and rollouts per prompt (PG total).
$q(\hat{m}; \mathcal{T})$	Objective credit: sigmoid target-matching score for properties, or predicted activity for bioactivity.
$r(\hat{m}; \mathcal{T})$	RL reward: mean structural and objective credit, divided by within-group multiplicity; invalid outputs receive zero.
$\mathrm{mult}_\mathcal{T}(\hat{m})$	Number of rollouts in the same GRPO prompt group that commit the same canonical molecule as $\hat{m}$.

A Data, experimental design, and corpus construction

A.1 Molecular pool curation

We combine compounds from PubChem (Kim et al., 2023), ChEMBL (Zdrazil et al., 2024), and ZINC250k (Irwin et al., 2012), and deduplicate them by canonical SMILES.

To establish a well-defined molecular space, we restrict reference molecules to 6–45 heavy atoms and the elements C, H, N, O, S, P, F, Cl, Br, and I. We require a single connected molecular structure and therefore exclude SMILES containing the disconnected-component marker (`.`). Isotope-labelled molecules and structures containing wildcard atoms (`*`) are also excluded. We do not require molecules to be electrically neutral: formal charges are retained, including charged functional groups and zwitterionic structures.

A.2 Scaffold and reference splits

We partition reference molecules by Bemis–Murcko scaffold (Bemis & Murcko, 1996), so that evaluation measures generalization to unseen chemotypes rather than recall of near-duplicates seen during training. A deterministic hash of the scaffold assigns 15% of scaffolds to evaluation prompts; pretraining, SFT, and RL draw references from the remaining 85%. Acyclic molecules have an empty Bemis–Murcko scaffold, so for these we hash the canonical SMILES instead. The hash is fixed across runs, making the split reproducible and independent of dataset ordering.

Within the training pool, molecules that serve as references for RL prompts are excluded from the pretraining and SFT corpora. The exclusion applies to individual molecules rather than scaffolds: these corpora may contain molecules sharing a scaffold with an RL prompt, but never an RL reference molecule itself.

Each generated corpus is split internally into training and validation sets by source identifier, so that all solutions to a given search problem fall on the same side. Without this constraint, near-identical solutions to the same problem would appear on both sides and inflate validation performance. These validation sets are used only during training and are disjoint from the held-out evaluation benchmark.

A.3 Experimental design

A.3.1 Property targeting

We use five molecular properties computed with RDKit (Landrum et al., 2006). $\log P$, calculated by the Crippen method, measures lipophilicity. Topological polar surface area (TPSA) estimates the surface area contributed by polar atoms, and Fsp^3 is the fraction of carbon atoms that are sp^3-hybridised. The quantitative estimate of drug-likeness (QED) combines molecular descriptors into a drug-likeness score. The synthetic-accessibility score (SA) estimates synthetic difficulty, with lower values indicating easier synthesis.

Training domain and reference pool. For each property j, training targets are sampled from a property-specific range $\mathcal{R}_j$, excluding an interior interval $\mathcal{G}_j$ reserved for evaluating interpolation to unseen target values. Table A1 summarizes these intervals.

Table A1: Property-specific training ranges $\mathcal{R}_j$ and held-out interpolation gaps $\mathcal{G}_j$.

Property	Training range $\mathcal{R}_j$	Held-out gap $\mathcal{G}_j$
$\log P$	$[-3.0, 8.0]$	$[2.5, 4.0]$
TPSA	$[0, 150]$	$[50, 80]$
Fsp^3	$[0.05, 0.85]$	$[0.30, 0.50]$
QED	$[0.15, 0.90]$	$[0.55, 0.70]$
SA	$[1.9, 5.0]$	$[2.7, 3.3]$

We specify the training-domain restrictions at three stages:

1. The property reference pool excludes any molecule whose value lies outside $\mathcal{R}_k$ or inside $\mathcal{G}_k$ for *any* of the five properties k. Thus every retained reference satisfies $f_k(m_{\text{ref}}) \in \mathcal{R}_k \setminus \mathcal{G}_k$ for all five properties simultaneously, even when the task requests only property j.
2. Target sampling rejects values in $\mathcal{G}_j$.
3. After hindsight relabelling (Appendix A.5.2), the retention criterion excludes solutions whose relabelled target lies outside $\mathcal{R}_j$ or inside $\mathcal{G}_j$.

Reference filtering is joint across all five properties; target sampling and the post-search criterion concern the requested property j. The latter is needed because a search initiated with an allowed target may reach a molecule whose attained value falls outside the training range or inside the held-out gap. Since hindsight relabelling replaces the original target with the attained value, retaining such a solution would introduce a target from a held-out evaluation region into training.

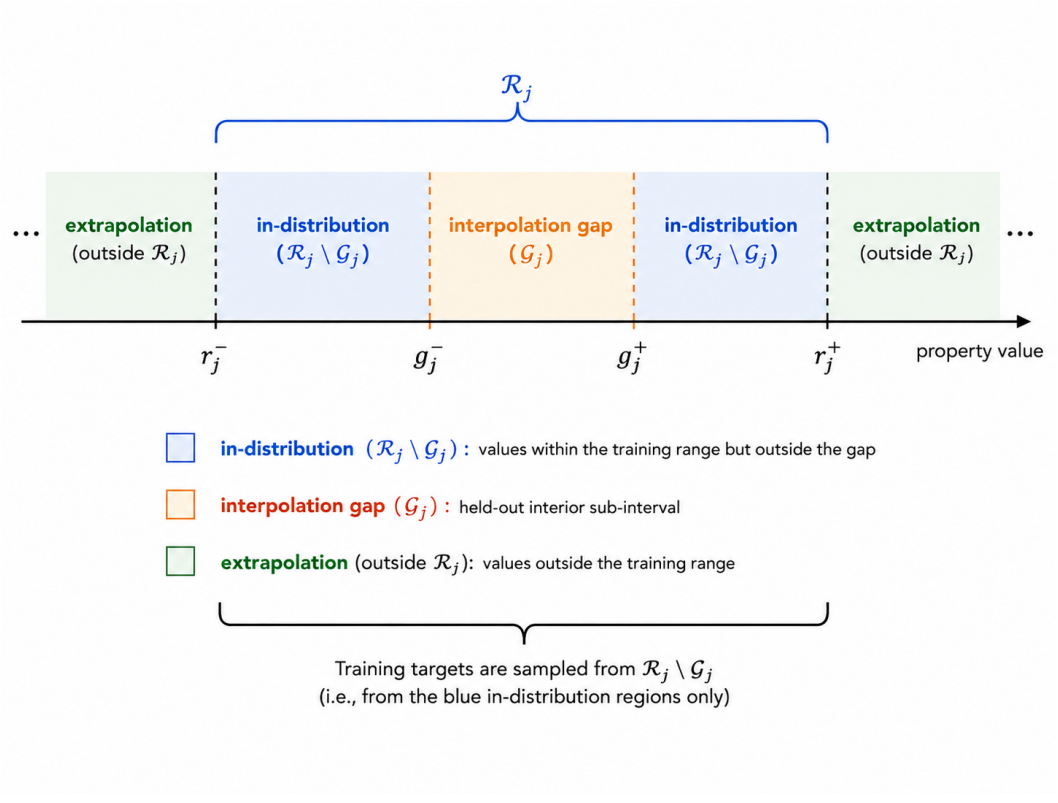

Figure A1: Target-value regimes for property j. Training targets lie in $\mathcal{R}_j \setminus \mathcal{G}_j$, interpolation gap targets lie in $\mathcal{G}_j$, and extrapolation targets lie outside $\mathcal{R}_j$.

Prompt construction. Each prompt specifies a reference molecule, a property, a target value, and a GED interval. RL prompts draw references from the training side of the scaffold split, excluding references used for the supervised corpora, and targets from $\mathcal{R}_j \setminus \mathcal{G}_j$.

GED intervals have the form $[\ell, h]$ with $h = \ell + w$, where the integer lower bound ℓ and width w each range from 0 to 30. This gives 961 possible intervals with $h \leq 60$; $w = 0$ requests an exact distance.

Evaluation design. Evaluation references come from the held-out side of the scaffold split. GED intervals are sampled uniformly from the grid above. We vary the requested target value across three regimes: in-distribution targets in $\mathcal{R}_j \setminus \mathcal{G}_j$, interpolation gap targets in $\mathcal{G}_j$, and extrapolation targets outside $\mathcal{R}_j$. For each property, we use 500 prompts per regime, giving 7,500 evaluation tasks. Figure A2 shows the target distributions for each property and regime.



Figure A2: Target distributions for the 7,500 evaluation tasks across five properties: in-distribution ($v^\star \in \mathcal{R}_j \setminus \mathcal{G}_j$), interpolation gap ($v^\star \in \mathcal{G}_j$), and extrapolation ($v^\star \notin \mathcal{R}_j$). TPSA extrapolation is restricted to values above the training range, which starts at zero.

A.3.2 Downstream bioactivity adaptation

We also evaluate three bioactivity objectives, replacing the property descriptors above with pretrained activity predictors. These scores are model predictions, not experimental measurements. We use the DRD2, GSK3β and JNK3 oracles distributed with the Therapeutics Data Commons (Huang et al., 2021), each returning a predicted activity probability in $[0, 1]$. DRD2 predicts dopamine D2 receptor activity using a support-vector classifier (Olivecrona et al., 2017). GSK3β and JNK3 predict inhibition of glycogen synthase kinase 3β and c-Jun N-terminal kinase 3 using the random-forest models of Jin et al. (2020), trained on ExCAPE-DB data (Sun et al., 2017). We use the pretrained oracles without modification or retraining.

Training domain and reference pool. We construct a separate bioactivity pool from the cleaned molecular library, without applying the five-property range and gap filters. A molecule is retained if at least one of the three oracles predicts an activity in $[0.4, 0.8]$, yielding 2,086,327 molecules. Training and evaluation references follow the scaffold split in Appendix A.2.

Prompt construction. Each prompt uses the property-targeting grammar and specifies an oracle j, a target value of 1.0, a GED interval, and a reference molecule. For RL on target j, we construct prompts by sampling references with replacement from the training partition, requiring activity $f_j(m_{\text{ref}}) \in [0.4, 0.8]$ on that target. For both RL and evaluation, we sample the GED lower bound ℓ and width w independently and uniformly from the integers 4 through 20, and set $h = \ell + w$, giving an upper bound of at most 40 edits. The lower bound requires a structural change to the input molecule, while the upper bound limits its extent. This restricted subset of the property-targeting grid

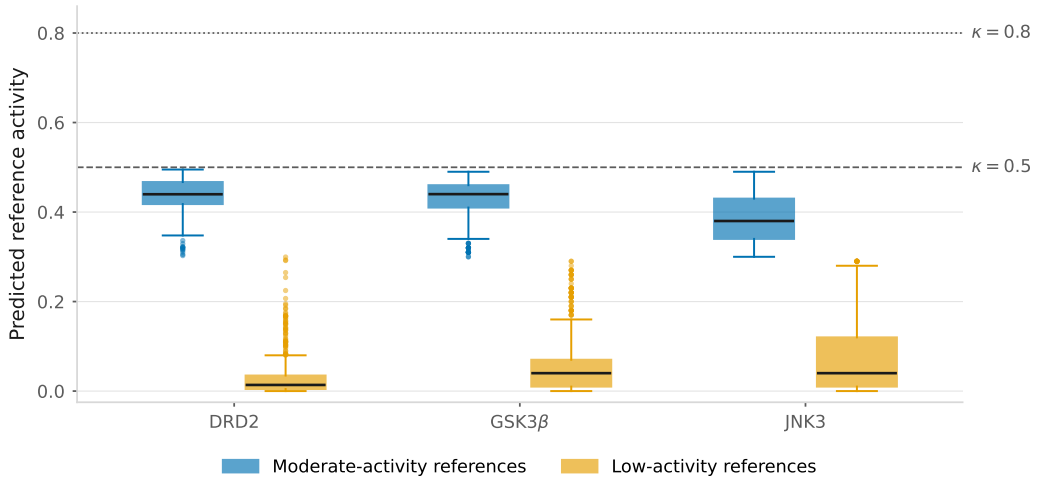

Figure A3: **Reference activities in the bioactivity benchmark.** Predicted scores $f_j(m_{\text{ref}})$ for 750 references per target and activity band: moderate $[0.3, 0.5)$ and low $[0.0, 0.3)$. Dashed and dotted lines mark the output activity-success thresholds $\kappa = 0.5$ and $\kappa = 0.8$, respectively.

loosely mimics scenarios such as lead optimization, where activity improvements are sought through controlled molecular modifications.

Evaluation design. For evaluation, we vary the reference activity while keeping the target value fixed. We sample references without replacement from the held-out partition of the same pool. Each target has 750 moderate-activity references with scores in $[0.3, 0.5)$ and 750 low-activity references with scores in $[0.0, 0.3)$. Evaluation references nevertheless remain scaffold-disjoint from training. All evaluation references start below the activity-success thresholds of 0.5 and 0.8, so success requires improvement over the input molecule.

Both reference-activity bands use the same GED-interval sampling rule. Figure A3 shows the reference-activity distributions for all three targets.

A.4 Success criteria and evaluation metrics

Graph edit distance. Graph edit distance (GED) counts the atom and bond operations that turn the reference molecule into the generated one. We compute it from the maximum common substructure (Dalke & Hastings, 2013): every atom and bond outside the common core costs one edit on each side, and within the core each atom that changes element or moves in or out of a ring, and each bond that changes order, costs one more. Symmetric cores admit many alignments, so we take the cheapest. The MCS is matched on topology alone, with atom and bond identity charged afterwards rather than decided by the matcher:

$$d_{\text{GED}}(m, m') = (N_m + N_{m'} - 2N_M) + (E_m + E_{m'} - 2E_M) + \Delta_M, \tag{A.1}$$

where N_x and E_x count the atoms and bonds in graph x, M is the selected common core, and Δ_M counts the atom- and bond-label mismatches within it.

We use a generous 60-second time limit per molecule pair, with timeouts occurring infrequently in the reliability audits. On timeout, we retain the best common core found, yielding an approximate GED that may overestimate the distance.

Success indicators. Both experiments use the decomposition in Equation (2). For a valid committed molecule, the property component s_{prop} is defined in Equation (1). With $\perp$ denoting an output without

a valid committed molecule, we set $s_{\text{prop}}(\perp; \mathcal{T}) = 0$ and define

$$s_{\text{struct}}(\hat{m}; \mathcal{T}) = \begin{cases} \mathbb{I}[\ell \le d_{\text{GED}}(m_{\text{ref}}, \hat{m}) \le h], & \hat{m} \in \mathcal{M}, \\ 0, & \hat{m} = \perp, \end{cases} \tag{A.2}$$

$$s_{\text{joint}}(\hat{m}; \mathcal{T}) = s_{\text{prop}}(\hat{m}; \mathcal{T})\, s_{\text{struct}}(\hat{m}; \mathcal{T}).$$

Structural success requires the requested GED interval, and joint success requires both the property and structural components. GED compliance is checked for every generation format; chemical validity alone does not imply structural success. Missing or invalid committed outputs receive zero for all indicators.

Property-targeting criterion. Property success requires $|f_j(\hat{m}) - v^\star| \le \varepsilon_j$. The tolerances are 0.5 ($\log P$), 8.0 (TPSA), 0.05 (Fsp^3), 0.05 (QED), and 0.4 (SA), fixed per property and used at evaluation.

Bioactivity criterion. An activity hit requires $f_j(\hat{m}) \ge \kappa$, with $\kappa = 0.5$ as the primary threshold and $\kappa = 0.8$ as a stricter criterion. This is the bioactivity definition of s_{prop}, not a separate success component. Joint success additionally requires the same structural condition as in property targeting. The activity scores are predictions, not experimental measurements.

Additional metrics. We additionally report commit rate (the fraction of rollouts yielding a committed molecule), validity, uniqueness per prompt and pooled across each target–condition pair, and DH-AUC over joint successes. For both experiments, DH-AUC is computed as in Section 2.5, using their respective joint-success criteria. It distinguishes repeated successes from structurally diverse solutions.

A.5 Training corpora and prompt sets

Supervised training uses only the five molecular properties in Appendix A.3.1: $\log P$, TPSA, Fsp^3, QED, and SA. Pretraining uses distorted property relations, followed by SFT on the original property values. Neither corpus contains bioactivity objectives or labels. For downstream bioactivity adaptation, we start from these property-pretrained models and apply RL directly, without bioactivity SFT. The construction below therefore concerns property-based supervised examples; RL and evaluation use prompt sets without supplied solutions.

A.5.1 Property relations for supervised training

Relation-distorted pretraining is designed to teach molecular syntax and the edit protocol without relying on a fixed relation between molecular structure and property values. For each search problem, we sample a transformation of the property's percentile ranks in the reference pool. We first convert the original property value $f_j(m)$ to its percentile using F_j, the empirical cumulative distribution function over that pool. We then apply the sampled transformation $\rho : [0, 1] \to [0, 1]$ to this percentile and use the quantile function F_j^{-1} to map the result back to the original property scale. These three steps define the distorted property value

$$\tilde{f}_j(m) = F_j^{-1}(\rho(F_j(f_j(m)))) \,.$$

Working with percentiles allows the same transformations to be used across all five properties, despite their different scales. SFT instead uses the original property values, $\tilde{f}_j = f_j$.

We sample relations from the six families in Table A2. They preserve, reverse, fold, or rearrange the property ordering, spanning rank correlations from -1 to $+1$. The pretraining corpus contains 704,290 recorded relation settings and 416,665 records from each family. The sampled relation is not disclosed to the model. It is fixed throughout each individual search and used consistently for fitness, solution selection, and hindsight relabelling.

A.5.2 Genetic search and solution selection

Genetic algorithms have a long history of successful use in molecular design, from early structure refinement to graph-based property optimization (Westhead et al., 1995; Jensen, 2019). Our use of

Table A2: Property-relation families used during pretraining. Each relation maps an empirical property quantile u to $\rho(u)$, with parameters sampled for each search problem. These transformations apply only to the five molecular properties, during pretraining data construction.

Family	Parameters	Effect on the true ordering
Warp: u^γ	$\log\gamma \sim \mathcal{U}(\log\frac{1}{3}, \log 3)$	preserved, respaced
Reflect: $1-u^\gamma$	$\log\gamma \sim \mathcal{U}(\log\frac{1}{3}, \log 3)$	reversed
Fold about a centre	$u_0 \sim \mathcal{U}(0.25, 0.75)$, $\gamma \sim \mathcal{U}(0.6, 2.0)$	folded onto itself
Sawtooth	teeth $\in \{2,3,4,5,7\}$, phase $\sim \mathcal{U}(0,1)$	repeated monotone runs
Permutation of b bins	$b \in \{8,16,32\}$, uniform permutation	shuffled between bins
Noisy monotone	scale $\sim \mathcal{U}(0.15, 0.8)$, frequency $\sim \mathcal{U}(3,12)$	locally perturbed

search-generated supervision is also related to Searchformer (Lehnert et al., 2024) and Stream of Search (Gandhi et al., 2024), which train models on serialized search processes rather than only final solutions. Here, the genetic algorithm provides molecular solutions and the paths used to reach them. We construct training trajectories from selected solution paths and short exploratory detours, rather than serializing the full population history.

Search procedure. For each reference molecule, target value, and GED interval, the genetic algorithm searches for a solution under $\tilde{f}_j$. Intervals are drawn from the grid in Appendix A.3.1, with sampling weights favouring intervals with lower estimated search yield. We also cap retained examples per interval to limit imbalance in the supervised corpus. The initial population contains the reference and nine valid one-step mutations. Each of at most 16 generations produces eight offspring, using tournaments of size three to select parents. Crossover is attempted with probability 0.4; otherwise, or if crossover fails, an offspring is produced by mutation. Mutations edit atoms, functional groups, rings, and fragments through addition, deletion, or replacement, and include multi-atom edits and ring opening. Each reproduction step proposes up to six alternatives and retains the best; the others are recorded as rejected siblings. The next population contains the 10 best parent and offspring molecules. Only distinct, chemically valid, connected molecules passing the structural filters are admitted.

Search objective. Each search requests a property value $v^\star$ and a GED interval $[\ell, h]$ relative to the reference molecule m_{ref}. The lower bound ℓ specifies the minimum structural change, and the upper bound h the maximum. For a candidate m, we write its GED from the reference as d; this is a distance between molecular graphs, not the number of actions in the trajectory. Its property value is $\tilde{f}_j(m)$, using the distorted relation for pretraining and the original property for SFT.

We rank candidates by the mean of two penalties: P_{GA} for property error and S_{GA} for falling outside the GED interval. Let $\delta_j > 0$ be the property error allowed without penalty during search. The fitness is

$$\begin{aligned} F_{\text{GA}}(m) &= \frac{1}{2}\left(P_{\text{GA}}(m) + S_{\text{GA}}(m)\right), \\ P_{\text{GA}}(m) &= \max\left(0, \frac{|\tilde{f}_j(m) - v^\star| - \delta_j}{\delta_j}\right), \\ S_{\text{GA}}(m) &= \frac{\max\{\ell - d, 0, d - h\}}{\max\{h - \ell, 1\}}, \qquad d = d_{\text{GED}}(m_{\text{ref}}, m). \end{aligned} \tag{A.3}$$

Lower is better. The property penalty is zero within δ_j of the target; beyond this, it measures the excess error in units of δ_j. The structural penalty is zero for $\ell \le d \le h$ and otherwise measures the distance to the nearest bound, divided by the interval width (or one when $\ell = h$). For example, a requested interval $[2, 4]$ gives zero structural penalty for GEDs from 2 to 4, and a penalty of 0.5 at either $d = 1$ or $d = 5$.

We set $\delta_j = 0.02B_j$, where $B_j = \overline{v}_j - \underline{v}_j$ is the span of fixed normalization bounds: $[-3, 8]$ for $\log P$, $[0, 250]$ for TPSA, $[0, 1]$ for Fsp^3 and QED, and $[1, 10]$ for SA. These bounds set the penalty scale, not the training target ranges in Appendix A.3.1. The search tolerance δ_j is distinct from the final

acceptance tolerance ε_j in Appendix A.4. This fitness ranks candidates during corpus construction; it is not the reward used for RL.

Solution selection and hindsight relabelling. A candidate solution must satisfy $|\tilde{f}_j(m) - v^\star| \leq \varepsilon_j$, lie within the requested GED interval, and have a primary-parent lineage of one to eight genetic transitions. We then replace the sampled target with the attained value $\tilde{f}_j(m)$, following hindsight relabelling (Andrychowicz et al., 2017), while retaining the original GED interval. The relabelled target is the distorted property value for pretraining and the original property value for SFT. The training-range and held-out-gap retention criterion applies to this relabelled value, as specified in Appendix A.3.1.

A.5.3 Trajectory construction and validation

Selected path. We trace each selected molecule's primary-parent lineage back to the reference and convert its transitions into actions from the model's edit vocabulary. For crossover, the second parent contributes to the search but its ancestry is not expanded in the trajectory. A transition may require several model actions; examples are discarded if a transition cannot be represented and executed in the inference environment.

Exploration and recovery. Rejected siblings supply short detours from the selected path. At an eligible transition, with probability 0.5 and at most twice per trajectory, we execute a clearly inferior sibling, backtrack, and continue along the selected path. These detours are alternatives evaluated during search, not arbitrary edits. The valid but inferior edits and the subsequent backtracking decisions are supervised.

We also insert invalid action attempts before eligible valid actions, with probability 0.35 and at most twice per trajectory. Examples include an unavailable attachment site, an out-of-range atom index, or an incorrect functional group. An attempt is included only if the environment confirms that it fails. The molecular state remains unchanged and the correct action follows. The corrupted action and its failure observation are masked from the supervised loss, exposing the model to recovery contexts without training it to reproduce the inserted error.

Commit and replay. After reaching the solution, a trajectory may attempt an invalid edit or a valid but inferior continuation before committing the saved solution state. The committed molecule therefore need not be the last state explored. We replay the complete trajectory from the reference and retain it only if the actions, backtracks, and state selections execute as intended and the committed molecule matches the selected solution. The same underlying solutions are serialized into all four generation formats (Appendix B).

A.5.4 Dataset sizes

Table A3 summarizes the datasets shared across model sizes and generation formats. Pretraining and SFT examples pair a task prompt with a verified solution; RL and evaluation datasets contain only task prompts. Supervised training covers the five molecular properties. Bioactivity adaptation starts from these pretrained models without an additional supervised corpus.

We construct 100,000 RL prompts per property or bioactivity target by sampling reference molecules with replacement from the corresponding eligible training pool, respecting the scaffold split and reference exclusions in Appendix A.2. Property targets are sampled from $\mathcal{R}_j \setminus \mathcal{G}_j$. For bioactivity, references must score in $[0.4, 0.8]$ on the requested oracle, and the target value is fixed at 1.0. This gives 500,000 prompts across the five properties and 300,000 across the three bioactivity targets. These counts refer to prompts, not distinct reference molecules.

Property evaluation uses 500 prompts per property in each of the three target-value regimes (in-distribution, interpolation gap, and extrapolation), giving 1,500 per property and 7,500 overall. Bioactivity evaluation uses 750 prompts per target in each of the moderate- and low-activity reference bands, giving 1,500 per target and 4,500 overall. Reference selection and data separation follow Appendices A.3 and A.2. RL training budgets are reported separately in Table C4.

Table A3: Dataset sizes in total and per property or bioactivity target. RL entries report the selected prompt sets.

Dataset	Total	Per property/target
Property targeting		
Pretraining examples	2,499,990	499,998
SFT examples	50,000	10,000
RL prompts	500,000	100,000
Evaluation prompts	7,500	1,500
Bioactivity adaptation		
RL prompts	300,000	100,000
Evaluation prompts	4,500	1,500

B GENERATION FORMATS AND INTERACTION PROTOCOLS

B.1 GENERATION FORMATS

The four generation formats receive the same task prompt $\mathbf{x}_{\mathcal{T}}$, specifying an objective, a target value, a GED interval, and a reference molecule. They differ in whether the response represents an extended molecular trajectory and whether an environment executes the edits. Table B1 summarizes what the model generates and what the environment returns.

Table B1: Generation formats and the content exchanged during inference.

Format	Model-generated content	Environment feedback
Direct SMILES	Final molecular SMILES	None
Agentic edit program	Edit arguments and site selections, with at most two edits	Site menus; no full molecular states
CoM (self-generated)	Action labels and intermediate molecular SMILES	None
Agentic CoM	Edit arguments, site selections, and search-control decisions	Site menus, executed molecular states, and state identifiers

Direct SMILES. The model generates only the final molecular SMILES, as in Section 2.2. The complete response is decoded to obtain the output molecule; generation involves neither intermediate states nor environment interaction.

Agentic edit program. This is the agentic counterpart to direct prediction. Instead of writing the answer as a SMILES string, the model specifies a short program that transforms the reference into the output molecule, with a maximum of two molecular edits in both training and inference. Supervised programs are derived from the reference–product pair rather than the full search trajectory. Each edit includes a site query and selection: the environment supplies the menu and executes the selected edit, retaining the updated molecule internally without returning its full SMILES. The final output is the product of this short program, not the endpoint of an extended CoM trajectory.

Self-generated CoM. The model generates action labels and intermediate molecular SMILES as one uninterrupted continuation, as in Equation (5). These labels describe the intended transitions; they do not include the fragment arguments and site selections needed for execution. No environment checks whether a generated transition agrees with its label. The last stated molecule before the commit token is taken as the output. Generation is subject to the token budget, but no turn limit applies.

Agentic CoM. The model generates executable edits, site selections, and search-control decisions. The environment supplies the resulting molecular states, as in Equation (6), so subsequent decisions condition on executed rather than predicted transitions. Within the generation budget, the model can continue editing, backtrack to an earlier state, or commit a state as its final output.

For supervised training, prompt tokens and environment-generated content are context only and do not contribute to the loss. Model-generated response tokens are supervised, including valid but

inferior exploratory edits and subsequent backtracking decisions. Injected invalid action attempts and their failure observations are excluded from the loss; the recovery actions that follow are supervised (Appendix A.5.3).

B.2 Molecular actions and addressing

The agentic formats express edits using SMILES fragments and attachment sites. Wildcard atoms, denoted by $\star$, mark connections between an edited fragment and the rest of the molecule. A one-point edit uses one attachment site; a bridge uses two. Table B2 lists the molecular edits and search-control actions.

Table B2: Molecular-edit and search-control actions in the agentic protocol. The agentic edit program uses a short edit sequence; Agentic CoM additionally supports an extended trajectory with backtracking.

Action	Arguments	Effect
Add	New fragment	Graft the fragment onto the selected anchor
Delete	Existing fragment	Remove the selected molecular region
Replace	Existing and new fragments	Replace a region at one attachment point
Bridge	Existing and new fragments	Replace a region connected through two attachment points
Backtrack	State identifier	Return to a previously explored molecular state
Final	Optional state identifier	Select a state as the final output

Each atom in the reference molecule receives a persistent identifier. An atom retains this identifier across subsequent edits as long as it remains in the molecule, while newly introduced atoms receive new identifiers. These identifiers refer to atom identity rather than position in a canonical SMILES string, whose ordering may change after an edit. State identifiers are separate and refer to complete molecular states; they are used by the backtrack and commit actions.

B.3 Interaction protocol

Both agentic formats separate the proposed transformation from the choice of where to apply it. Each edit follows the same interaction loop:

1. The model emits an action and its fragment arguments, followed by `<|sites|>`, without yet specifying an attachment site.
2. The environment enumerates candidate anchor sets on the current molecule. For addition, these are atoms with sufficient available valence. For deletion, replacement, and bridge, they are obtained by matching the specified existing fragment and identifying its attachment points.

   Candidates are trial-executed; invalid products are discarded and sites producing the same canonical product are merged. The environment returns a deterministically ordered menu of up to 64 distinct legal outcomes.
3. Each offered anchor is shown with its persistent atom identifier and a description of its local chemical neighbourhood. The description starts at a one-bond radius and expands up to four bonds to distinguish the offered anchors whenever possible.
4. The model selects an offered site using `<|pick|>`. A one-anchor edit requires one pick, whereas a bridge requires two ordered picks. The environment executes the selected edit and updates its current molecular state.

Execution feedback. If a proposed edit cannot be executed, the environment returns `<|invalid|>` and leaves the current molecular state unchanged. After a successful edit, Agentic CoM receives the resulting molecule and its state identifier.

None of the four formats receives additional intermediate readings in the experiments reported here. Such readings could be added to the observation interface in future work, informing better search decisions.

Format	SFT sequence
Direct SMILES	<|bos|> <|sa|> <3> <.> <5> <9> <|ged|> <1> <|to|> <3> <|ref|> C # C C C N 1 C C O C 2 C C 2 1 <|com|> C C C N 1 C C O C 2 C C 2 1 <|eos|>
Agentic edit program	<|bos|> <|sa|> <3> <.> <5> <9> <|ged|> <1> <|to|> <3> <|ref|> C # C C C N 1 C C O C 2 C C 2 1 <|com|> <|del|> * # C <|sites|> <|at|> <a2> C (# C) C <|pick|> <a2> <|final|> <|eos|>
CoM (self-generated)	<|bos|> <|sa|> <3> <.> <5> <9> <|ged|> <1> <|to|> <3> <|ref|> C # C C C N 1 C C O C 2 C C 2 1 <|com|> <|del|> C C C N 1 C C O C 2 C C 2 1 <|del|> C C C <|backtrack|> C C C N 1 C C O C 2 C C 2 1 <|final|> <|eos|>
Agentic CoM	<|bos|> <|sa|> <3> <.> <5> <9> <|ged|> <1> <|to|> <3> <|ref|> C # C C C N 1 C C O C 2 C C 2 1 <|com|> <|del|> * # N <|sites|> <|invalid|> <|del|> * # C <|sites|> <|at|> <a2> C (# C) C <|pick|> <a2> <|obs|> C C C N 1 C C O C 2 C C 2 1 <|id|> <s1> <|del|> * N 1 C C O C 2 C C 2 1 <|sites|> <|at|> <a4> C (C) N <|pick|> <a4> <|obs|> C C C <|id|> <s2> <|del|> * N 1 N C O C 2 C C 2 1 <|sites|> <|invalid|> <|backtrack|> <|id|> <s1> <|final|> <|eos|>

Shared final molecule: `CCCN1CCOC2CC21`.

Supervised Masked (context only)

Figure B1: One matched SFT example across four formats. Each box is one tokenizer token: blue tokens are supervised; grey tokens are masked from the loss but remain visible as context. Every row shows the complete prompt and response. In Agentic CoM, invalid attempts and environment feedback are masked, whereas valid detours and backtracking are supervised.

Backtracking and commitment. In Agentic CoM, backtrack moves the current search head to a selected state, which was previously explored in its trace. Commit terminates the trajectory and selects either an explicitly referenced state or the current search head. The committed molecule therefore need not be the most recently visited state.

B.4 Matched SFT examples

Figure B1 shows one matched SFT example as four token sequences, with the same prompt and final molecule. Each box represents one token, coloured according to its stored loss mask. Masked tokens remain part of the model's context but do not contribute to the supervised loss.

C Model and training details

We follow common language-model training paradigms. Training begins with relation-distorted pretraining on the five molecular properties. For property targeting, we then apply RL either directly or after supervised finetuning (SFT) on a limited set of examples with the original, undistorted property objectives. Bioactivity adaptation follows the direct route, using RL to adapt the property-pretrained models to each activity objective. Within each model size and training route, the formats share the architecture, tokenizer, and task prompts; their response formats and interaction protocols are described in Appendix B.

C.1 MODEL AND TOKENIZER

Architecture. All variants are randomly initialized decoder-only transformers following the Qwen3 architecture (Yang et al., 2025). We train them from scratch on our molecular corpora; we do not initialize them from pretrained Qwen3 language-model weights. Table C1 gives the two model sizes. Both use RMSNorm, rotary position embeddings, and tied input and output embeddings. The architecture is fixed across generation formats within each size, and the tokenizer is shared across both sizes.

Table C1: Model configurations shared across generation formats. Parameter counts include tied embeddings once; both sizes use the same tokenizer.

	1M family	20M family
Parameters	1,019,264	20,530,432
Transformer layers	6	24
Hidden dimension	128	256
Attention heads / KV heads	2 / 2	4 / 4
Head dimension	64	64
MLP intermediate dimension	256	768
Vocabulary size	264	264
Supervised training context	4,096	4,096
RL / evaluation sequence limit	6,144	6,144

Tokenizer. All formats share a fixed 264-token word-level vocabulary with no learned subword merges. It preserves SMILES and protocol boundaries, including the atom identifiers used by the agentic formats. Of the 264 tokens, 188 cover SMILES syntax, element symbols, and aromatic forms, and 76 are structural markers for task framing, property names, molecular-edit and search-control actions, and the environment interface. Numbers are written digit by digit in three disjoint alphabets according to their referent: values (<0>–<9>), atom identifiers (<a0>–<a9>), and state identifiers (<s0>–<s9>). Holding one vocabulary fixed across formats also makes supervised-token budgets directly comparable.

C.2 SUPERVISED PRETRAINING AND FINETUNING

Training stages. Pretraining is intended to teach molecular syntax and format-specific generation protocols, including search behaviour in the trajectory-based formats. The relation-distorted corpus (Appendix A.5.1) applies a randomly sampled transformation to property values for each search problem. The transformation remains fixed throughout that search: values are not assigned independently at random to individual molecules. Candidate molecules can therefore still be compared under a consistent objective, and edits can improve or worsen that objective. This preserves meaningful search examples while varying the relation between molecular structure and property values across problems. We train one model per size and format. SFT then adapts each pretrained model to the original, undistorted property objectives using a limited set of 50,000 examples, with 10,000 per property (Table A3). Both stages use matched task–solution pairs across formats, serialized as described in Appendix B. Bioactivity objectives are introduced only during RL; neither pretraining nor SFT uses bioactivity examples.

Supervised objective. Pretraining and SFT use the same class-balanced next-token objective, with the masking rules in Appendix B. The loss groups tokens into four possible classes: molecular tokens (SMILES and fragments), control tokens (edits, site selections, and state references), conditioning tokens (property names and values), and termination tokens (commit and end of sequence). We first average the token losses within each non-empty class, then average these class losses with equal weight. This gives infrequent control and termination decisions their own contribution rather than weighting them only by their frequency among molecular tokens.

Let $\mathbf{s}^{(n)}$ be a serialized example and $\mathcal{I}_c$ the set of unmasked target positions (n, i) assigned to class c in the current batch. Writing $\mathcal{C}_+$ for the classes with at least one such token, the shared objective is

$$\mathcal{L}_{\text{sup}} = -\frac{1}{|\mathcal{C}_+|} \sum_{c \in \mathcal{C}_+} \frac{1}{|\mathcal{I}_c|} \sum_{(n,i) \in \mathcal{I}_c} \log p_\theta \left(s_i^{(n)} \mid s_{<i}^{(n)} \right). \tag{C.1}$$

Only classes containing unmasked targets contribute. For example, Direct SMILES uses molecular and termination losses; a conditioning class is inactive when property names and values occur only in masked prompts or observations. Thus the four-class configuration does not imply four active losses for every format or batch.

Optimization. Muon (Jordan et al., 2024) optimizes the two-dimensional weight matrices in the transformer blocks. AdamW optimizes the remaining parameters, including the tied embeddings and normalization parameters. Both optimizers use linear learning-rate warm-up followed by cosine decay to zero, with the peak rates in Table C2. AdamW uses $\beta_1 = 0.9$ and $\beta_2 = 0.95$; Muon momentum ramps from 0.85 to 0.95 over the first 300 optimizer steps. Both training stages use a 4,096-token sequence limit, gradient clipping at 1.0 and weight decay 0.1. We use the final checkpoint at the configured optimizer-step budget.

Table C2: Supervised optimization settings shared by the 1M and 20M models. Format-specific step budgets are given in Table C3.

	Pretraining	Finetuning
Global batch size	64	32
Peak Muon learning rate	5×10^{-3}	1×10^{-3}
Peak AdamW learning rate	1×10^{-4}	2.5×10^{-5}
Warm-up fraction	0.01	0.05

Matched supervised-token budgets. We match supervised-token exposure across formats: approximately 900M tokens during pretraining and 98M during SFT. Since formats contain different numbers of supervised tokens per example, their optimizer-step budgets differ (Table C3). This matches supervised tokens, since context lengths and environment execution differ across formats.

Table C3: Matched supervised-token budgets for both model sizes. Global batch sizes are 64 for pretraining and 32 for finetuning. Optimizer-step counts differ across formats because examples contain different numbers of supervised tokens.

	Pretraining		Finetuning	
Format	Steps	Sup. tokens	Steps	Sup. tokens
Direct SMILES	315,842	≈900M	68,403	≈98M
Agentic edit program	260,174	≈900M	54,967	≈98M
CoM (self-generated)	78,125	≈900M	15,625	≈98M
Agentic CoM	205,621	≈900M	41,596	≈98M

C.3 Reinforcement learning

We train a separate policy for each property or bioactivity target, starting from the checkpoints specified at the beginning of this section. We use Group Relative Policy Optimization (GRPO) (Shao et al., 2024), implemented with NeMo RL (NeMo, 2025), with a leave-one-out baseline and normalized within-prompt advantages. Rewards are computed from the committed molecules. Only model-generated tokens contribute to the policy loss; prompts and environment observations remain context only.

Reward. Both experiments use the same combination of structural and objective credit. For a valid committed molecule $\hat{m}$,

$$r(\hat{m}; \mathcal{T}) = \frac{s_{\text{struct}}(\hat{m}; \mathcal{T}) + q(\hat{m}; \mathcal{T})}{2 \ \text{mult}_{\mathcal{T}}(\hat{m})}. \tag{C.2}$$

Here, s_{struct} is the GED-interval indicator defined in Appendix A.4, q is the objective credit defined below, and $\text{mult}_{\mathcal{T}}(\hat{m})$ counts occurrences of the same canonical molecule within the prompt's rollout group. Dividing by this count discourages repeated identical outputs. Invalid outputs and rollouts without a valid commit receive zero reward. Structural and objective credit are additive, so satisfying either component can earn partial reward; joint success at evaluation still requires both.

For property targeting, objective credit decreases smoothly with the absolute error from the requested value:

$$q(\hat{m}; \mathcal{T}) = \left[1 + \exp\left(50 \left[\frac{|f_j(\hat{m}) - v^\star|}{B_j} - 0.05\right]\right)\right]^{-1}. \tag{C.3}$$

The normalization spans B_j are those defined in Appendix A.5.2, not the training target ranges. This shaping is distinct from the property-specific evaluation tolerance ε_j. For bioactivity, we use the predicted activity directly, $q(\hat{m}; \mathcal{T}) = f_j(\hat{m})$; the activity thresholds $\kappa \in \{0.5, 0.8\}$ are used only for evaluation. The reward definition is shared across generation formats within each experiment.

Optimization and budgets. The primary runs use $P = 256$ prompts per RL iteration and $G = 16$ rollouts per prompt, giving $PG = 4{,}096$ trajectories. Each iteration uses two global optimizer batches of 2,048 sequences, for 500 iterations and nominally 1,000 optimizer updates per target-specific policy. These settings apply to both model sizes, both property-targeting initialization routes, and bioactivity adaptation. Table C4 also gives the $G = 64$ comparison.

We use AdamW with weight decay 0.01, gradient clipping at 1.0, and PPO ratio clipping of 0.2. The learning rate increases linearly from 10^{-6} to 10^{-5} over 50 scheduler steps and then remains constant. No reference-policy KL penalty is applied. The training micro-batch size is 16 sequences, and rollouts are generated in batches of 512.

Table C4: RL budgets per target-specific policy. The $G = 16$ column applies to both model sizes, both property-targeting initialization routes, and bioactivity RL from pretraining. The $G = 64$ comparison uses 1M property-targeting models after SFT. Training group size is distinct from the 128 outputs per evaluation prompt. Processed-prompt counts include repeated use of the same dataset entry. Optimizer-update counts are nominal counts implied by the configured batches.

	$G = 16$	$G = 64$
Prompts per RL iteration (P)	256	64
Rollouts per prompt (G)	16	64
Trajectories per RL iteration (PG)	4,096	4,096
RL iterations	500	500
Global optimizer batch	2,048	2,048
Optimizer updates per run	1,000	1,000
AdamW learning rate	10^{-5}	10^{-5}
Warm-up scheduler steps	50	50
Reference-policy KL coefficient	0	0
Evaluation outputs per prompt	128	128

Rollout settings. Sampling uses temperature 1.0 without nucleus or top-k truncation. The total sequence limit is 6,144 tokens, including the prompt, generated response, and environment observations. The implementation permits one turn for Direct SMILES and self-generated CoM, up to six for the agentic edit program, and 48 for Agentic CoM. These calls include site queries and selections; they are not counts of molecular edits. The edit program retains its two-edit budget. The same sampling and sequence/interaction limits are used at evaluation.

D Extended results

We report property-targeting performance, comparisons of RL initialization and group size, trajectory behaviour, downstream bioactivity adaptation, and directional property optimization on TOMG-Bench.

Reporting conventions. For the property and bioactivity performance tables, each task has 128 sampled outputs. Success rates and DH-AUC include all outputs, counting missing or invalid commits as failures. All traces are evaluated at the format-specific turn limits; commits beyond these limits count as failures. Means and standard errors (SEs) are computed across evaluation tasks, with equal numbers of tasks per objective. Each training setting has one seed, so the SEs do not measure variation across training runs. Values are rounded to three decimal places; displayed zeros may reflect rounding. Bold highlights means within one SE of the best mean for each metric and condition, separately for each model size and training setting. We use the best result's SE and unrounded values; each highlighted result retains its own SE.

Validity is the fraction of outputs with a valid committed molecule. Uniqueness is the mean fraction of distinct molecules among a task's valid outputs, excluding tasks with no valid output. Median trace tokens are pooled over all attempts, including failures; they exclude the initial prompt but include environment observations. These medians are descriptive and have no SE. Success criteria and DH-AUC follow Section 2.5 and Appendix A.4; TOMG-Bench uses its own reporting protocol below.

D.1 Property-targeting performance

After RL following SFT, Agentic CoM has the highest joint success on in-distribution and interpolation gap targets at both model sizes, and the highest DH-AUC in all three regimes. Increasing model size raises its in-distribution joint success from 0.460 to 0.553 and its gap success from 0.495 to 0.576. Extrapolation remains harder: the edit program has the highest joint success at 1M (0.134 versus 0.121 for Agentic CoM), while Agentic CoM has the highest mean at 20M (0.155 versus 0.148).

Figure D1 complements the main 1M figure, and Figure D2 compares the two sizes directly across training stages. Tables D1–D6 report both sizes after pretraining, SFT, and RL after SFT, including validity, uniqueness, success components, DH-AUC, and trace length. Each regime contains 2,500 tasks, with 500 per property. Results for RL directly from pretraining follow in the initialization comparison.

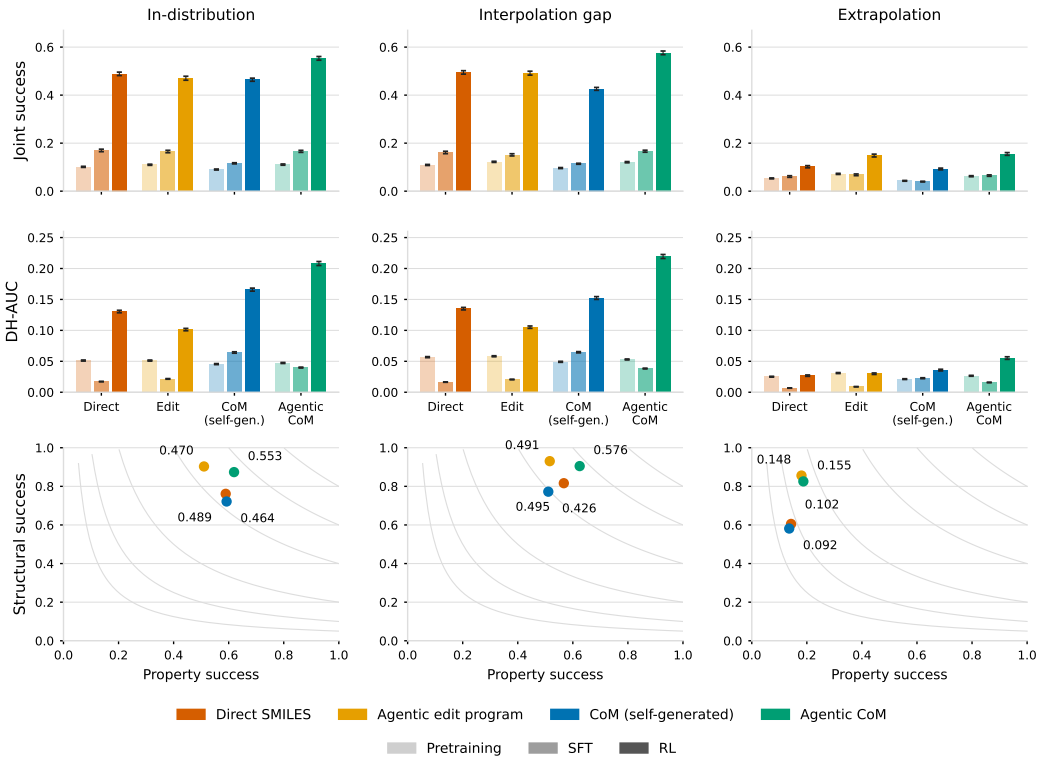

Figure D1: **Optimization performance and the trade-off between task constraints. 20M models. Top:** Joint success and DH-AUC across four generators, three training stages, and three target regimes. Colour identifies the generation format and saturation identifies the training stage; RL denotes GRPO after SFT with $G = 16$. **Bottom:** Property and structural success after GRPO. Each point is labelled with its measured joint success. Grey curves show constant products of the two marginal success rates, not constant joint success. Error bars are ± 1 SE over evaluation tasks per regime. The corresponding 1M results use the same axis limits (Figure 3).

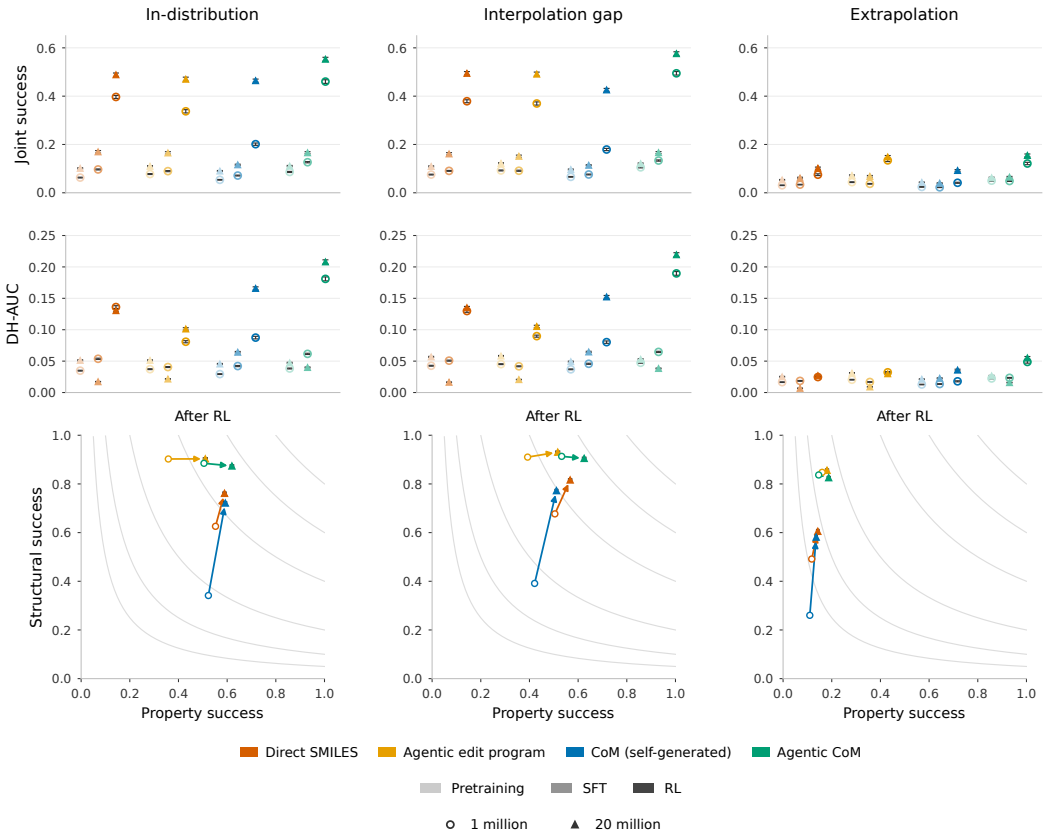

Figure D2: **Property-targeting performance at both model sizes. Top:** Joint success and DH-AUC across training stages. Open circles denote 1M models and filled triangles denote 20M models. Colour identifies the generation format; within each colour group, pretraining, SFT and RL appear from left to right with increasing saturation. RL denotes GRPO after SFT with $G = 16$. **Bottom:** Property and structural success after RL, with arrows pointing from 1M to 20M for each format. Grey curves show constant products of the marginal success rates, not constant joint success. Error bars are ± 1 SE over evaluation tasks per regime.

Table D1: 1M models after pretraining. Values are means ± 1 SE over evaluation tasks per regime; trace tokens are medians. Reporting conventions are given in Appendix D.

Metric	Format	ID	Gap	Extrapolation
Validity	Direct SMILES	0.847 ± 0.003	0.864 ± 0.002	0.763 ± 0.005
	Agentic edit program	0.765 ± 0.003	0.780 ± 0.002	0.703 ± 0.003
	CoM (self-generated)	0.739 ± 0.003	0.759 ± 0.003	0.659 ± 0.005
	Agentic CoM	$\mathbf{0.980 \pm 0.000}$	$\mathbf{0.981 \pm 0.000}$	$\mathbf{0.976 \pm 0.001}$
Uniqueness	Direct SMILES	0.891 ± 0.003	0.882 ± 0.003	0.896 ± 0.003
	Agentic edit program	0.889 ± 0.002	0.885 ± 0.002	0.874 ± 0.002
	CoM (self-generated)	$\mathbf{0.902 \pm 0.002}$	$\mathbf{0.895 \pm 0.002}$	$\mathbf{0.912 \pm 0.002}$
	Agentic CoM	0.831 ± 0.002	0.831 ± 0.002	0.818 ± 0.002
Structural success	Direct SMILES	0.407 ± 0.005	0.430 ± 0.005	0.327 ± 0.005
	Agentic edit program	0.467 ± 0.004	0.483 ± 0.004	0.413 ± 0.004
	CoM (self-generated)	0.341 ± 0.005	0.364 ± 0.005	0.273 ± 0.005
	Agentic CoM	$\mathbf{0.527 \pm 0.005}$	$\mathbf{0.534 \pm 0.005}$	$\mathbf{0.480 \pm 0.005}$
Property success	Direct SMILES	0.131 ± 0.002	0.159 ± 0.002	0.068 ± 0.002
	Agentic edit program	0.124 ± 0.002	0.152 ± 0.002	0.072 ± 0.002
	CoM (self-generated)	0.113 ± 0.002	0.139 ± 0.002	0.054 ± 0.002
	Agentic CoM	$\mathbf{0.161 \pm 0.003}$	$\mathbf{0.201 \pm 0.003}$	$\mathbf{0.102 \pm 0.003}$
Joint success	Direct SMILES	0.063 ± 0.002	0.075 ± 0.002	0.031 ± 0.001
	Agentic edit program	0.078 ± 0.002	0.092 ± 0.002	0.044 ± 0.002
	CoM (self-generated)	0.054 ± 0.001	0.066 ± 0.002	0.024 ± 0.001
	Agentic CoM	$\mathbf{0.086 \pm 0.002}$	$\mathbf{0.105 \pm 0.002}$	$\mathbf{0.050 \pm 0.002}$
DH-AUC	Direct SMILES	0.035 ± 0.001	0.043 ± 0.001	0.017 ± 0.001
	Agentic edit program	0.037 ± 0.001	0.045 ± 0.001	0.020 ± 0.001
	CoM (self-generated)	0.029 ± 0.001	0.037 ± 0.001	0.013 ± 0.000
	Agentic CoM	$\mathbf{0.038 \pm 0.001}$	$\mathbf{0.047 \pm 0.001}$	$\mathbf{0.022 \pm 0.001}$
Median trace tokens	Direct SMILES	45	43	53
	Agentic edit program	154	139	252
	CoM (self-generated)	108	101	120
	Agentic CoM	348	331	417

Table D2: 20M models after pretraining. Values are means ± 1 SE over evaluation tasks per regime; trace tokens are medians. Reporting conventions are given in Appendix D.

Metric	Format	ID	Gap	Extrapolation
Validity	Direct SMILES	0.937 ± 0.002	0.947 ± 0.001	0.879 ± 0.004
	Agentic edit program	$\mathbf{0.999 \pm 0.000}$	$\mathbf{1.000 \pm 0.000}$	$\mathbf{0.998 \pm 0.000}$
	CoM (self-generated)	0.908 ± 0.002	0.918 ± 0.002	0.847 ± 0.004
	Agentic CoM	0.985 ± 0.000	0.986 ± 0.000	0.982 ± 0.000
Uniqueness	Direct SMILES	0.882 ± 0.003	0.866 ± 0.003	0.885 ± 0.003
	Agentic edit program	0.808 ± 0.003	0.818 ± 0.002	0.767 ± 0.003
	CoM (self-generated)	$\mathbf{0.893 \pm 0.002}$	$\mathbf{0.886 \pm 0.002}$	$\mathbf{0.903 \pm 0.002}$
	Agentic CoM	0.840 ± 0.003	0.841 ± 0.003	0.817 ± 0.003
Structural success	Direct SMILES	0.605 ± 0.005	0.621 ± 0.005	0.524 ± 0.005
	Agentic edit program	0.630 ± 0.004	0.642 ± 0.004	0.582 ± 0.005
	CoM (self-generated)	0.529 ± 0.005	0.546 ± 0.005	0.442 ± 0.005
	Agentic CoM	$\mathbf{0.640 \pm 0.005}$	$\mathbf{0.648 \pm 0.005}$	$\mathbf{0.588 \pm 0.005}$
Property success	Direct SMILES	0.155 ± 0.003	0.172 ± 0.003	0.085 ± 0.002
	Agentic edit program	$\mathbf{0.173 \pm 0.003}$	$\mathbf{0.196 \pm 0.003}$	$\mathbf{0.117 \pm 0.003}$
	CoM (self-generated)	0.152 ± 0.002	0.163 ± 0.002	0.078 ± 0.002
	Agentic CoM	$\mathbf{0.171 \pm 0.003}$	0.188 ± 0.003	0.105 ± 0.003
Joint success	Direct SMILES	0.101 ± 0.002	0.109 ± 0.002	0.053 ± 0.002
	Agentic edit program	$\mathbf{0.110 \pm 0.002}$	$\mathbf{0.122 \pm 0.002}$	$\mathbf{0.072 \pm 0.002}$
	CoM (self-generated)	0.090 ± 0.002	0.096 ± 0.002	0.043 ± 0.002
	Agentic CoM	$\mathbf{0.111 \pm 0.003}$	$\mathbf{0.121 \pm 0.003}$	0.062 ± 0.002
DH-AUC	Direct SMILES	$\mathbf{0.051 \pm 0.001}$	0.057 ± 0.001	0.025 ± 0.001
	Agentic edit program	$\mathbf{0.051 \pm 0.001}$	$\mathbf{0.058 \pm 0.001}$	$\mathbf{0.031 \pm 0.001}$
	CoM (self-generated)	0.045 ± 0.001	0.049 ± 0.001	0.021 ± 0.001
	Agentic CoM	0.047 ± 0.001	0.053 ± 0.001	0.027 ± 0.001
Median trace tokens	Direct SMILES	46	44	55
	Agentic edit program	103	101	116
	CoM (self-generated)	121	118	142
	Agentic CoM	349	331	416

Table D3: 1M models after supervised finetuning. Values are means ± 1 SE over evaluation tasks per regime; trace tokens are medians. Reporting conventions are given in Appendix D.

Metric	Format	ID	Gap	Extrapolation
Validity	Direct SMILES	0.683 ± 0.004	0.724 ± 0.003	0.581 ± 0.005
	Agentic edit program	0.614 ± 0.004	0.646 ± 0.004	0.560 ± 0.004
	CoM (self-generated)	0.633 ± 0.004	0.670 ± 0.003	0.546 ± 0.005
	Agentic CoM	$\mathbf{0.885 \pm 0.002}$	$\mathbf{0.900 \pm 0.001}$	$\mathbf{0.878 \pm 0.002}$
Uniqueness	Direct SMILES	0.902 ± 0.002	0.900 ± 0.002	0.905 ± 0.003
	Agentic edit program	0.800 ± 0.004	0.809 ± 0.004	0.740 ± 0.005
	CoM (self-generated)	$\mathbf{0.942 \pm 0.002}$	$\mathbf{0.937 \pm 0.002}$	$\mathbf{0.953 \pm 0.002}$
	Agentic CoM	0.871 ± 0.003	0.860 ± 0.003	0.844 ± 0.003
Structural success	Direct SMILES	0.295 ± 0.004	0.322 ± 0.004	0.226 ± 0.004
	Agentic edit program	0.324 ± 0.004	0.348 ± 0.004	0.278 ± 0.004
	CoM (self-generated)	0.225 ± 0.004	0.260 ± 0.004	0.168 ± 0.004
	Agentic CoM	$\mathbf{0.413 \pm 0.005}$	$\mathbf{0.427 \pm 0.005}$	$\mathbf{0.374 \pm 0.005}$
Property success	Direct SMILES	0.215 ± 0.003	0.200 ± 0.002	0.081 ± 0.002
	Agentic edit program	0.160 ± 0.003	0.166 ± 0.003	0.071 ± 0.002
	CoM (self-generated)	0.177 ± 0.002	0.182 ± 0.002	0.057 ± 0.002
	Agentic CoM	$\mathbf{0.248 \pm 0.003}$	$\mathbf{0.267 \pm 0.003}$	$\mathbf{0.110 \pm 0.003}$
Joint success	Direct SMILES	0.097 ± 0.002	0.091 ± 0.002	0.034 ± 0.001
	Agentic edit program	0.090 ± 0.002	0.091 ± 0.002	0.037 ± 0.002
	CoM (self-generated)	0.071 ± 0.002	0.076 ± 0.002	0.023 ± 0.001
	Agentic CoM	$\mathbf{0.127 \pm 0.003}$	$\mathbf{0.133 \pm 0.003}$	$\mathbf{0.049 \pm 0.002}$
DH-AUC	Direct SMILES	0.054 ± 0.001	0.051 ± 0.001	0.019 ± 0.001
	Agentic edit program	0.041 ± 0.001	0.042 ± 0.001	0.017 ± 0.001
	CoM (self-generated)	0.042 ± 0.001	0.046 ± 0.001	0.014 ± 0.001
	Agentic CoM	$\mathbf{0.062 \pm 0.001}$	$\mathbf{0.065 \pm 0.001}$	$\mathbf{0.023 \pm 0.001}$
Median trace tokens	Direct SMILES	46	44	54
	Agentic edit program	240	197	344
	CoM (self-generated)	199	175	228
	Agentic CoM	648	574	739

Table D4: 20M models after supervised finetuning. Values are means ±1 SE over evaluation tasks per regime; trace tokens are medians. Reporting conventions are given in Appendix D.

Metric	Format	ID	Gap	Extrapolation
Validity	Direct SMILES	$\mathbf{0.932 \pm 0.003}$	$\mathbf{0.941 \pm 0.003}$	0.854 ± 0.005
	Agentic edit program	0.911 ± 0.003	0.921 ± 0.003	0.861 ± 0.004
	CoM (self-generated)	0.795 ± 0.003	0.811 ± 0.003	0.711 ± 0.005
	Agentic CoM	0.899 ± 0.003	0.918 ± 0.002	$\mathbf{0.880 \pm 0.003}$
Uniqueness	Direct SMILES	0.211 ± 0.003	0.201 ± 0.003	0.254 ± 0.004
	Agentic edit program	0.250 ± 0.003	0.250 ± 0.003	0.234 ± 0.003
	CoM (self-generated)	$\mathbf{0.944 \pm 0.002}$	$\mathbf{0.942 \pm 0.002}$	$\mathbf{0.941 \pm 0.002}$
	Agentic CoM	0.532 ± 0.005	0.504 ± 0.005	0.491 ± 0.005
Structural success	Direct SMILES	0.458 ± 0.007	0.472 ± 0.007	0.362 ± 0.007
	Agentic edit program	0.482 ± 0.007	0.497 ± 0.007	0.427 ± 0.007
	CoM (self-generated)	0.351 ± 0.004	0.380 ± 0.004	0.283 ± 0.004
	Agentic CoM	$\mathbf{0.495 \pm 0.006}$	$\mathbf{0.517 \pm 0.006}$	$\mathbf{0.444 \pm 0.006}$
Property success	Direct SMILES	$\mathbf{0.329 \pm 0.006}$	$\mathbf{0.313 \pm 0.006}$	$\mathbf{0.128 \pm 0.005}$
	Agentic edit program	0.284 ± 0.006	0.278 ± 0.006	$\mathbf{0.126 \pm 0.005}$
	CoM (self-generated)	0.239 ± 0.003	0.233 ± 0.002	0.091 ± 0.002
	Agentic CoM	0.282 ± 0.005	0.288 ± 0.005	0.121 ± 0.004
Joint success	Direct SMILES	$\mathbf{0.170 \pm 0.005}$	0.161 ± 0.005	0.061 ± 0.003
	Agentic edit program	$\mathbf{0.165 \pm 0.005}$	0.151 ± 0.005	$\mathbf{0.068 \pm 0.003}$
	CoM (self-generated)	0.116 ± 0.002	0.114 ± 0.002	0.040 ± 0.001
	Agentic CoM	$\mathbf{0.166 \pm 0.004}$	$\mathbf{0.166 \pm 0.004}$	$\mathbf{0.065 \pm 0.003}$
DH-AUC	Direct SMILES	0.017 ± 0.000	0.016 ± 0.000	0.007 ± 0.000
	Agentic edit program	0.022 ± 0.001	0.021 ± 0.000	0.009 ± 0.000
	CoM (self-generated)	$\mathbf{0.064 \pm 0.001}$	$\mathbf{0.065 \pm 0.001}$	$\mathbf{0.023 \pm 0.001}$
	Agentic CoM	0.040 ± 0.001	0.038 ± 0.001	0.016 ± 0.000
Median trace tokens	Direct SMILES	47	46	58
	Agentic edit program	124	114	171
	CoM (self-generated)	201	179	208
	Agentic CoM	602	533	665

Table D5: 1M models after pretraining, SFT, and GRPO ($G = 16$). Values are means ± 1 SE over evaluation tasks per regime; trace tokens are medians. Reporting conventions are given in Appendix D.

Metric	Format	ID	Gap	Extrapolation
Validity	Direct SMILES	0.909 ± 0.003	0.932 ± 0.003	0.820 ± 0.005
	Agentic edit program	0.983 ± 0.001	0.984 ± 0.000	0.976 ± 0.001
	CoM (self-generated)	0.912 ± 0.002	0.927 ± 0.002	0.874 ± 0.003
	Agentic CoM	$\mathbf{0.998 \pm 0.000}$	$\mathbf{0.999 \pm 0.000}$	$\mathbf{0.990 \pm 0.001}$
Uniqueness	Direct SMILES	0.883 ± 0.003	0.880 ± 0.003	0.901 ± 0.003
	Agentic edit program	0.841 ± 0.003	0.859 ± 0.003	0.812 ± 0.004
	CoM (self-generated)	0.909 ± 0.002	0.908 ± 0.002	$\mathbf{0.923 \pm 0.002}$
	Agentic CoM	$\mathbf{0.915 \pm 0.003}$	$\mathbf{0.915 \pm 0.003}$	0.888 ± 0.003
Structural success	Direct SMILES	0.626 ± 0.006	0.677 ± 0.006	0.491 ± 0.007
	Agentic edit program	$\mathbf{0.903 \pm 0.004}$	$\mathbf{0.910 \pm 0.004}$	$\mathbf{0.848 \pm 0.005}$
	CoM (self-generated)	0.341 ± 0.006	0.391 ± 0.006	0.260 ± 0.006
	Agentic CoM	0.885 ± 0.005	$\mathbf{0.913 \pm 0.004}$	0.837 ± 0.006
Property success	Direct SMILES	$\mathbf{0.552 \pm 0.006}$	0.504 ± 0.006	0.118 ± 0.005
	Agentic edit program	0.358 ± 0.007	0.392 ± 0.007	$\mathbf{0.161 \pm 0.006}$
	CoM (self-generated)	0.523 ± 0.005	0.421 ± 0.006	0.110 ± 0.004
	Agentic CoM	0.505 ± 0.008	$\mathbf{0.532 \pm 0.007}$	0.146 ± 0.005
Joint success	Direct SMILES	0.397 ± 0.006	0.379 ± 0.006	0.075 ± 0.004
	Agentic edit program	0.337 ± 0.007	0.370 ± 0.007	$\mathbf{0.134 \pm 0.005}$
	CoM (self-generated)	0.201 ± 0.005	0.179 ± 0.005	0.041 ± 0.002
	Agentic CoM	$\mathbf{0.460 \pm 0.008}$	$\mathbf{0.495 \pm 0.007}$	0.121 ± 0.005
DH-AUC	Direct SMILES	0.136 ± 0.002	0.130 ± 0.002	0.025 ± 0.001
	Agentic edit program	0.081 ± 0.002	0.090 ± 0.002	0.032 ± 0.001
	CoM (self-generated)	0.087 ± 0.002	0.080 ± 0.002	0.018 ± 0.001
	Agentic CoM	$\mathbf{0.181 \pm 0.003}$	$\mathbf{0.190 \pm 0.003}$	$\mathbf{0.049 \pm 0.002}$
Median trace tokens	Direct SMILES	41	39	49
	Agentic edit program	314	302	364
	CoM (self-generated)	195	175	227
	Agentic CoM	1,202	1,126	1,405

Table D6: 20M models after pretraining, SFT, and GRPO ($G = 16$). Values are means ± 1 SE over evaluation tasks per regime; trace tokens are medians. Reporting conventions are given in Appendix D.

Metric	Format	ID	Gap	Extrapolation
Validity	Direct SMILES	0.947 ± 0.003	0.962 ± 0.003	0.860 ± 0.006
	Agentic edit program	0.993 ± 0.001	0.994 ± 0.001	$\mathbf{0.990 \pm 0.001}$
	CoM (self-generated)	0.948 ± 0.003	0.958 ± 0.002	0.868 ± 0.005
	Agentic CoM	$\mathbf{0.997 \pm 0.000}$	$\mathbf{0.998 \pm 0.000}$	0.987 ± 0.001
Uniqueness	Direct SMILES	0.836 ± 0.004	0.850 ± 0.003	0.843 ± 0.004
	Agentic edit program	0.789 ± 0.004	0.798 ± 0.004	0.752 ± 0.005
	CoM (self-generated)	$\mathbf{0.888 \pm 0.003}$	$\mathbf{0.886 \pm 0.003}$	$\mathbf{0.893 \pm 0.003}$
	Agentic CoM	0.881 ± 0.003	$\mathbf{0.887 \pm 0.003}$	0.855 ± 0.003
Structural success	Direct SMILES	0.761 ± 0.007	0.816 ± 0.006	0.605 ± 0.008
	Agentic edit program	$\mathbf{0.903 \pm 0.005}$	$\mathbf{0.930 \pm 0.004}$	$\mathbf{0.856 \pm 0.006}$
	CoM (self-generated)	0.722 ± 0.006	0.772 ± 0.006	0.581 ± 0.007
	Agentic CoM	0.874 ± 0.005	0.905 ± 0.004	0.826 ± 0.006
Property success	Direct SMILES	0.589 ± 0.007	0.567 ± 0.007	0.143 ± 0.005
	Agentic edit program	0.510 ± 0.008	0.516 ± 0.008	0.181 ± 0.007
	CoM (self-generated)	0.592 ± 0.006	0.511 ± 0.006	0.136 ± 0.004
	Agentic CoM	$\mathbf{0.619 \pm 0.007}$	$\mathbf{0.625 \pm 0.007}$	$\mathbf{0.187 \pm 0.006}$
Joint success	Direct SMILES	0.489 ± 0.007	0.495 ± 0.007	0.102 ± 0.005
	Agentic edit program	0.470 ± 0.008	0.491 ± 0.008	0.148 ± 0.006
	CoM (self-generated)	0.464 ± 0.006	0.426 ± 0.006	0.092 ± 0.004
	Agentic CoM	$\mathbf{0.553 \pm 0.008}$	$\mathbf{0.576 \pm 0.008}$	$\mathbf{0.155 \pm 0.006}$
DH-AUC	Direct SMILES	0.130 ± 0.002	0.135 ± 0.002	0.027 ± 0.001
	Agentic edit program	0.101 ± 0.002	0.105 ± 0.002	0.030 ± 0.001
	CoM (self-generated)	0.166 ± 0.002	0.152 ± 0.002	0.036 ± 0.001
	Agentic CoM	$\mathbf{0.208 \pm 0.003}$	$\mathbf{0.219 \pm 0.003}$	$\mathbf{0.055 \pm 0.002}$
Median trace tokens	Direct SMILES	54	53	62
	Agentic edit program	283	269	360
	CoM (self-generated)	290	290	300
	Agentic CoM	1,478	1,438	1,680

D.2 RL INITIALIZATION AND GROUP SIZE

RL initialization. SFT improves subsequent RL performance on in-distribution and interpolation gap targets for all four formats at both sizes (Table D7 and Figure D3). The difference is smaller for the 20M Agentic CoM: in-distribution joint success reaches 0.492 with RL directly from pretraining, compared with 0.553 after SFT; the corresponding 1M values are 0.203 and 0.460. SFT is not uniformly beneficial under extrapolation: the 20M Agentic CoM reaches 0.164 without SFT versus 0.155 with it. Tables D8 and D9 give the complete results for the direct route.

Table D7: In-distribution joint success for both model sizes and RL initializations. Values are means ± 1 task-level SE. All RL runs use 500 iterations and $G = 16$. Within each model size and training stage, bold marks the highest mean and any mean within one SE of it, using the highest mean's SE.

Size	Format	Pretraining	SFT	SFT $\rightarrow$ RL	Pre $\rightarrow$ RL
1M	Direct SMILES	0.063 ± 0.002	0.097 ± 0.002	0.397 ± 0.006	0.151 ± 0.004
	Agentic edit program	0.078 ± 0.002	0.090 ± 0.002	0.337 ± 0.007	0.164 ± 0.005
	CoM (self-generated)	0.054 ± 0.001	0.071 ± 0.002	0.201 ± 0.005	0.103 ± 0.003
	Agentic CoM	$\mathbf{0.086 \pm 0.002}$	$\mathbf{0.127 \pm 0.003}$	$\mathbf{0.460 \pm 0.008}$	$\mathbf{0.203 \pm 0.005}$
20M	Direct SMILES	0.101 ± 0.002	$\mathbf{0.170 \pm 0.005}$	0.489 ± 0.007	0.302 ± 0.006
	Agentic edit program	$\mathbf{0.110 \pm 0.002}$	$\mathbf{0.165 \pm 0.005}$	0.470 ± 0.008	0.352 ± 0.007
	CoM (self-generated)	0.090 ± 0.002	0.116 ± 0.002	0.464 ± 0.006	0.177 ± 0.005
	Agentic CoM	$\mathbf{0.111 \pm 0.003}$	$\mathbf{0.166 \pm 0.004}$	$\mathbf{0.553 \pm 0.008}$	$\mathbf{0.492 \pm 0.008}$

RL group size. Table D10 compares 1M models trained with RL group sizes $G = 16$ and $G = 64$ after SFT. Both use 4,096 trajectories per iteration for 500 iterations, with 256 and 64 prompts per iteration, respectively; other training settings and evaluation tasks are unchanged. All four formats attain higher mean joint success and DH-AUC with $G = 64$. Agentic CoM retains the highest joint success on in-distribution and gap targets, and the highest DH-AUC in all regimes. Its in-distribution joint success increases from 0.460 to 0.547, and gap success from 0.495 to 0.573; the corresponding DH-AUC scores increase from 0.181 to 0.258 and from 0.190 to 0.275. Agentic CoM has the largest absolute DH-AUC gain in each regime. The edit program retains a small joint-success lead under extrapolation at both group sizes.

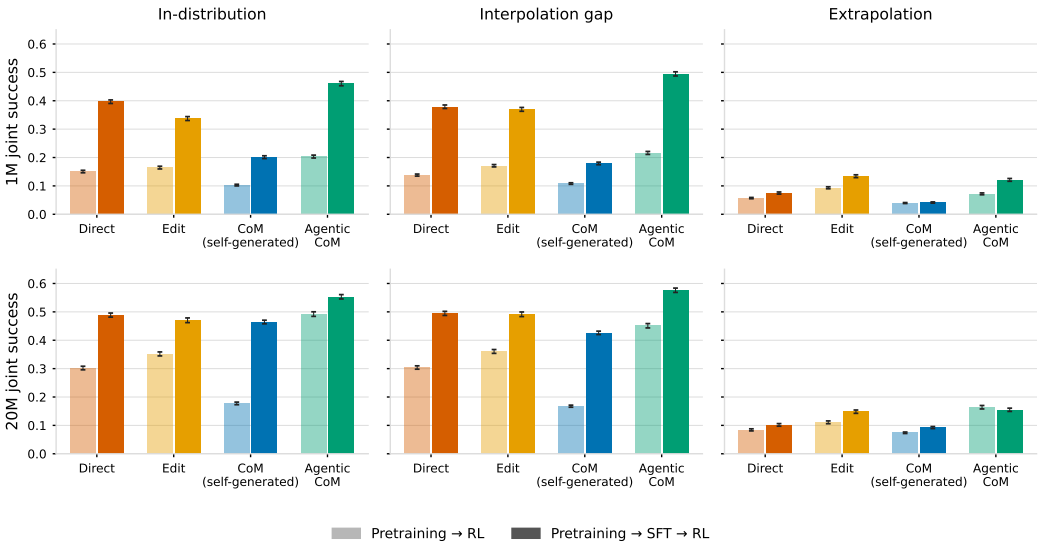

Figure D3: **RL initialization at both model sizes.** Light bars start RL directly from pretraining; dark bars insert SFT first. All runs use $G = 16$, 256 prompts per iteration and 500 iterations. Error bars are ± 1 SE over evaluation tasks per regime.

Table D8: 1M models after pretraining and GRPO ($G = 16$), without SFT. Values are means ± 1 SE over evaluation tasks per regime; trace tokens are medians. Reporting conventions are given in Appendix D.

Metric	Format	ID	Gap	Extrapolation
Validity	Direct SMILES	0.899 ± 0.003	0.915 ± 0.002	0.813 ± 0.005
	Agentic edit program	0.975 ± 0.001	0.976 ± 0.001	0.971 ± 0.001
	CoM (self-generated)	0.851 ± 0.003	0.873 ± 0.003	0.769 ± 0.005
	Agentic CoM	$\mathbf{0.999 \pm 0.000}$	$\mathbf{0.999 \pm 0.000}$	$\mathbf{0.994 \pm 0.001}$
Uniqueness	Direct SMILES	0.922 ± 0.002	0.912 ± 0.002	$\mathbf{0.926 \pm 0.002}$
	Agentic edit program	0.889 ± 0.003	0.884 ± 0.003	0.857 ± 0.003
	CoM (self-generated)	0.866 ± 0.003	0.854 ± 0.003	0.875 ± 0.003
	Agentic CoM	$\mathbf{0.937 \pm 0.003}$	$\mathbf{0.925 \pm 0.003}$	0.920 ± 0.003
Structural success	Direct SMILES	0.656 ± 0.006	0.681 ± 0.006	0.531 ± 0.007
	Agentic edit program	0.914 ± 0.003	0.920 ± 0.003	$\mathbf{0.863 \pm 0.005}$
	CoM (self-generated)	0.564 ± 0.006	0.596 ± 0.006	0.454 ± 0.006
	Agentic CoM	$\mathbf{0.931 \pm 0.004}$	$\mathbf{0.936 \pm 0.004}$	$\mathbf{0.859 \pm 0.006}$
Property success	Direct SMILES	0.196 ± 0.005	0.182 ± 0.004	0.086 ± 0.003
	Agentic edit program	0.173 ± 0.005	0.180 ± 0.004	$\mathbf{0.111 \pm 0.004}$
	CoM (self-generated)	0.147 ± 0.003	0.160 ± 0.003	0.065 ± 0.002
	Agentic CoM	$\mathbf{0.213 \pm 0.005}$	$\mathbf{0.229 \pm 0.005}$	0.093 ± 0.004
Joint success	Direct SMILES	0.151 ± 0.004	0.138 ± 0.004	0.057 ± 0.002
	Agentic edit program	0.164 ± 0.005	0.171 ± 0.004	$\mathbf{0.093 \pm 0.003}$
	CoM (self-generated)	0.103 ± 0.003	0.108 ± 0.003	0.040 ± 0.002
	Agentic CoM	$\mathbf{0.203 \pm 0.005}$	$\mathbf{0.216 \pm 0.005}$	0.072 ± 0.003
DH-AUC	Direct SMILES	0.061 ± 0.002	0.058 ± 0.001	0.025 ± 0.001
	Agentic edit program	0.049 ± 0.001	0.051 ± 0.001	$\mathbf{0.031 \pm 0.001}$
	CoM (self-generated)	0.043 ± 0.001	0.046 ± 0.001	0.017 ± 0.001
	Agentic CoM	$\mathbf{0.088 \pm 0.002}$	$\mathbf{0.085 \pm 0.002}$	$\mathbf{0.032 \pm 0.001}$
Median trace tokens	Direct SMILES	46	43	54
	Agentic edit program	261	243	330
	CoM (self-generated)	134	127	142
	Agentic CoM	1,076	962	1,202

Table D9: 20M models after pretraining and GRPO ($G = 16$), without SFT. Values are means ± 1 SE over evaluation tasks per regime; trace tokens are medians. Reporting conventions are given in Appendix D.

Metric	Format	ID	Gap	Extrapolation
Validity	Direct SMILES	0.943 ± 0.002	0.955 ± 0.002	0.863 ± 0.005
	Agentic edit program	$\mathbf{0.998 \pm 0.000}$	$\mathbf{0.997 \pm 0.000}$	$\mathbf{0.997 \pm 0.000}$
	CoM (self-generated)	0.936 ± 0.003	0.945 ± 0.002	0.854 ± 0.005
	Agentic CoM	0.989 ± 0.001	0.988 ± 0.001	0.979 ± 0.002
Uniqueness	Direct SMILES	0.880 ± 0.003	0.880 ± 0.003	0.878 ± 0.003
	Agentic edit program	0.848 ± 0.003	0.855 ± 0.003	0.839 ± 0.003
	CoM (self-generated)	0.891 ± 0.003	0.890 ± 0.003	$\mathbf{0.881 \pm 0.003}$
	Agentic CoM	$\mathbf{0.909 \pm 0.003}$	$\mathbf{0.910 \pm 0.003}$	0.874 ± 0.004
Structural success	Direct SMILES	0.820 ± 0.005	0.853 ± 0.004	0.705 ± 0.007
	Agentic edit program	$\mathbf{0.945 \pm 0.003}$	$\mathbf{0.943 \pm 0.003}$	$\mathbf{0.921 \pm 0.003}$
	CoM (self-generated)	0.800 ± 0.005	0.813 ± 0.005	0.668 ± 0.006
	Agentic CoM	0.924 ± 0.004	0.936 ± 0.003	0.871 ± 0.005
Property success	Direct SMILES	0.339 ± 0.006	0.332 ± 0.006	0.107 ± 0.004
	Agentic edit program	0.362 ± 0.007	0.375 ± 0.007	0.122 ± 0.005
	CoM (self-generated)	0.200 ± 0.005	0.190 ± 0.004	0.098 ± 0.003
	Agentic CoM	$\mathbf{0.513 \pm 0.008}$	$\mathbf{0.467 \pm 0.008}$	$\mathbf{0.181 \pm 0.006}$
Joint success	Direct SMILES	0.302 ± 0.006	0.304 ± 0.006	0.084 ± 0.004
	Agentic edit program	0.352 ± 0.007	0.361 ± 0.007	0.111 ± 0.005
	CoM (self-generated)	0.177 ± 0.005	0.168 ± 0.004	0.074 ± 0.003
	Agentic CoM	$\mathbf{0.492 \pm 0.008}$	$\mathbf{0.451 \pm 0.008}$	$\mathbf{0.164 \pm 0.006}$
DH-AUC	Direct SMILES	0.090 ± 0.002	0.090 ± 0.001	0.027 ± 0.001
	Agentic edit program	0.087 ± 0.002	0.090 ± 0.001	0.030 ± 0.001
	CoM (self-generated)	0.075 ± 0.002	0.068 ± 0.001	0.032 ± 0.001
	Agentic CoM	$\mathbf{0.199 \pm 0.003}$	$\mathbf{0.187 \pm 0.003}$	$\mathbf{0.063 \pm 0.002}$
Median trace tokens	Direct SMILES	54	53	64
	Agentic edit program	216	205	266
	CoM (self-generated)	190	186	205
	Agentic CoM	1,237	1,219	1,341

Table D10: RL group-size comparison for 1M models after SFT at a fixed trajectory budget. $G = 16$ uses $P = 256$ prompts per iteration; $G = 64$ uses $P = 64$. Both use 4,096 trajectories per iteration for 500 iterations. Values are means ± 1 SE over evaluation tasks per regime. Within each metric, regime, and group size, bold marks the highest mean and any mean within one SE of it, using the highest mean's SE.

Format	G	ID	Gap	Extrapolation
Joint success				
Direct SMILES	16	0.397 ± 0.006	0.379 ± 0.006	0.075 ± 0.004
	64	0.410 ± 0.006	0.382 ± 0.006	0.076 ± 0.004
Agentic edit program	16	0.337 ± 0.007	0.370 ± 0.007	$\mathbf{0.134 \pm 0.005}$
	64	0.395 ± 0.007	0.437 ± 0.007	$\mathbf{0.136 \pm 0.005}$
CoM (self-generated)	16	0.201 ± 0.005	0.179 ± 0.005	0.041 ± 0.002
	64	0.296 ± 0.006	0.262 ± 0.006	0.056 ± 0.003
Agentic CoM	16	$\mathbf{0.460 \pm 0.008}$	$\mathbf{0.495 \pm 0.007}$	0.121 ± 0.005
	64	$\mathbf{0.547 \pm 0.008}$	$\mathbf{0.573 \pm 0.007}$	$\mathbf{0.133 \pm 0.005}$
DH-AUC				
Direct SMILES	16	0.136 ± 0.002	0.130 ± 0.002	0.025 ± 0.001
	64	0.170 ± 0.003	0.163 ± 0.002	0.031 ± 0.001
Agentic edit program	16	0.081 ± 0.002	0.090 ± 0.002	0.032 ± 0.001
	64	0.122 ± 0.002	0.137 ± 0.002	0.041 ± 0.002
CoM (self-generated)	16	0.087 ± 0.002	0.080 ± 0.002	0.018 ± 0.001
	64	0.126 ± 0.003	0.111 ± 0.003	0.024 ± 0.001
Agentic CoM	16	$\mathbf{0.181 \pm 0.003}$	$\mathbf{0.190 \pm 0.003}$	$\mathbf{0.049 \pm 0.002}$
	64	$\mathbf{0.258 \pm 0.004}$	$\mathbf{0.275 \pm 0.004}$	$\mathbf{0.059 \pm 0.002}$

D.3 TRAJECTORY BEHAVIOUR AND CONSISTENCY

RL improves trajectory endpoints without necessarily improving action-label consistency (Table D11). For 1M self-generated CoM, the fraction of trajectories improving property error rises from 51% after SFT to 76% after RL with $G = 16$, while backtrack inconsistency rises from 26% to 65%. Bioactivity exhibits the same separation: for 20M self-generated CoM across both reference-activity bands, activity-improving trajectories rise from 33% before RL to 80% after RL, while delete-label inconsistency rises from 7% to 88%. Agentic CoM reaches 88% activity-improving trajectories in this setting while enforcing state-transition consistency through execution. Its zero inconsistency entries reflect this protocol, not estimated error rates. Figure D4 compares action-marker use across training stages on both benchmarks. Counts pool all three property evaluation regimes and both bioactivity reference bands. All RL settings in the figure use $G = 16$; "SFT + RL" starts from SFT, whereas "RL" starts from pretraining.

The trajectory-reliability analysis covers both model sizes, using in-distribution tasks for the five molecular properties and pooling moderate- and low-activity references for DRD2, GSK3β and JNK3 within each bioactivity row. Property models are compared after SFT and SFT + RL ($G = 16$, plus $G = 64$ at 1M); bioactivity models are compared before and after RL directly from pretraining ($G = 16$).

Progress statistics pool eligible trajectories across both reference bands; action inconsistency rates pool the corresponding action counts.

Progress compares the reference with the last parseable intermediate state, including trajectories that do not commit a molecule. Property error is absolute target error divided by the task tolerance; bioactivity error is $1 - f_j(m)$ for the evaluated state m, since the requested target is $v^\star = 1.0$. Structural error is distance from the requested GED interval. An improving trajectory has strictly lower endpoint error than reference error. These measures describe endpoint progress, not monotonic improvement along every step or committed-output success.

For self-generated CoM, add/delete inconsistency counts parseable steps whose atom count fails to increase/decrease; backtrack inconsistency counts parseable states that do not match any earlier state, including the reference. Each rate uses its own action-step denominator, not a count of inconsistent trajectories. Unparseable states are skipped, and backtrack matching uses canonical isomeric SMILES for both the reference and intermediate states. These checks probe action-label consistency rather than complete trajectory semantics.

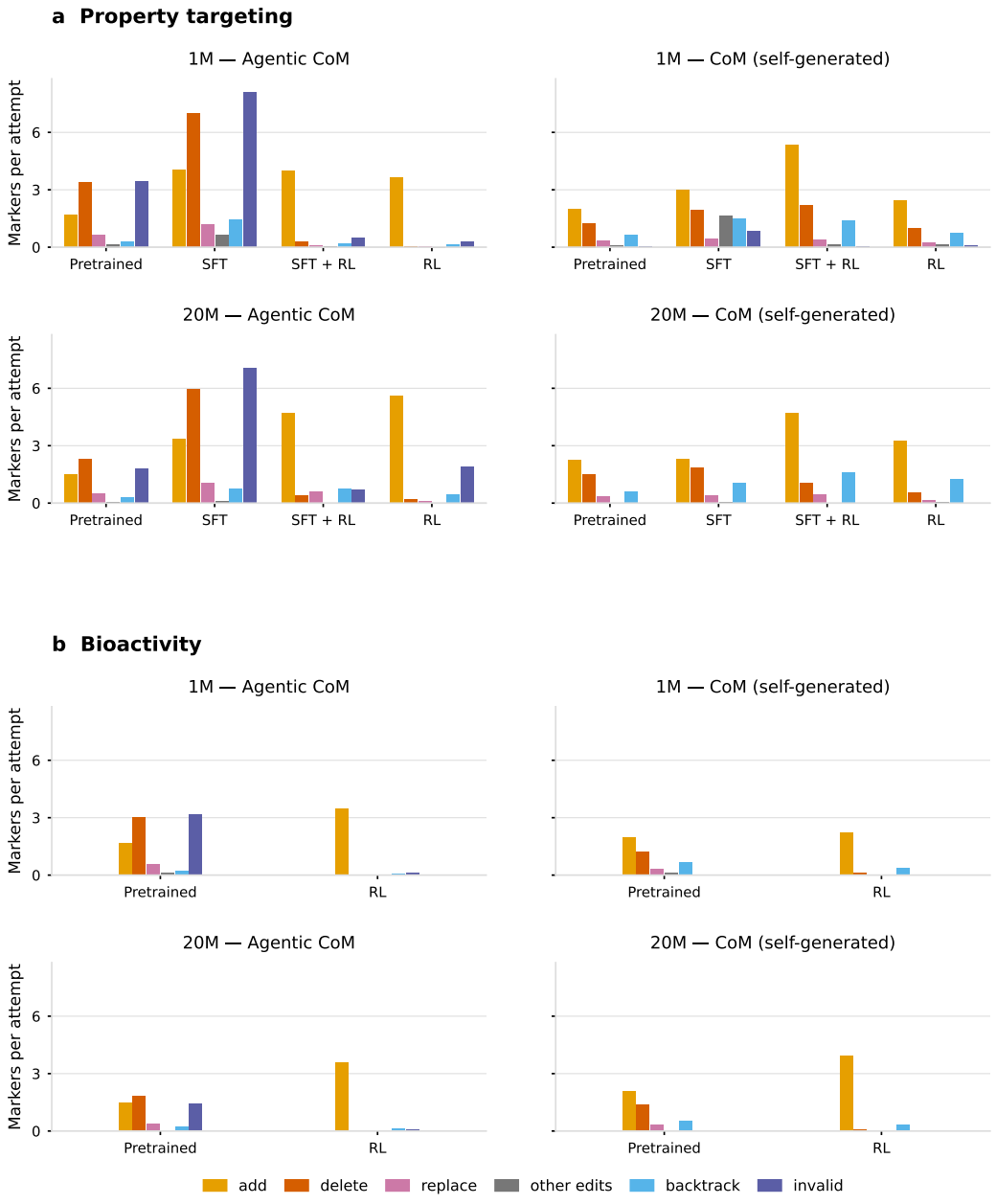

Figure D4: **Action markers across both benchmarks.** Mean post-prompt marker counts per attempt, including failures. Each setting contains 960,000 property-targeting attempts or 576,000 bioactivity attempts. "Other edits" groups bridge, bond and crossover. Invalid markers are environment-generated for Agentic CoM and model-written for self-generated CoM.

Table D11: **Trajectory progress and action-label consistency.** Improved is the fraction of trajectories that reduce error from the reference to the last parseable state; Error shows reference → endpoint. Trajectories without a final commit are included. Objective error is tolerance-normalized for property targeting and $1 - f_j(m)$ for bioactivity; structural error is distance from the requested GED interval. Inconsistency measures add/delete and backtrack mismatches as defined in Appendix D.3.

		Objective		Structural		Inconsistency	
Variant	Stage	Improved	Error	Improved	Error	add / del	backtrack
Property targeting: 1M							
CoM							
(self-generated)	SFT	51%	3.1 → 3.4	66%	14.8 → 12.2	20%/14%	26%
	SFT + RL ($G = 16$)	76%	3.1 → 1.4	69%	14.8 → 10.6	29%/12%	65%
	SFT + RL ($G = 64$)	83%	3.1 → 0.9	72%	14.8 → 9.2	30%/14%	86%
Agentic CoM	SFT	53%	3.1 → 3.0	86%	14.7 → 5.8	0	0
	SFT + RL ($G = 16$)	66%	3.1 → 1.7	96%	14.7 → 0.4	0	0
	SFT + RL ($G = 64$)	74%	3.1 → 1.4	96%	14.7 → 0.4	0	0
Property targeting: 20M							
CoM							
(self-generated)	SFT	57%	3.1 → 2.7	77%	14.8 → 7.8	7%/10%	10%
	SFT + RL ($G = 16$)	78%	3.1 → 1.1	91%	14.7 → 2.5	10%/15%	20%
Agentic CoM	SFT	57%	3.1 → 2.8	87%	14.5 → 4.7	0	0
	SFT + RL ($G = 16$)	74%	3.1 → 1.3	96%	14.7 → 0.6	0	0
Bioactivity: 1M							
CoM							
(self-generated)	Pretrained	28%	0.77 → 0.87	64%	11.9 → 12.5	11%/12%	18%
	RL ($G = 16$)	65%	0.76 → 0.73	97%	11.9 → 1.0	19%/48%	26%
Agentic CoM	Pretrained	37%	0.76 → 0.83	93%	11.8 → 3.2	0	0
	RL ($G = 16$)	83%	0.76 → 0.66	100%	11.9 → 0.0	0	0
Bioactivity: 20M							
CoM							
(self-generated)	Pretrained	33%	0.76 → 0.85	81%	11.8 → 6.8	4%/7%	5%
	RL ($G = 16$)	80%	0.76 → 0.67	100%	11.9 → 0.2	13%/88%	3%
Agentic CoM	Pretrained	36%	0.76 → 0.83	95%	11.8 → 2.0	0	0
	RL ($G = 16$)	88%	0.76 → 0.60	100%	11.9 → 0.1	0	0

D.4 DOWNSTREAM BIOACTIVITY ADAPTATION

RL substantially improves joint success over pretraining at both model sizes. At $\kappa = 0.5$, Agentic CoM has the highest mean joint success and DH-AUC with moderate-activity references: joint success increases from 0.056 to 0.479 at 1M and from 0.066 to 0.650 at 20M. The aggregate advantage is not uniform across reference-activity bands or metrics. With low-activity references, Direct SMILES has slightly higher DH-AUC at 1M, while the 20M Agentic CoM gains are concentrated in GSK3β. The stricter threshold reduces success substantially; joint successes at $\kappa = 0.8$ occur predominantly on DRD2, with none on JNK3 in these evaluations.

Tables D12–D15 report pretrained models and models adapted directly through RL, separately for each size, using the configuration in Appendix C.3. Results aggregate DRD2, GSK3β, and JNK3, with 750 tasks per target and 128 outputs per task. Moderate and Low denote reference-activity bands $[0.3, 0.5)$ and $[0.0, 0.3)$, respectively (Appendix A.3.2); they describe the starting molecules.

Activity hit instead tests the generated molecule against an output threshold $\kappa \in \{0.5, 0.8\}$. Joint success additionally requires the requested GED interval, and DH-AUC uses the joint successes at that threshold. Both thresholds are applied to the same saved outputs, without retraining. These activity scores are predictions from fixed oracles, not experimental measurements. The tables follow the shared reporting conventions above, with 2,250 tasks per reference-activity band.

Table D12: **Adaptation to downstream bioactivity objectives. 1M models.** Results across four generation formats after pretraining, before bioactivity adaptation, macro-averaged across DRD2, GSK3β, and JNK3. Moderate and Low denote reference activities in $[0.3, 0.5)$ and $[0.0, 0.3)$, respectively. Activity hit uses the indicated threshold $\kappa \in \{0.5, 0.8\}$; joint success additionally requires the requested GED interval. DH-AUC captures the number and diversity of jointly successful molecules. Values are means ± 1 SE over evaluation tasks per reference band; trace tokens are medians. Bold marks means within one SE of the best result for each metric and band, using its SE.

Metric	Format	Moderate	Low
Validity	Direct SMILES	0.822 ± 0.003	0.829 ± 0.003
	Agentic edit program	0.754 ± 0.002	0.760 ± 0.002
	CoM (self-generated)	0.669 ± 0.004	0.682 ± 0.004
	Agentic CoM	$\mathbf{0.977 \pm 0.000}$	$\mathbf{0.978 \pm 0.000}$
Uniqueness	Direct SMILES	0.880 ± 0.002	0.885 ± 0.002
	Agentic edit program	0.878 ± 0.002	0.881 ± 0.002
	CoM (self-generated)	$\mathbf{0.940 \pm 0.001}$	$\mathbf{0.942 \pm 0.001}$
	Agentic CoM	0.833 ± 0.002	0.834 ± 0.002
Structural success	Direct SMILES	0.215 ± 0.003	0.224 ± 0.003
	Agentic edit program	0.486 ± 0.003	0.483 ± 0.003
	CoM (self-generated)	0.257 ± 0.003	0.264 ± 0.003
	Agentic CoM	$\mathbf{0.591 \pm 0.004}$	$\mathbf{0.588 \pm 0.003}$
Activity hit ($\kappa = 0.5$)	Direct SMILES	0.048 ± 0.001	$\mathbf{0.001 \pm 0.000}$
	Agentic edit program	0.063 ± 0.002	0.001 ± 0.000
	CoM (self-generated)	0.044 ± 0.001	0.001 ± 0.000
	Agentic CoM	$\mathbf{0.088 \pm 0.002}$	0.001 ± 0.000
Joint success ($\kappa = 0.5$)	Direct SMILES	0.017 ± 0.001	0.000 ± 0.000
	Agentic edit program	0.045 ± 0.001	$\mathbf{0.001 \pm 0.000}$
	CoM (self-generated)	0.023 ± 0.001	0.000 ± 0.000
	Agentic CoM	$\mathbf{0.056 \pm 0.002}$	$\mathbf{0.001 \pm 0.000}$
DH-AUC ($\kappa = 0.5$)	Direct SMILES	0.010 ± 0.000	0.000 ± 0.000
	Agentic edit program	0.020 ± 0.000	0.000 ± 0.000
	CoM (self-generated)	0.013 ± 0.000	0.000 ± 0.000
	Agentic CoM	$\mathbf{0.025 \pm 0.001}$	$\mathbf{0.000 \pm 0.000}$
Activity hit ($\kappa = 0.8$)	Direct SMILES	0.006 ± 0.000	$\mathbf{0.000 \pm 0.000}$
	Agentic edit program	0.008 ± 0.000	$\mathbf{0.000 \pm 0.000}$
	CoM (self-generated)	0.007 ± 0.000	$\mathbf{0.000 \pm 0.000}$
	Agentic CoM	$\mathbf{0.010 \pm 0.001}$	$\mathbf{0.000 \pm 0.000}$
Joint success ($\kappa = 0.8$)	Direct SMILES	0.003 ± 0.000	$\mathbf{0.000 \pm 0.000}$
	Agentic edit program	0.006 ± 0.000	$\mathbf{0.000 \pm 0.000}$
	CoM (self-generated)	0.004 ± 0.000	$\mathbf{0.000 \pm 0.000}$
	Agentic CoM	$\mathbf{0.007 \pm 0.000}$	$\mathbf{0.000 \pm 0.000}$
DH-AUC ($\kappa = 0.8$)	Direct SMILES	0.002 ± 0.000	0.000 ± 0.000
	Agentic edit program	0.003 ± 0.000	$\mathbf{0.000 \pm 0.000}$
	CoM (self-generated)	0.002 ± 0.000	0.000 ± 0.000
	Agentic CoM	$\mathbf{0.004 \pm 0.000}$	$\mathbf{0.000 \pm 0.000}$
Median trace tokens	Direct SMILES	48	45
	Agentic edit program	227	211
	CoM (self-generated)	126	121
	Agentic CoM	399	389

Table D13: **Adaptation to downstream bioactivity objectives. 1M models.** Results across four generation formats after GRPO directly from pretraining ($G = 16$), macro-averaged across DRD2, GSK3β, and JNK3. Moderate and Low denote reference activities in $[0.3, 0.5)$ and $[0.0, 0.3)$, respectively. Activity hit uses the indicated threshold $\kappa \in \{0.5, 0.8\}$; joint success additionally requires the requested GED interval. DH-AUC captures the number and diversity of jointly successful molecules. Values are means ± 1 SE over evaluation tasks per reference band; trace tokens are medians. Bold marks means within one SE of the best result for each metric and band, using its SE.

Metric	Format	Moderate	Low
Validity	Direct SMILES	0.928 ± 0.002	0.923 ± 0.003
	Agentic edit program	0.991 ± 0.000	0.989 ± 0.001
	CoM (self-generated)	0.928 ± 0.003	0.910 ± 0.003
	Agentic CoM	$\mathbf{1.000 \pm 0.000}$	$\mathbf{1.000 \pm 0.000}$
Uniqueness	Direct SMILES	0.870 ± 0.003	0.880 ± 0.003
	Agentic edit program	0.826 ± 0.002	0.831 ± 0.002
	CoM (self-generated)	0.839 ± 0.003	0.851 ± 0.003
	Agentic CoM	$\mathbf{0.876 \pm 0.002}$	$\mathbf{0.887 \pm 0.002}$
Structural success	Direct SMILES	0.843 ± 0.004	0.806 ± 0.005
	Agentic edit program	0.984 ± 0.001	0.975 ± 0.002
	CoM (self-generated)	0.859 ± 0.004	0.821 ± 0.005
	Agentic CoM	$\mathbf{0.995 \pm 0.000}$	$\mathbf{0.986 \pm 0.002}$
Activity hit ($\kappa = 0.5$)	Direct SMILES	0.341 ± 0.008	$\mathbf{0.029 \pm 0.003}$
	Agentic edit program	0.407 ± 0.009	0.021 ± 0.002
	CoM (self-generated)	0.228 ± 0.006	0.003 ± 0.001
	Agentic CoM	$\mathbf{0.481 \pm 0.009}$	0.025 ± 0.003
Joint success ($\kappa = 0.5$)	Direct SMILES	0.305 ± 0.007	$\mathbf{0.023 \pm 0.002}$
	Agentic edit program	0.405 ± 0.009	0.021 ± 0.002
	CoM (self-generated)	0.211 ± 0.006	0.002 ± 0.001
	Agentic CoM	$\mathbf{0.479 \pm 0.009}$	$\mathbf{0.025 \pm 0.003}$
DH-AUC ($\kappa = 0.5$)	Direct SMILES	0.075 ± 0.002	$\mathbf{0.007 \pm 0.001}$
	Agentic edit program	0.076 ± 0.002	0.004 ± 0.000
	CoM (self-generated)	0.055 ± 0.002	0.001 ± 0.000
	Agentic CoM	$\mathbf{0.122 \pm 0.002}$	0.006 ± 0.001
Activity hit ($\kappa = 0.8$)	Direct SMILES	0.146 ± 0.006	$\mathbf{0.009 \pm 0.001}$
	Agentic edit program	0.137 ± 0.006	0.004 ± 0.001
	CoM (self-generated)	0.065 ± 0.004	0.000 ± 0.000
	Agentic CoM	$\mathbf{0.171 \pm 0.007}$	0.006 ± 0.001
Joint success ($\kappa = 0.8$)	Direct SMILES	0.128 ± 0.006	$\mathbf{0.007 \pm 0.001}$
	Agentic edit program	0.135 ± 0.006	0.004 ± 0.001
	CoM (self-generated)	0.060 ± 0.003	0.000 ± 0.000
	Agentic CoM	$\mathbf{0.170 \pm 0.007}$	$\mathbf{0.006 \pm 0.001}$
DH-AUC ($\kappa = 0.8$)	Direct SMILES	0.035 ± 0.002	$\mathbf{0.002 \pm 0.000}$
	Agentic edit program	0.026 ± 0.001	0.001 ± 0.000
	CoM (self-generated)	0.019 ± 0.001	0.000 ± 0.000
	Agentic CoM	$\mathbf{0.044 \pm 0.002}$	0.001 ± 0.000
Median trace tokens	Direct SMILES	61	57
	Agentic edit program	284	289
	CoM (self-generated)	114	102
	Agentic CoM	1,191	1,172

Table D14: **Adaptation to downstream bioactivity objectives. 20M models.** Results across four generation formats after pretraining, before bioactivity adaptation, macro-averaged across DRD2, GSK3β, and JNK3. Moderate and Low denote reference activities in $[0.3, 0.5)$ and $[0.0, 0.3)$, respectively. Activity hit uses the indicated threshold $\kappa \in \{0.5, 0.8\}$; joint success additionally requires the requested GED interval. DH-AUC captures the number and diversity of jointly successful molecules. Values are means ± 1 SE over evaluation tasks per reference band; trace tokens are medians. Bold marks means within one SE of the best result for each metric and band, using its SE.

Metric	Format	Moderate	Low
Validity	Direct SMILES	0.926 ± 0.002	0.930 ± 0.002
	Agentic edit program	$\mathbf{1.000 \pm 0.000}$	$\mathbf{1.000 \pm 0.000}$
	CoM (self-generated)	0.886 ± 0.002	0.886 ± 0.002
	Agentic CoM	0.986 ± 0.000	0.986 ± 0.000
Uniqueness	Direct SMILES	0.916 ± 0.002	0.921 ± 0.002
	Agentic edit program	0.787 ± 0.002	0.786 ± 0.003
	CoM (self-generated)	$\mathbf{0.925 \pm 0.001}$	$\mathbf{0.924 \pm 0.002}$
	Agentic CoM	0.833 ± 0.002	0.834 ± 0.002
Structural success	Direct SMILES	0.607 ± 0.003	0.601 ± 0.003
	Agentic edit program	0.683 ± 0.003	0.677 ± 0.003
	CoM (self-generated)	0.504 ± 0.003	0.504 ± 0.003
	Agentic CoM	$\mathbf{0.723 \pm 0.003}$	$\mathbf{0.718 \pm 0.003}$
Activity hit ($\kappa = 0.5$)	Direct SMILES	0.077 ± 0.002	$\mathbf{0.001 \pm 0.000}$
	Agentic edit program	0.081 ± 0.002	$\mathbf{0.001 \pm 0.000}$
	CoM (self-generated)	0.066 ± 0.002	0.001 ± 0.000
	Agentic CoM	$\mathbf{0.088 \pm 0.002}$	$\mathbf{0.001 \pm 0.000}$
Joint success ($\kappa = 0.5$)	Direct SMILES	0.052 ± 0.001	$\mathbf{0.001 \pm 0.000}$
	Agentic edit program	0.061 ± 0.002	$\mathbf{0.001 \pm 0.000}$
	CoM (self-generated)	0.044 ± 0.001	0.001 ± 0.000
	Agentic CoM	$\mathbf{0.066 \pm 0.002}$	$\mathbf{0.001 \pm 0.000}$
DH-AUC ($\kappa = 0.5$)	Direct SMILES	0.026 ± 0.001	$\mathbf{0.001 \pm 0.000}$
	Agentic edit program	0.026 ± 0.001	0.000 ± 0.000
	CoM (self-generated)	0.022 ± 0.000	0.000 ± 0.000
	Agentic CoM	$\mathbf{0.028 \pm 0.001}$	$\mathbf{0.000 \pm 0.000}$
Activity hit ($\kappa = 0.8$)	Direct SMILES	0.009 ± 0.001	$\mathbf{0.000 \pm 0.000}$
	Agentic edit program	0.010 ± 0.001	$\mathbf{0.000 \pm 0.000}$
	CoM (self-generated)	0.008 ± 0.000	$\mathbf{0.000 \pm 0.000}$
	Agentic CoM	$\mathbf{0.011 \pm 0.001}$	$\mathbf{0.000 \pm 0.000}$
Joint success ($\kappa = 0.8$)	Direct SMILES	0.006 ± 0.000	$\mathbf{0.000 \pm 0.000}$
	Agentic edit program	0.007 ± 0.000	$\mathbf{0.000 \pm 0.000}$
	CoM (self-generated)	0.006 ± 0.000	$\mathbf{0.000 \pm 0.000}$
	Agentic CoM	$\mathbf{0.008 \pm 0.001}$	$\mathbf{0.000 \pm 0.000}$
DH-AUC ($\kappa = 0.8$)	Direct SMILES	0.004 ± 0.000	$\mathbf{0.000 \pm 0.000}$
	Agentic edit program	$\mathbf{0.004 \pm 0.000}$	$\mathbf{0.000 \pm 0.000}$
	CoM (self-generated)	0.003 ± 0.000	0.000 ± 0.000
	Agentic CoM	$\mathbf{0.004 \pm 0.000}$	0.000 ± 0.000
Median trace tokens	Direct SMILES	56	52
	Agentic edit program	125	117
	CoM (self-generated)	132	127
	Agentic CoM	389	381

Table D15: **Adaptation to downstream bioactivity objectives. 20M models.** Results across four generation formats after GRPO directly from pretraining ($G = 16$), macro-averaged across DRD2, GSK3β, and JNK3. Moderate and Low denote reference activities in $[0.3, 0.5)$ and $[0.0, 0.3)$, respectively. Activity hit uses the indicated threshold $\kappa \in \{0.5, 0.8\}$; joint success additionally requires the requested GED interval. DH-AUC captures the number and diversity of jointly successful molecules. Values are means ± 1 SE over evaluation tasks per reference band; trace tokens are medians. Bold marks means within one SE of the best result for each metric and band, using its SE.

Metric	Format	Moderate	Low
Validity	Direct SMILES	0.970 ± 0.002	0.962 ± 0.002
	Agentic edit program	$\mathbf{1.000 \pm 0.000}$	$\mathbf{1.000 \pm 0.000}$
	CoM (self-generated)	0.986 ± 0.001	0.976 ± 0.002
	Agentic CoM	1.000 ± 0.000	1.000 ± 0.000
Uniqueness	Direct SMILES	0.845 ± 0.003	0.860 ± 0.003
	Agentic edit program	0.838 ± 0.002	0.843 ± 0.002
	CoM (self-generated)	$\mathbf{0.864 \pm 0.002}$	$\mathbf{0.873 \pm 0.003}$
	Agentic CoM	0.828 ± 0.003	0.854 ± 0.002
Structural success	Direct SMILES	0.934 ± 0.003	0.881 ± 0.005
	Agentic edit program	0.991 ± 0.000	0.985 ± 0.001
	CoM (self-generated)	0.978 ± 0.002	0.954 ± 0.003
	Agentic CoM	$\mathbf{0.998 \pm 0.000}$	$\mathbf{0.995 \pm 0.001}$
Activity hit ($\kappa = 0.5$)	Direct SMILES	0.417 ± 0.009	0.049 ± 0.004
	Agentic edit program	0.474 ± 0.009	0.059 ± 0.004
	CoM (self-generated)	0.434 ± 0.009	0.023 ± 0.003
	Agentic CoM	$\mathbf{0.650 \pm 0.009}$	$\mathbf{0.125 \pm 0.006}$
Joint success ($\kappa = 0.5$)	Direct SMILES	0.395 ± 0.008	0.040 ± 0.003
	Agentic edit program	0.472 ± 0.009	0.058 ± 0.004
	CoM (self-generated)	0.430 ± 0.009	0.022 ± 0.003
	Agentic CoM	$\mathbf{0.650 \pm 0.009}$	$\mathbf{0.125 \pm 0.006}$
DH-AUC ($\kappa = 0.5$)	Direct SMILES	0.088 ± 0.002	0.010 ± 0.001
	Agentic edit program	0.089 ± 0.002	0.010 ± 0.001
	CoM (self-generated)	0.101 ± 0.002	0.005 ± 0.001
	Agentic CoM	$\mathbf{0.129 \pm 0.002}$	$\mathbf{0.031 \pm 0.001}$
Activity hit ($\kappa = 0.8$)	Direct SMILES	$\mathbf{0.199 \pm 0.007}$	$\mathbf{0.020 \pm 0.002}$
	Agentic edit program	0.186 ± 0.008	0.017 ± 0.002
	CoM (self-generated)	0.166 ± 0.007	0.007 ± 0.001
	Agentic CoM	$\mathbf{0.192 \pm 0.008}$	0.003 ± 0.001
Joint success ($\kappa = 0.8$)	Direct SMILES	$\mathbf{0.184 \pm 0.007}$	$\mathbf{0.016 \pm 0.002}$
	Agentic edit program	$\mathbf{0.185 \pm 0.008}$	$\mathbf{0.017 \pm 0.002}$
	CoM (self-generated)	0.164 ± 0.007	0.006 ± 0.001
	Agentic CoM	$\mathbf{0.192 \pm 0.008}$	0.003 ± 0.001
DH-AUC ($\kappa = 0.8$)	Direct SMILES	$\mathbf{0.042 \pm 0.002}$	$\mathbf{0.004 \pm 0.000}$
	Agentic edit program	0.033 ± 0.001	0.003 ± 0.000
	CoM (self-generated)	0.037 ± 0.002	0.001 ± 0.000
	Agentic CoM	0.039 ± 0.002	0.001 ± 0.000
Median trace tokens	Direct SMILES	62	56
	Agentic edit program	307	323
	CoM (self-generated)	249	243
	Agentic CoM	1,240	1,250

D.5 DIRECTIONAL PROPERTY OPTIMIZATION

TOMG-Bench (Li et al., 2026a) asks models to increase or decrease a molecular property while preserving similarity to the input. We evaluate all 5,000 QED and 5,000 logP requests distributed with RePO (Li et al., 2026b), retaining the original molecules and requested directions. We use our trained checkpoints for Agentic CoM and self-generated CoM at both model sizes. Following RePO's evaluation protocol, we generate one rollout per request.

Prompt adaptation and inference. We translate the directional instructions into our numeric prompt format, using decrease/increase targets of $0.15/0.90$ for QED and $-3/8$ for unpenalized RDKit logP. For 160 increase-QED requests, the reference value already reaches or exceeds 0.90, the upper bound of the QED training range $\mathcal{R}_j$. We therefore set the target to the nearest two-decimal value strictly above the reference, retaining the full test set. To encourage the small structural changes favoured by this benchmark, we condition on GED $[0, 2]$. This is our prompt adaptation, not a constraint specified by RePO or an equivalent of a Tanimoto-similarity interval. Sampling and interaction limits remain those in Appendix C.3.

Evaluation. Following RePO's released evaluator, SR is the fraction of all requests whose valid final molecule moves strictly in the requested direction. Missing, invalid, and unchanged outputs fail; reaching the numeric prompt target is not required. Sim averages input–output Tanimoto similarity over valid outputs, using radius-2, 2,048-bit Morgan fingerprints without chirality. SR$\times$Sim is the product of these two aggregate metrics. Neither GED compliance nor a hard Tanimoto threshold gates success. These directional metrics pool increase and decrease requests and differ from our target-matching and joint-success metrics.

Baselines. We take all baseline results directly from Li et al. (2026b), without rerunning the methods. These include Qwen2.5 (Qwen et al., 2025) and Llama 3.1 (Dubey et al., 2024) with SFT, GRPO (Shao et al., 2024), and RePO, plus a Qwen Distill-SFT baseline using the s1.1K corpus (Muennighoff et al., 2025). Additional baselines are MolT5 (Edwards et al., 2022), BioT5 (Pei et al., 2023), Graph GA (Jensen, 2019), REINVENT (Olivecrona et al., 2017), MOLLEO (Wang et al., 2025), and GPT-4o-mini (OpenAI, 2024).

Results. Table D16 compares these published results with our eight model settings. Our models were trained for numeric property targeting, not for this directional benchmark, and their RL reward contains no Tanimoto-similarity term. Nevertheless, the 20M Agentic CoM trained with RL directly from its pretrained checkpoint attains the highest SR$\times$Sim among our settings for both QED (0.407) and logP (0.701). For this model, SFT before RL increases directional success but reduces similarity and the resulting product. Training and inference budgets differ across systems, and training-molecule overlap has not been audited.

Table D16: **Directional property optimization on TOMG-Bench.** Baseline results are taken from Li et al. (2026b), without rerunning the methods. Our models use one rollout per request, conditioned on GED $[0, 2]$. SR is directional success; Sim is mean similarity over valid outputs. Both of our training routes start from pretrained checkpoints: RL omits SFT, whereas SFT + RL includes supervised finetuning before RL. Bold marks the best values in each column, including ties at the reported precision.

	QED			logP		
Model / training	SR	Sim	SR×Sim	SR	Sim	SR×Sim
Qwen2.5-3B-Instruct						
Base model	0.188	0.693	0.130	0.268	0.627	0.168
Distill-SFT	0.208	0.594	0.124	0.234	0.579	0.135
SFT	0.297	0.697	0.207	0.298	0.692	0.206
GRPO	0.138	**0.889**	0.123	0.379	0.806	0.305
GRPO (SFT init)	0.223	0.863	0.192	0.212	**0.863**	0.183
RePO	0.312	0.756	0.236	0.415	0.715	0.297
Llama-3.1-8B-Instruct						
Base model	0.159	0.722	0.115	0.237	0.692	0.164
SFT	0.185	0.809	0.150	0.287	0.763	0.219
GRPO	0.115	0.805	0.093	0.197	0.763	0.151
GRPO (SFT init)	0.121	0.811	0.098	0.192	0.775	0.149
RePO	0.243	0.783	0.190	0.360	0.746	0.269
Other baselines						
MolT5-large	0.465	0.119	0.055	0.424	0.102	0.043
BioT5-base	0.507	0.158	0.080	0.516	0.153	0.079
Graph GA	0.493	0.140	0.064	0.509	0.125	0.064
REINVENT	0.558	0.115	0.058	0.465	0.125	0.058
MOLLEO	0.496	0.122	0.061	0.510	0.103	0.053
GPT-4o-mini	0.231	0.752	0.174	0.499	0.706	0.352
Ours (20M)						
Agentic CoM, RL	0.586	0.694	**0.407**	0.965	0.726	**0.701**
CoM (self-generated), RL	0.489	0.672	0.329	0.497	0.691	0.344
Agentic CoM, SFT + RL	**0.664**	0.509	0.338	0.985	0.440	0.433
CoM (self-generated), SFT + RL	0.641	0.633	0.405	0.847	0.433	0.367
Ours (1M)						
Agentic CoM, RL	0.511	0.697	0.356	0.508	0.704	0.357
CoM (self-generated), RL	0.352	0.700	0.247	0.350	0.696	0.244
Agentic CoM, SFT + RL	0.585	0.600	0.351	**0.998**	0.370	0.369
CoM (self-generated), SFT + RL	0.531	0.466	0.247	0.950	0.075	0.072

E EXTENDED LIMITATIONS

Our experiments use computational objectives: deterministic functions for property targeting and pretrained predictors for bioactivity adaptation. The activity gains have not been experimentally validated, and chemical validity alone does not establish synthetic feasibility or biological efficacy. Both supervised examples and RL rewards are computationally derived. These settings do not test adaptation to noisy assays, delayed feedback, or optimization of multiple experimental endpoints.

We evaluate 1M and 20M models with one training seed per condition. Reported SEs reflect variation across evaluation tasks, not independent training runs; The group-size comparison does not establish an optimal allocation of rollouts across prompts. Our TOMG-Bench comparison (Appendix D.5) is not matched in training data or compute. Its GED $[0, 2]$ prompt band was selected through an exploratory sweep on the evaluation sets rather than independent validation data.

Supervised data for all formats reflect the proposal biases of the genetic search and its fragment library. Agentic formats use a fixed set of edit operations, but generate fragment arguments autoregressively rather than selecting them from a predefined library. These fragments remain subject to the environment's chemical-validity checks.

The action-marker analysis shows a preference for additions after RL (Figure D4), despite no explicit reward for particular action types. These counts describe generated traces, not necessarily successful edits, and do not establish that addition is the most effective strategy. The causes of this preference and its generality on tasks requiring other transformations remain untested.

Future work should examine larger models, multiple training seeds, synthesis-aware edits, and adaptation to experimental feedback.

F COMPUTATIONAL RESOURCES

Experiments used NVIDIA A100 GPUs (40 GB PCIe or 80 GB SXM4), with single-GPU jobs and multi-GPU 20M pretraining. We estimate that model training and evaluation required approximately $7{,}000$ GPU-hours, with reinforcement learning accounting for most of the cost. This estimate covers both model sizes, the property-targeting and bioactivity experiments, and the TOMG-Bench comparison. Multi-GPU runs are counted in device-hours.

Corpus construction, graph-edit-distance computation, rollout scoring, and genetic-search baselines required an estimated $80{,}000$ CPU-core-hours. This counts CPU-only work, excluding host CPU allocations accompanying GPU jobs. Both figures are rounded estimates extrapolated from available run timings and earlier resource estimates, rather than exact scheduler totals.