WHAT, WHERE, AND HOW TO EDIT: BENCHMARKING INSTRUCTION FOLLOWING FOR LOCALIZED MOLECULAR EDITING

Anonymous authors
Paper under double-blind review

ABSTRACT

Large language models are increasingly used to generate and modify drug-like molecules from natural-language instructions, yet existing benchmarks rarely test whether a model follows compositional editing requests precisely. We introduce ***IF-DrugBench***, a benchmark of 2,400 localized small-molecule editing tasks organized along three complementary axes: what to achieve, where to act, and how to use reference information. Each task is released in aligned one-dimensional SMILES and two-dimensional molecular-image forms, grounded in an experimentally reported molecular transformation, and paired one-to-one with machine-readable criteria and automated evaluation tools. The benchmark evaluates molecular validity, localization of substituent or linker edits, chemical-operation fidelity, explicit reference-motif transfer when requested, and property directions relative to the starting molecule. On a balanced 480-task evaluation subset, the evaluated models achieve at most 48.1% task pass rate (TPR) with text input and 51.5% TPR with image-conditioned input. These results show that current models still have substantial room for improvement in reliably following compositional molecular editing instructions and satisfying multiple constraints simultaneously. The benchmark and evaluation tools can be checked at https://anonymous.4open.science/r/ICLR-DrugBench-6D27.

1 INTRODUCTION

Large language models are showing increasing potential for supporting molecular design, such as generating or modifying drug-like molecules according to user-specified requirements (Edwards et al., 2022; Pei et al., 2023; Wang et al., 2025b; Zheng et al., 2025; Ling et al., 2025). In realistic drug discovery workflows, however, a model rarely satisfies all requirements in a single generation. Users often need to iteratively refine their requests, inspect generated molecules, and provide additional instructions until the candidate structure meets their desired objectives (Liu et al., 2024c; Zhu et al., 2025; Hu et al., 2026). Recent workflows attempt to formalize this iterative editing through conditional prompting (Zheng et al., 2024; Srinivas & Runkana, 2024; Yu et al., 2025) or agentic proposal-critique-revision loops (M. Bran et al., 2024; Liu et al., 2024d; Gao et al., 2025). However, the efficiency of this iterative process depends critically on the model's ability to accurately follow each user instruction. If the model modifies the wrong region, ignores a requested chemical operation, or optimizes one property while violating structural constraints, users must spend additional rounds correcting the output (Grisoni, 2023; Srinivas & Runkana, 2024). This makes the molecular design process inefficient, unreliable, and difficult to trust.

Despite this need, existing molecular generation benchmarks often focus heavily on broad distributional metrics (Brown et al., 2019; Polykovskiy et al., 2020; Wang et al., 2025a) or macro-level property optimization (Guevorguian et al., 2024), or rely on semantic text alignment scores that treat the task monolithically (Li et al., 2024). Consequently, these benchmarks typically evaluate only one aspect of this instruction space, such as property optimization or scaffold preservation. As a result, they cannot determine whether a model can satisfy rich, compositional user instructions that jointly constrain the goal, the edit location, and the intended use of reference information (Liu et al., 2025; Guo et al., 2023).

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.

To address this gap, we construct a localized small-molecule drug editing benchmark for instruction-following evaluation. First, through a literature review and a formative study with seven medicinal chemists, we identify three major axes and six subtypes of instructions that commonly arise when chemists interact with LLMs for molecular optimization. These instruction types capture not only desired optimization goals, but also the intended edit location and the way reference molecules should be used. Then, by combining these instruction types, we construct benchmark tasks that reflect realistic user requests for localized molecular editing. Each task provides a starting-molecule SMILES and a natural-language instruction specifying the desired property change and the editable substituent or linker region. Some tasks additionally specify how a reference molecule should be used. In addition, each task is derived from a validated molecular transformation, ensuring that the instruction is feasible and that at least one target molecule in the real world satisfies the specified edit and property objectives.

Furthermore, our benchmark is designed so that each instruction is paired with explicit and verifiable evaluation criteria and tools. Rather than relying solely on overall similarity to a target molecule, our benchmark evaluates whether each generated molecule satisfies the specific criteria encoded in the instruction, such as edit localization, reference usage, and property-direction constraints. We then assign automated evaluation tools that we develop to each criterion, so that a generated molecule can be judged according to the same constraints expressed in the prompt. This prompt-criteria-tool alignment makes the evaluation stricter and more interpretable, and allows us to diagnose which part of a compositional instruction a model fails to satisfy.

Finally, based on our evaluation results, we find that current models still struggle to reliably satisfy compositional molecular editing instructions. The strongest text-input model in our study reaches 48.1% TPR on the 480 evaluated tasks, while the strongest image-conditioned result reaches 51.5% TPR. Criterion pass rate (CPR) is higher than TPR, indicating that failures are usually compositional: a model may produce a valid molecule, preserve part of the context, or move one descriptor in the correct direction, yet still miss another requirement in the same instruction. Overall, our contributions are fourfold as follows.

- We define a three-axis taxonomy for localized molecular editing instructions, covering property objectives, chemical-operation objectives, substituent and linker scope, and explicit or implicit reference use.
- We construct ***IF-DrugBench***, a 2,400-task benchmark grounded in experimentally reported ChEMBL transformations and used as one-dimensional SMILES editing tasks to evaluate instruction-following molecular editing with machine-checkable chemical constraints.
- We build an aligned two-dimensional package for the same task identities, using rendered molecular diagrams with matched receiver and donor highlighting to test whether models can follow localized edit instructions from molecular images.
- We provide prompt-aligned criteria and automated evaluation tools, and use them to evaluate 18 model-representation settings on a balanced 480-task subset.

2 Related Works

2.1 LLMs for Molecule Generation

AI-driven molecular design has shifted from unconditional de novo generation toward interactive, instruction-guided editing. Early LLM-based chemistry systems treated molecule design as sequence generation over SMILES or SELFIES, using autoregressive decoding or cross-modal translation to produce valid molecular strings (Edwards et al., 2022; Pei et al., 2023; Irwin et al., 2022; Savit et al., 2025; Lu et al., 2023; Yue et al., 2024; Chitsaz et al., 2025; Thomas et al., 2024). More recent work frames LLMs as interactive editors or agents: conditional prompting methods such as Prompt-MolOpt and DrugLLM use matched molecular pairs or in-context learning to propose edits under ADMET constraints (Liu et al., 2024d; Zheng et al., 2024; Srinivas & Runkana, 2024; Lee et al., 2024; Wang et al., 2022; Wu et al., 2024), while agentic systems such as ChemCrow and ChatDrug place the model inside proposal–critique–revision loops with external tools and iterative feedback (M. Bran et al., 2024; Liu et al., 2024d; Zheng et al., 2025; Liu et al., 2024a; Ye et al., 2025; Zhuang et al., 2025; Liu et al., 2024b). These workflows make instruction following central: the model must correctly interpret what to optimize, where to edit, and which constraints to preserve.

2.2 Molecular Generation Benchmarks

Existing benchmarks do not yet evaluate this instruction-following ability at the required granularity. GuacaMol and MOSES focus on unconditional generation and broad distributional metrics such as validity, uniqueness, and Fréchet ChemNet Distance (Brown et al., 2019; Polykovskiy et al., 2020; Wang et al., 2025a); optimization suites such as Practical Molecular Optimization evaluate macro-level property improvement (Guevorguian et al., 2024; Park et al., 2023; Guo & Schwaller, 2024); and multimodal benchmarks such as S$\hat{2}$-Bench score semantic alignment between text prompts and molecular outputs (Li et al., 2024; Liu et al., 2023; Zhu et al., 2025). These settings typically treat instruction execution as a single global objective, relying on overall similarity, broad property shifts, or text–molecule alignment. As a result, they cannot diagnose whether a model satisfies compositional instructions that jointly specify the goal, the edit region, and the intended use of reference molecules. This motivates an evaluation framework based on prompt–criteria–tool alignment.

3 Methodology

We construct the benchmark with a deterministic pipeline that turns ChEMBL bioactivity records[1] into localized molecular editing instructions. The purpose of the pipeline is to make every task both realistic and diagnostically evaluable: the instruction should resemble how chemists ask for edits, the requested change should be feasible for the starting molecule, and the evaluation should test the exact constraints expressed in the prompt. Each task is grounded in a directed molecular transformation (x_0, x^*), where x_0 is the starting molecule and x^* is a hidden feasible molecule used only for dataset construction and validation. The pipeline has four stages: defining the instruction taxonomy, mining directional molecular transformations, converting transformations into localized instruction tasks, and aligning each prompt with explicit evaluation criteria.

3.1 Defining Instruction Taxonomy

Our taxonomy is designed to reflect how medicinal chemists describe their iterative molecular optimization requests to an LLM. We derive this taxonomy from two sources: a literature survey of LLM-based molecular design and a formative study with seven medicinal chemists. We observe that, in realistic use, an instruction is rarely a single unconstrained property target. Instead, medicinal chemists often combine three kinds of information: the desired optimization outcome, the molecular region where the model should act, and, when available, a reference molecule that helps the model infer a useful local design pattern.

We therefore organize instructions along three complementary axes. The first axis specifies *what to achieve*. A request may be property-driven (T1), where the model is asked to increase or decrease molecular properties relative to x_0, or operation-driven (T2), where the model is asked to perform a concrete local chemical operation such as linker shortening or ring removal. The second axis specifies *where to act*. We focus on localized edits with chemically identifiable anchor context: substituent edits (S1), where the editable region has one neighboring anchor, and linker edits (S2), where the editable region connects two neighboring anchors. The third axis specifies *how to learn*. A reference molecule can either provide an explicit local donor motif for grafting (R1), or be given as an example from which the model should infer the relevant editing strategy without being told exactly what structural correspondence to copy (R2). Reference information is optional: a task without a reference (NR) instantiates only the T and S axes and is not assigned a third R subtype.

3.2 Stage I: Mining Directional Molecular Transformations

Given the taxonomy, the next step is to find suitable starting molecules that can support those instruction types. A directed pair (x_0, x^*) can provide the essential ingredients of a benchmark task: a starting molecule to edit, a known feasible endpoint, the local structural difference that defines where the edit can occur, and measurable property or activity changes that define what the instruction can ask for. This approach helps us construct instructions while ensuring that each resulting instruction is satisfiable by at least one real-world molecule, rather than being an unconstrained or chemically

[1] https://www.ebi.ac.uk/chembl/

unrealistic request. We therefore start from the fixed collection of ChEMBL bioactivity records and extract pairs that provide feasible local edits with measurable property or activity differences.

Molecular pair collection. A molecular pair is only to be seen as a meaningful design example if the two molecules are compared in a consistent experimental and SAR context. In ChEMBL, each experimental record links a molecule to a biological target, an activity value, and the paper or document reporting the experiment. We group these records by biological target and document identifier, so molecules are paired only when they come from the same study and target the same biological target. Additionally, we also retain records with non-missing pChEMBL values, exact standard relations, and standard types for each ChEMBL target. These values can help to determine whether molecules are standardized, numerically defined, and based on common drug-discovery endpoints, making the direction of each molecular pair more reliable.

MCS-based pair filtering. Because our benchmark focuses on localized editing, each molecular pair should consist of two molecules that are mostly the same except for a local structural change. We therefore compare molecules within each document cluster using the maximum common substructure (MCS), the largest shared subgraph between two molecules. A large MCS indicates that the molecules share a common core and differ mainly in a local region, making the pair suitable for an editing task. We keep only pairs whose shared heavy-atom structure is sufficiently large relative to the larger molecule. Molecules are parsed and sanitized with RDKit, and MCS matching is performed with valence and ring constraints to avoid chemically incompatible matches. Specifically, we retain a pair only if

$$\rho(m_i, m_j) = \frac{H(\mathrm{MCS}(m_i, m_j))}{\max(H(m_i), H(m_j))} \geq 0.65,$$

where m_i and m_j are two candidate molecules, $\mathrm{MCS}(m_i, m_j)$ is their largest shared molecular substructure, and $H(\cdot)$ counts heavy atoms, i.e., all non-hydrogen atoms. The score $\rho(m_i, m_j)$ therefore measures how much of the larger molecule is covered by the structure shared by both molecules. A low value indicates that the molecules are too different overall, so the pair is removed because it cannot support a localized edit instruction.

Structural labeling and direction assignment. After filtering for similar pairs, we need to decide what kind of localized instruction the pair can support and which molecule should serve as the starting point. We assign an initial structural label using Murcko scaffolds. If two molecules have identical canonical Murcko scaffolds, the pair is treated as a substituent-edit candidate. Otherwise, we compute the MCS ratio between their Murcko scaffolds; pairs with a scaffold-level MCS ratio at least 0.45 are treated as linker-like continuous edits. We retain only substituent and linker candidates, because these provide localized editable regions with reliable anchor context. It is important to note that these are merely the initial labels; the realistic labels will still be determined and verified in subsequent steps.

Each retained undirected pair is then oriented into $x_0 \rightarrow x^*$. For an enumerated pair (A, B), we compute the activity difference as $\Delta a = a_C(B) - a_C(A)$, where $a_C(x)$ denotes the measured activity of molecule x under target or assay context C. If the activity difference is large enough to indicate a clear improvement, we orient the pair toward the more active molecule. When activity is approximately unchanged, we instead orient the pair using a lightweight physicochemical preference score that favors higher quantitative estimate of drug-likeness (QED) and lower molecular weight, lower octanol–water partition coefficient (logP), and lower topological polar surface area (TPSA). Specifically, higher QED indicates a more favorable overall drug-like profile, while lower molecular weight, logP, and TPSA generally reflect simpler structures with reduced risks related to excessive size, lipophilicity, or polarity. These criteria are consistent with common medicinal chemistry practices, where molecules are often optimized not only for potency but also for more favorable physicochemical and developability profiles.

3.3 Stage II: Constructing Instruction Tasks

After mining feasible directed pairs, we convert each pair into a natural-language instruction that a user would plausibly provide to an LLM. The hidden molecule x^* is used to ensure that the task is chemically feasible.

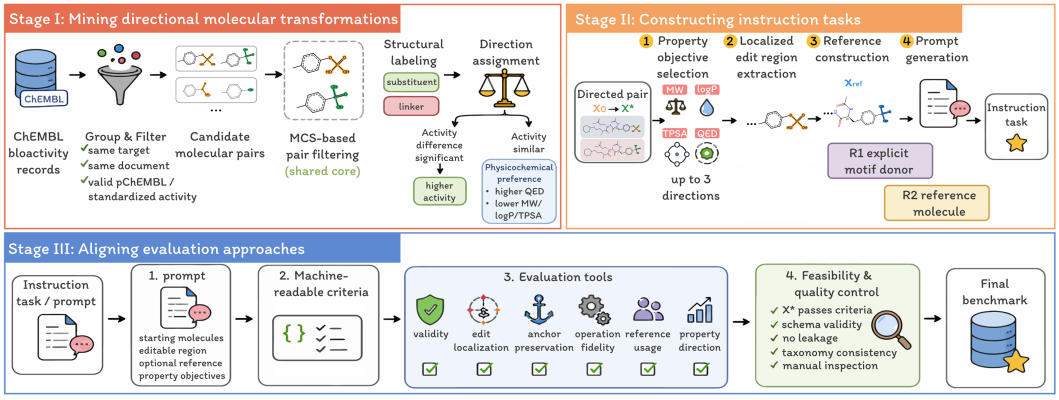


Figure 1: Stage I-III of the benchmark construction pipeline.

Property objective selection. The property objectives should reflect real differences between x_0 and the feasible target, but they should not overload the prompt with too many simultaneous requirements. For every directed pair, we compute RDKit 2D descriptors for both x_0 and x^*, such as molecular weight, logP, TPSA, QED, hydrogen-bond counts, rotatable bonds, ring counts, heteroatom count, heavy atom count, Lipinski violations, and topological descriptors. For each descriptor d, we compute $\Delta d = d(x^*) - d(x_0)$. Count-like changes smaller than 0.25 are ignored. Each instruction can select at most three objectives, stated only as directions relative to x_0.

Localized edit region extraction. To specify where the model should edit, we derive the x_0-side difference atoms from the MCS with the hidden feasible molecule x^* and split them into connected components. We retain a single connected component with one outside anchor as an S1 substituent edit, or a connected path with two distinct outside anchors as an S2 linker edit. Scaffold-level heuristics provide candidate labels, but the final S label is retained only after anchor-aware topology validation. Internal heavy-atom ranks parameterize the evaluator; the model-facing prompt renders the same atom set as an ASCII annotation under the canonical SMILES, avoiding ambiguous numeric ranges.

Reference construction. For reference-based tasks, we select a third molecule x_{ref} from the same document cluster, excluding x_0 and x^*. A candidate reference must share several RDKit descriptor directions with x^* relative to x_0, while also containing at least one counter-directional descriptor. The counter-directional descriptor is used to construct instructions that prevent LLMs from directly copying x_{ref}. Furthermore, it should contain a local structural pattern relevant to the change needed in x^* (i.e., reference sites). For R1 tasks, these sites are exposed as explicit motif-transfer hints. For R2 tasks, the prompt shows only x_{ref}, requiring the model to infer the editing strategy without motif ranks or atom correspondences.

Prompt generation. Prompts are generated by gpt-4o-mini[2] from the structured task specification and expose only user-facing information: the starting molecule SMILES, an ASCII annotation of the editable region, an optional reference molecule, and one to three property or operation objectives. The hidden feasible molecule x^* is never shown. For S1 and S2 tasks, the annotation marks the editable atoms directly under the displayed SMILES and the text states the required anchor-preservation constraint. Internal heavy-atom ranks remain available to the evaluator but are not exposed as ambiguous numeric ranges in the prompt. Thus, each prompt is precise enough for localized editing while avoiding leakage of the hidden target.

Two-dimensional task construction. For every retained task, we also create an aligned 2D molecular-image package. The 2D package uses the same benchmark index, taxonomy label, starting molecule, optional reference molecule, criteria, and evaluator configuration as the 1D JSON record. The only model-facing change is representational: the starting molecule is rendered as a

[2]https://developers.openai.com/api/docs/models/gpt-4o-mini

molecular diagram with the receiver edit region highlighted, and reference-based tasks additionally include a rendered reference molecule. For R1, the donor motif is highlighted because the instruction explicitly asks for motif transfer. For R2, the reference image is unannotated beyond ordinary molecule rendering, so the model receives the example structure but not hidden donor-site supervision. The hidden feasible molecule x^* and its rendered annotations remain evaluator-only artifacts. This paired design lets us ask whether molecular images improve instruction following while holding task identity and scoring criteria fixed.

3.4 Stage III: Aligning Evaluation Approaches

The final stage converts each instruction into an evaluable task. Prompts, criteria, and evaluation tools are derived from the same structured task specification, ensuring that every stated requirement is directly checked. Instead of measuring only similarity to x^*, we evaluate whether the generated molecule satisfies the explicit constraints in the instruction.

Prompt–criteria–tool alignment. For each task, every requirement stated in the prompt is converted into a machine-readable criterion and assigned to an automated evaluation tool. We implement tools for validity checks, edit localization, anchor preservation, chemical-operation fidelity, reference usage, and RDKit-based property directions; details are provided in the Appendix. Formally, let P_i be the prompt for task i, $C_i = \{c_{i1}, \ldots, c_{ik}\}$ the criteria derived from it, and $E_i = \{e_{i1}, \ldots, e_{ik}\}$ the corresponding evaluation tools. We enforce a one-to-one mapping $c_{ij} \leftrightarrow e_{ij}$. A generated molecule $\hat{x}$ passes only if all criteria are satisfied.

Feasibility and quality control. We verify each generated instruction task through multiple steps. First, we apply the assigned evaluation tools to the hidden target molecule x^*; any pair for which x^* fails the criteria is discarded:

$$\mathrm{Pass}(x^*; P_i, C_i, E_i) = 1.$$

Second, the generated prompt is checked for schema validity, taxonomy consistency, and leakage constraints, such as preventing R2 instructions from exposing explicit motif ranks. Third, we enforce prompt–criteria–tool alignment, so every requirement stated in the prompt has a corresponding machine-readable criterion and evaluation tool. Finally, we use LLM-assisted checking and manual inspection to identify inconsistent prompts, missing criteria, weak evaluation tools, or chemically ambiguous cases. Together, these checks ensure that each benchmark instruction is well-formed and feasible, meaning that a sufficiently capable model could in principle generate a chemically valid molecule satisfying the stated requirements.

4 Experiments

We evaluate model outputs on a balanced 480-task subset of ***IF-DrugBench***. The subset contains 40 tasks in each of the 12 combinations formed by two goal types (T1/T2), two edit locations (S1/S2), and three reference settings (No Reference/R1/R2). Each evaluated model sees the same task indices within a representation, and models evaluated in both representations use the same 480 task identities for the 1D and 2D comparison. The 1D benchmark input is the textual package described in Section 3: canonical SMILES, an ASCII annotation of the editable receiver region, an optional reference SMILES, and the output contract. We evaluate eleven 1D settings: GPT-5.6 Sol (OpenAI, 2026), Claude Opus 5 (Anthropic, 2026), Gemini 3.7 Flash (Google DeepMind, 2026), Qwen3.7 Plus (Alibaba Cloud Model Studio, 2026), Kimi K3 (Team et al., 2026), GLM-5.2 and GLM-5.3 (Zeng et al., 2026), Qwen3.5-397B-A17B (Qwen Team, 2026), DeepSeek-V4-Pro (Xu et al., 2026), ChemDFM-v1.5-8B (Zhao et al., 2024), and ChemLLM-20B-Chat-SFT (Zhang et al., 2024). The 2D benchmark input keeps the same task semantics and evaluator configuration but adds rendered molecular diagrams for the starting molecule and an optional reference molecule. We evaluate seven image-conditioned settings: GPT-5.6 Sol, Claude Opus 5, Gemini 3.7 Flash, Qwen3.7 Plus, Kimi K3, Qwen3.5-397B-A17B, and ChemDFM-X (Zhao et al., 2024). ChemDFM-v1.5-8B, ChemLLM-20B-Chat-SFT, and ChemDFM-X are chemistry-domain baselines. In the 2D setting, model outputs may not fully follow the requested serialization format. To ensure fair evaluation, we transform every single generated SMILES token that RDKit can parse and sanitize into RDKit canonical isomeric SMILES before applying the downstream evaluation tools. Invalid, multi-component, or non-SMILES outputs are still rejected. We report two metrics. *Task pass rate* (TPR)

is the percentage of tasks for which every assigned criterion passes. *Criterion pass rate* (CPR) is the macro-average percentage of top-level criteria satisfied per task.

5 RESULTS

5.1 TEXT-INPUT EDITING PERFORMANCE

Tables 1 and 2 report task pass rate (TPR) and criterion pass rate (CPR) for the 1D SMILES-based benchmark. TPR gives credit only when every assigned evaluator passes, whereas CPR measures the average fraction of top-level criteria satisfied per task. The rightmost Avg. column reports row-wise means, and the final Avg. row reports column-wise averages across models.

Table 1: Task pass rate (TPR) on the 1D benchmark. Among model rows, best, second-best, and third-best values in each numeric column are shown in **bold**, underlined, and marked with †.

Access	Model	T1/S1			T1/S2			T2/S1			T2/S2			Avg.
		NR	R1	R2	NR	R1	R2	NR	R1	R2	NR	R1	R2	
Closed	GPT-5.6 Sol	42.5	25.0†	47.5†	35.0	30.0	25.0†	47.5†	10.0†	42.5	12.5	10.0†	20.0	29.0†
	Claude Opus 5	60.0	30.0	57.5	47.5	**40.0**	52.5	57.5	**30.0**	**52.5**	12.5	20.0	27.5	40.6
	Gemini 3.7 Flash	**62.5**	**45.0**	**70.0**	**65.0**	27.5†	**57.5**	**62.5**	**30.0**	50.0	**15.0**	**42.5**	**50.0**	**48.1**
	Qwen3.7 Plus	17.5	10.0	30.0	32.5	12.5	25.0†	27.5	12.5	37.5	12.5	5.0	10.0	19.4
Open-weight	Kimi K3	22.5	7.5	30.0	45.0†	17.5	22.5	30.0	10.0†	45.0†	7.5	10.0†	20.0	22.3
	GLM-5.2	15.0	7.5	22.5	25.0	2.5	10.0	15.0	5.0	20.0	10.0†	7.5	12.5	12.7
	GLM-5.3	45.0†	0.0	40.0	2.5	2.5	2.5	27.5	2.5	10.0	5.0	0.0	2.5	11.7
	Qwen3.5-397B-A17B	25.0	17.5	35.0	25.0	17.5	25.0†	27.5	10.0†	17.5	2.5	10.0†	17.5	19.2
	DeepSeek-V4-Pro	15.0	5.0	22.5	25.0	12.5	15.0	12.5	5.0	37.5	5.0	10.0†	25.0†	15.8
Domain	ChemDFM-v1.5-8B	2.5	0.0	5.0	10.0	0.0	7.5	12.5	0.0	5.0	0.0	5.0	2.5	4.2
	ChemLLM-20B-Chat-SFT	2.5	0.0	5.0	10.0	0.0	2.5	0.0	0.0	2.5	2.5	2.5	5.0	2.7
Avg.		28.2	13.4	33.2	29.3	14.8	22.3	29.1	10.5	29.1	7.7	11.1	17.5	20.5

The strongest text-input model is Gemini 3.7 Flash, with 48.1 TPR and 78.0 CPR. Claude Opus 5 follows at 40.6 TPR and 75.1 CPR, while GPT-5.6 Sol is third by TPR at 29.0. Among open-weight models, Kimi K3 has the best TPR/CPR balance. A striking implication of the cell-level table is that high CPR can be misleadingly optimistic. Several models satisfy many individual criteria on average, but their TPR collapses once all criteria must hold at the same time. This is exactly the behavior a localized editing benchmark should expose.

Table 2: Criterion pass rate (CPR) on the 1D benchmark. Ranking markers follow Table 1.

Access	Model	T1/S1			T1/S2			T2/S1			T2/S2			Avg.
		NR	R1	R2	NR	R1	R2	NR	R1	R2	NR	R1	R2	
Closed	GPT-5.6 Sol	82.5†	60.0†	78.1†	80.6†	66.0†	73.1†	73.0†	54.2	75.0	60.0†	40.8	56.5†	66.7†
	Claude Opus 5	87.5	66.5	86.9	85.0	**70.5**	85.0	83.0	**72.5**	**88.0**	60.5	54.2	61.0	75.0
	Gemini 3.7 Flash	**90.6**	**74.0**	**90.0**	**89.4**	68.0	**87.5**	**86.0**	65.8	81.0†	**62.0**	**65.4**	**76.0**	**78.0**
	Qwen3.7 Plus	66.9	45.5	67.5	75.6	52.0	68.8	63.5	54.2	74.5	52.0	35.8	46.5	58.6
Open-weight	Kimi K3	62.5	46.0	71.2	78.1	52.0	69.4	61.0	54.6†	82.0	51.5	42.9†	52.5	60.3
	GLM-5.2	50.6	46.5	59.4	64.4	40.5	53.1	55.5	42.5	55.0	45.0	37.5	48.5	49.9
	GLM-5.3	68.8	3.0	57.5	20.6	5.0	16.2	38.5	9.2	16.0	22.0	3.3	12.5	22.7
	Qwen3.5-397B-A17B	68.8	52.0	67.5	63.8	56.0	65.0	62.0	50.8	57.5	43.5	40.4	46.5	56.1
	DeepSeek-V4-Pro	51.9	40.0	53.1	64.4	43.5	55.6	48.0	43.8	67.0	40.0	35.4	52.0	49.6
Domain	ChemDFM-v1.5-8B	46.9	26.5	41.9	47.5	21.5	44.4	47.0	26.2	33.5	27.0	23.8	31.5	34.8
	ChemLLM-20B-Chat-SFT	35.6	38.0	45.0	47.5	28.5	48.1	31.5	39.2	48.5	34.0	30.8	38.5	38.8
Avg.		64.8	45.3	65.3	65.2	45.8	60.6	59.0	46.6	61.6	45.2	37.3	47.5	53.7

The cell-level results are especially informative along the S and R axes. S2 linker edits are generally harder than S1 substituent edits, even though both are localized transformations. S2 requires the model to preserve two receiver anchors and redesign only the path between them, creating a constrained reconnection problem. By contrast, S1 usually involves a single attachment point, so a model can often succeed by modifying or replacing a pendant substituent while leaving the parent structure intact. Another clearer pattern emerges along the reference axis. R1, which requires an

explicit donor graft from the reference molecule, is often worse than the no-reference setting. This does not mean that reference molecules are useless; rather, it shows that forcing a model to reuse a specific molecular fragment turns the task into a much stricter graph-editing problem. The model must not only generate a valid and property-improving molecule, but also identify the donor motif, preserve its attachment topology, and graft it onto the designated receiver site. Many models appear able to propose plausible edits when they are not constrained by an exact donor requirement, but fail once the user specifies what fragment must be transferred and where it should attach. R2 behaves differently. Because R2 provides a reference molecule as a softer design guide rather than an exact fragment-transfer constraint, it does not consistently hurt performance, but it also does not reliably improve it over the no-reference setting. In some cells R2 is better, while in others the no-reference condition remains competitive. This suggests that current models do not yet use reference molecules as robust sources of transferable chemical experience. They may imitate superficial motifs or follow the property direction independently, but they often fail to extract the design rationale from the reference and apply it to the new molecular context.

5.2 Image-Conditioned Editing Performance

Tables 3 and 4 repeat the same 480 task identities in the 2D molecular-image setting. The model sees rendered molecule diagrams with receiver and, for R1, donor highlighting. Given that image-conditioned models may not fully adhere to the requested output format, we transform every SMILES token that RDKit can parse and sanitize into RDKit-canonical isomeric SMILES before evaluation, ensuring a fair comparison.

Table 3: Task pass rate (TPR) on the 2D benchmark. Ranking markers follow Table 1.

Access	**Model**	**T1/S1**			**T1/S2**			**T2/S1**			**T2/S2**			**Avg.**
		NR	R1	R2	NR	R1	R2	NR	R1	R2	NR	R1	R2	
Closed	GPT-5.6 Sol	2.5	0.0	5.0†	22.5	0.0	2.5	10.0†	0.0†	5.0†	7.5†	5.0†	10.0†	5.8
	Claude Opus 5	40.0	20.0	45.0	52.5	20.0	45.0	**60.0**	7.5	45.0	22.5	30.0	42.5	35.8
	Gemini 3.7 Flash	**57.5**	**47.5**	**47.5**	**60.0**	**42.5**	**55.0**	**60.0**	**47.5**	**57.5**	**45.0**	**45.0**	**52.5**	**51.5**
	Qwen3.7 Plus	5.0	2.5†	5.0†	30.0†	2.5†	12.5†	7.5	0.0†	2.5	7.5†	5.0†	5.0	7.1†
Open-weight	Kimi K3	0.0	0.0	2.5	17.5	0.0	0.0	7.5	0.0†	2.5	0.0	0.0	7.5	3.1
	Qwen3.5-397B-A17B	5.0	0.0	5.0†	15.0	0.0	5.0	5.0	0.0†	5.0†	0.0	2.5	2.5	3.8
Domain	ChemDFM-X	10.0†	0.0	0.0	20.0	0.0	0.0	20.0	0.0†	0.0	7.5†	0.0	0.0	4.8
Avg.		17.1	10.0	15.7	31.1	9.3	17.1	24.3	7.9	16.8	12.9	12.5	17.1	16.0

Table 4: Criterion pass rate (CPR) on the 2D benchmark. Ranking markers follow Table 1.

Access	**Model**	**T1/S1**			**T1/S2**			**T2/S1**			**T2/S2**			**Avg.**
		NR	R1	R2	NR	R1	R2	NR	R1	R2	NR	R1	R2	
Closed	GPT-5.6 Sol	46.9	36.5	58.1†	67.5	43.0	51.2	50.0	30.0	45.0	45.0	32.5	51.0†	46.4
	Claude Opus 5	81.2	62.0	78.1	78.8	61.0	79.4	**79.5**	50.8	75.0	62.0	62.9	66.5	69.8
	Gemini 3.7 Flash	**86.2**	**74.5**	**80.0**	**85.6**	**74.0**	**86.2**	79.0	**68.8**	**81.5**	**73.5**	**70.4**	**78.0**	**78.2**
	Qwen3.7 Plus	64.4†	41.0	51.9	70.6†	43.5†	63.8†	53.5	35.0†	47.0†	49.0†	37.1	48.5	50.4†
Open-weight	Kimi K3	48.1	42.0†	51.9	65.0	38.5	55.0	51.0	32.9	36.0	44.5	38.3†	46.0	45.8
	Qwen3.5-397B-A17B	58.1	40.0	51.2	61.9	38.0	47.5	45.0	34.2	40.0	44.5	35.8	42.5	44.9
Domain	ChemDFM-X	55.6	18.5	13.8	62.5	20.0	21.9	56.0†	13.3	17.5	48.5	7.1	9.0	28.6
Avg.		62.9	44.9	55.0	70.3	45.4	57.9	59.1	37.9	48.9	52.4	40.6	48.8	52.0

The 2D results are highly concentrated. Gemini 3.7 Flash reaches 51.5 TPR and 78.1 CPR, becoming the strongest image-conditioned setting and slightly exceeding its own 1D TPR. Claude Opus 5 is second at 35.8 TPR and 69.8 CPR. ChemDFM-X, the domain-specific 2D baseline, reaches 4.8 TPR and 28.6 CPR; it remains close to the weaker open-weight image-conditioned models and far below Gemini and Claude. The main remaining failures should be interpreted primarily as failures of image-grounded graph editing, reference transfer, operation fidelity, and multi-criterion satisfaction rather than mere serialization failures. The R1 cells are especially revealing. The model must not only recognize the donor motif in the reference image; it must attach that motif at the specified

receiver boundary. Gemini is the only image-conditioned model with strong R1 performance, while other models often collapse near zero. This suggests that current multimodal molecule understanding remains uneven: visual highlighting can help locate an edit site, but it does not automatically become a correct site-specific graph transplant.

5.3 Representation Shift and Error Structure

Table 5 suggests that 2D molecular understanding is not the same skill as 1D molecular editing. A model that performs well on canonical SMILES can still collapse when the same task is presented as a diagram. This is clearest for GPT-5.6 Sol, whose TPR drops from 29.0 to 5.8, and for Kimi K3, which drops from 22.3 to 3.1. They point to a deeper failure: the model does not reliably convert visual molecular information into the exact graph constraints required by the evaluator. The exception is Gemini 3.7 Flash. Its 2D TPR is higher than its 1D TPR, suggesting that visual highlighting can help when the model has strong enough multimodal graph grounding. This makes the 2D benchmark more than a harder version of the 1D benchmark. It tests whether diagrams can actually improve localized molecular editing, rather than merely decorate the prompt. For most models, they do not; for Gemini, they sometimes do. This contrast is useful because it separates chemical fluency from representation-grounded control. Many models can generate valid molecules and satisfy some criteria, but the benchmark asks whether the same output preserves the specified region, anchors, reference relation, operation, and property direction. The 2D setting makes this coordination problem harder: the model must align visual marks with graph edits. The results therefore expose a multimodal gap that would be hidden by validity-only or similarity-based evaluation.

Table 5: Average performance across molecular representations for models evaluated in both 1D and 2D. Δ is 2D minus 1D. Ranking markers follow Table 1.

Access	**Model**	**1D TPR**	**1D CPR**	**2D TPR**	**2D CPR**	**Δ TPR**	**Δ CPR**
Closed	GPT-5.6 Sol	29.0†	66.7†	5.8	46.4	-23.1	-20.3
	Claude Opus 5	<u>40.6</u>	<u>75.0</u>	<u>35.8</u>	<u>69.8</u>	<u>-4.8</u>	<u>-5.3</u>
	Gemini 3.7 Flash	**48.1**	**78.0**	**51.5**	**78.2**	**3.3**	**0.2**
	Qwen3.7 Plus	19.4	58.6	7.1†	50.4†	-12.3†	-8.1†
Open-weight	Kimi K3	22.3	60.3	3.1	45.8	-19.2	-14.5
	Qwen3.5-397B-A17B	19.2	56.1	3.8	44.9	-15.4	-11.2

6 Conclusion

We present ***IF-DrugBench***, a benchmark for evaluating whether LLMs can follow localized molecular editing instructions with explicit, machine-checkable constraints. ***IF-DrugBench*** combines a three-axis instruction taxonomy, experimentally grounded feasible transformations, aligned 1D and 2D task packages, and automated prompt-matched evaluation tools. The 480-task model evaluation shows that current models remain far from reliable compositional editing: the best TPR values are 48.1% for 1D input and 51.5% for 2D input, and most models exhibit much higher CPR than TPR. These findings point to a concrete evaluation gap for molecular design systems: generating plausible molecules is not enough; models must preserve the right context, modify the right region, use references in the intended way, and satisfy all stated objectives simultaneously.

7 Limitations

IF-DrugBench focuses on localized molecular editing rather than unconstrained molecule generation. It does not cover every medicinal-chemistry interaction, such as multi-turn optimization, synthetic route planning, cost, novelty, protein-structure-conditioned design, or experimental activity prediction. The automated evaluators are strict by design and may reject chemically plausible alternatives that violate the requested graph-locality, reference-transfer, or descriptor-direction constraints. Finally, descriptor-direction objectives support scalable evaluation, but they do not replace potency, selectivity, permeability, toxicity, or synthesizability assays.

AI USE STATEMENT

Generative AI tools were used to assist with code implementation and language editing. All AI-generated code and manuscript text were reviewed and verified by the authors. The authors take responsibility for the final content of this work.

REPRODUCIBILITY STATEMENT

The methodology specifies the pair-mining, localization, instruction-generation, and evaluator-alignment procedures. The appendix documents the released task schema, descriptor contract, RDKit version, evaluation tools, quality-control checks, dataset distributions, and representative instances. The accompanying benchmark files provide the model-facing instructions and machine-readable evaluation configurations needed to reproduce the reported dataset statistics and task-level evaluation.

REFERENCES

Alibaba Cloud Model Studio. qwen3.7-plus-us. https://docs.modelstudio.console.alibabacloud.com/en/model-studio/qwen3-7-plus-us, 2026. Accessed: 2026-09-01.

Anthropic. Claude Opus 5 system card. https://anthropic.com/claude-opus-5-system-card, July 2026. Accessed: 2026-09-01.

Nathan Brown, Marco Fiscato, Marwin HS Segler, and Alain C Vaucher. Guacamol: benchmarking models for de novo molecular design. *Journal of chemical information and modeling*, 59(3): 1096–1108, 2019.

Kamran Chitsaz, Roshan Balaji, Quentin Fournier, Nirav Pravinbhai Bhatt, and Sarath Chandar. Novomolgen: Rethinking molecular language model pretraining. *arXiv preprint arXiv:2508.13408*, 2025.

Carl Edwards, Tuan Lai, Kevin Ros, Garrett Honke, Kyunghyun Cho, and Heng Ji. Translation between molecules and natural language. In *Proceedings of the 2022 Conference on Empirical Methods in Natural Language Processing*, pp. 375–413, 2022.

Bowen Gao, Yanwen Huang, Yiqiao Liu, Wenxuan Xie, Wei-Ying Ma, Ya-Qin Zhang, and Yanyan Lan. Pushing the boundaries of structure-based drug design through collaboration with large language models. *arXiv preprint arXiv:2503.01376*, 2025.

Google DeepMind. Gemini 3.7 Flash model card. https://deepmind.google/models/model-cards/gemini-3-7-flash/, August 2026. Accessed: 2026-09-01.

Francesca Grisoni. Chemical language models for de novo drug design: Challenges and opportunities. *Current Opinion in Structural Biology*, 79:102527, 2023.

Philipp Guevorguian, Menua Bedrosian, Tigran Fahradyan, Gayane Chilingaryan, Hrant Khachatrian, and Armen Aghajanyan. Small molecule optimization with large language models. *arXiv preprint arXiv:2407.18897*, 2024.

Jeff Guo and Philippe Schwaller. Beam enumeration: probabilistic explainability for sample efficient self-conditioned molecular design. In *International Conference on Learning Representations*, volume 2024, pp. 53252–53292, 2024.

Taicheng Guo, Kehan Guo, Bozhao Nan, Zhenwen Liang, Zhichun Guo, Nitesh V. Chawla, Olaf Wiest, and Xiangliang Zhang. What can large language models do in chemistry? a comprehensive benchmark on eight tasks. *Advances in neural information processing systems*, 36:59662–59688, 2023.

Xuanning Hu, Anchen Li, Qianli Xing, Jinglong Ji, Hao Tuo, and Bo Yang. Empowering llms for structure-based drug design via exploration-augmented latent inference. In *Proceedings of the ACM Web Conference 2026*, pp. 4244–4255, 2026.

Ross Irwin, Spyridon Dimitriadis, Jiazhen He, and Esben Jannik Bjerrum. Chemformer: a pre-trained transformer for computational chemistry. *Machine Learning: Science and Technology*, 3 (1):015022, 2022.

Seul Lee, Karsten Kreis, Srimukh P Veccham, Meng Liu, Danny Reidenbach, Saee Paliwal, Arash Vahdat, and Weili Nie. Molecule generation with fragment retrieval augmentation. *Advances in Neural Information Processing Systems*, 37:132463–132490, 2024.

Jiatong Li, Junxian Li, Weida Wang, Yunqing Liu, Changmeng Zheng, Yatao Bian, Dongzhan Zhou, Xiao-yong Wei, and Qing Li. Speak-to-structure: Evaluating llms in open-domain natural language-driven molecule generation. *arXiv preprint arXiv:2412.14642*, 2024.

Xiaowen Ling, Zhiqiang Li, Yanbin Wang, and Zhuhong You. Transformers in protein: A survey. *arXiv preprint arXiv:2505.20098*, 2025.

Gang Liu, Michael Sun, Wojciech Matusik, Meng Jiang, and Jie Chen. Multimodal large language models for inverse molecular design with retrosynthetic planning. In *International Conference on Learning Representations*, volume 2025, pp. 41744–41771, 2025.

Pengfei Liu, Yiming Ren, Jun Tao, and Zhixiang Ren. Git-mol: A multi-modal large language model for molecular science with graph, image, and text. *Computers in biology and medicine*, 171:108073, 2024a.

Shengchao Liu, Weili Nie, Chengpeng Wang, Jiarui Lu, Zhuoran Qiao, Ling Liu, Jian Tang, Chaowei Xiao, and Animashree Anandkumar. Multi-modal molecule structure–text model for text-based retrieval and editing. *Nature Machine Intelligence*, 5(12):1447–1457, 2023.

Shengchao Liu, Jiongxiao Wang, Yijin Yang, Chengpeng Wang, Ling Liu, Hongyu Guo, and Chaowei Xiao. Conversational drug editing using retrieval and domain feedback. In *International Conference on Learning Representations*, volume 2024, pp. 48325–48361, 2024b.

Xianggen Liu, Yan Guo, Haoran Li, Jin Liu, Shudong Huang, Bowen Ke, and Jiancheng Lv. Drugllm: Open large language model for few-shot molecule generation. *arXiv preprint arXiv:2405.06690*, 2024c.

Xuefeng Liu, Chih-chan Tien, Peng Ding, Songhao Jiang, and Rick L Stevens. Entropy-reinforced planning with large language models for drug discovery. *arXiv preprint arXiv:2406.07025*, 2024d.

Hao Lu, Zhiqiang Wei, Xuze Wang, Kun Zhang, and Hao Liu. Graphgpt: a graph enhanced generative pretrained transformer for conditioned molecular generation. *International Journal of Molecular Sciences*, 24(23):16761, 2023.

Andres M. Bran, Sam Cox, Oliver Schilter, Carlo Baldassari, Andrew D White, and Philippe Schwaller. Augmenting large language models with chemistry tools. *Nature machine intelligence*, 6(5):525–535, 2024.

OpenAI. GPT-5.6 Sol model. `https://developers.openai.com/api/docs/models/gpt-5.6-sol`, 2026. Accessed: 2026-09-01.

Ryan Park, Ryan Theisen, Navriti Sahni, Marcel Patek, Anna Cichońska, and Rayees Rahman. Preference optimization for molecular language models. *arXiv preprint arXiv:2310.12304*, 2023.

Qizhi Pei, Wei Zhang, Jinhua Zhu, Kehan Wu, Kaiyuan Gao, Lijun Wu, Yingce Xia, and Rui Yan. 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.

Daniil Polykovskiy, Alexander Zhebrak, Benjamin Sanchez-Lengeling, Sergey Golovanov, Oktai Tatanov, Stanislav Belyaev, Rauf Kurbanov, Aleksey Artamonov, Vladimir Aladinskiy, Mark Veselov, et al. Molecular sets (moses): a benchmarking platform for molecular generation models. *Frontiers in pharmacology*, 11:565644, 2020.

Qwen Team. Qwen3.5: Towards native multimodal agents. `https://qwen.ai/blog?id=qwen3.5`, February 2026. Accessed: 2026-09-01.

Anagha Savit, Harikrishna Sahu, Shivank Shukla, Wei Xiong, and Rampi Ramprasad. polybart: A chemical linguist for polymer property prediction and generative design. *arXiv preprint arXiv:2506.04233*, 2025.

Sakhinana Sagar Srinivas and Venkataramana Runkana. Crossing new frontiers: Knowledge-augmented large language model prompting for zero-shot text-based de novo molecule design. *arXiv preprint arXiv:2408.11866*, 2024.

Kimi Team, Tongtong Bai, Yifan Bai, Yiping Bao, Jianfeng Cai, Xinyuan Cai, Peizhou Cao, Yuxuan Cao, Ziwei Chai, Y Charles, et al. Kimi k3: Open frontier intelligence. *arXiv preprint arXiv:2607.24653*, 2026.

Morgan Thomas, Mazen Ahmad, Gary Tresadern, and Gianni De Fabritiis. Promptsmiles: prompting for scaffold decoration and fragment linking in chemical language models. *Journal of Cheminformatics*, 16(1):77, 2024.

Huibin Wang, Zehui Wang, Minghua Shi, Zixian Cheng, and Ying Qian. Enhancing unconditional molecule generation via online knowledge distillation of scaffolds. *Molecules*, 30(6):1262, 2025a.

Zichao Wang, Weili Nie, Zhuoran Qiao, Chaowei Xiao, Richard Baraniuk, and Anima Anandkumar. Retrieval-based controllable molecule generation. *arXiv preprint arXiv:2208.11126*, 2022.

Ziqing Wang, Kexin Zhang, Zihan Zhao, Yibo Wen, Abhishek Pandey, Han Liu, and Kaize Ding. A survey of large language models for text-guided molecular discovery: from molecule generation to optimization. *arXiv preprint arXiv:2505.16094*, 2025b.

Zhenxing Wu, Odin Zhang, Xiaorui Wang, Li Fu, Huifeng Zhao, Jike Wang, Hongyan Du, Dejun Jiang, Yafeng Deng, Dongsheng Cao, et al. Leveraging language model for advanced multi-property molecular optimization via prompt engineering. *Nature Machine Intelligence*, 6(11): 1359–1369, 2024.

Anyi Xu, Bangcai Lin, Bing Xue, Bingxuan Wang, Bingzheng Xu, Bochao Wu, Bowei Zhang, Chaofan Lin, Chen Dong, Chenchen Ling, et al. Deepseek-v4: Towards highly efficient million-token context intelligence. *arXiv preprint arXiv:2606.19348*, 2026.

Geyan Ye, Xibao Cai, Houtim Lai, Xing Wang, Junhong Huang, Longyue Wang, Wei Liu, and Xiangxiang Zeng. Drugassist: A large language model for molecule optimization. *Briefings in Bioinformatics*, 26(1):bbae693, 2025.

Jiajun Yu, Yizhen Zheng, Huan Yee Koh, Shirui Pan, Tianyue Wang, and Haishuai Wang. Collaborative expert llms guided multi-objective molecular optimization. *arXiv preprint arXiv:2503.03503*, 2025.

Jie Yue, Bingxin Peng, Yu Chen, Jieyu Jin, Xinda Zhao, Chao Shen, Xiangyang Ji, Chang-Yu Hsieh, Jianfei Song, Tingjun Hou, et al. Unlocking comprehensive molecular design across all scenarios with large language model and unordered chemical language. *Chemical Science*, 15(34):13727–13740, 2024.

Aohan Zeng, Xin Lv, Zhenyu Hou, Zhengxiao Du, Qinkai Zheng, Bin Chen, Da Yin, Chendi Ge, Chenghua Huang, Chengxing Xie, et al. Glm-5: from vibe coding to agentic engineering. *arXiv preprint arXiv:2602.15763*, 2026.

Di Zhang, Wei Liu, Qian Tan, Jingdan Chen, Hang Yan, Yuliang Yan, Jiatong Li, Weiran Huang, Xiangyu Yue, Wanli Ouyang, et al. Chemllm: A chemical large language model. *arXiv preprint arXiv:2402.06852*, 2024.

Zihan Zhao, Bo Chen, Jingpiao Li, Lu Chen, Liyang Wen, Pengyu Wang, Zichen Zhu, Danyang Zhang, Ziping Wan, Yansi Li, Zhongyang Dai, et al. Chemdfm-x: towards large multimodal model for chemistry. *Science China Information Sciences*, 67(12), 2024.

Yizhen Zheng, Huan Yee Koh, Maddie Yang, Li Li, Lauren T May, Geoffrey I Webb, Shirui Pan, and George Church. Large language models in drug discovery and development: From disease mechanisms to clinical trials. *arXiv preprint arXiv:2409.04481*, 2024.

Yizhen Zheng, Huan Yee Koh, Jiaxin Ju, Madeleine Yang, Lauren T May, Geoffrey I Webb, Li Li, Shirui Pan, and George Church. Large language models for drug discovery and development. *Patterns*, 6(10), 2025.

Wenyu Zhu, Chengzhu Li, Xiaohe Tian, Yifan Wang, Yinjun Jia, Jianhui Wang, Bowen Gao, Ya-Qin Zhang, Wei-Ying Ma, and Yanyan Lan. Coder as editor: Code-driven interpretable molecular optimization. *arXiv preprint arXiv:2510.14455*, 2025.

Xiang Zhuang, Keyan Ding, Tianwen Lyu, Yinuo Jiang, Xiaotong Li, Zhuoyi Xiang, Zeyuan Wang, Ming Qin, Kehua Feng, Jike Wang, et al. Advancing biomolecular understanding and design following human instructions. *Nature Machine Intelligence*, 7(7):1154–1167, 2025.

A Detailed Model-Evaluation Tables and Error Analysis

A.1 Criterion-Family Error Patterns

Tables 6 and 7 break the aggregate scores into evaluator families. Each column is computed only over tasks where the corresponding evaluator applies, so Ground-S1 and Ground-S2 should be read as localization checks for different edit scopes, Intent-T2 as the chemical-operation check, and Ref.-R1 as the explicit donor-transfer check.

Table 6: Criterion-family pass rates on the 1D benchmark.

Model	Valid	Non-copy	Ground-S1	Ground-S2	Intent-T2	Ref.-R1	Property	TPR	CPR
GPT-5.6 Sol	97.5	96.0	52.5	32.1	33.8	18.8	56.0	29.0	66.7
Claude Opus 5	99.2	97.9	76.2	46.7	54.2	30.0	61.5	40.6	75.0
Gemini 3.7 Flash	98.8	98.3	68.3	52.9	53.3	36.2	74.6	48.1	78.0
Qwen3.7 Plus	96.9	86.2	47.9	25.4	32.5	10.0	37.7	19.4	58.6
Kimi K3	98.5	86.7	48.8	33.3	37.9	11.2	37.3	22.3	60.3
GLM-5.2	90.0	77.7	30.0	19.2	20.0	5.6	32.9	12.7	49.9
GLM-5.3	30.0	29.8	25.0	3.8	9.2	1.2	21.9	11.7	22.7
Qwen3.5-397B-A17B	94.8	80.6	44.2	27.9	27.9	13.8	36.2	19.2	56.1
DeepSeek-V4-Pro	99.8	62.7	32.5	23.8	27.5	8.1	28.5	15.8	49.6
ChemDFM-v1.5-8B	84.4	50.8	14.6	11.7	9.2	1.3	9.8	4.2	34.8
ChemLLM-20B-Chat-SFT	88.3	67.3	15.8	11.2	10.4	0.6	10.4	2.7	38.8

Table 7: Criterion-family pass rates on the 2D benchmark.

Model	Valid	Non-copy	Ground-S1	Ground-S2	Intent-T2	Ref.-R1	Property	TPR	CPR
GPT-5.6 Sol	82.5	81.7	5.4	13.3	8.8	1.2	40.2	5.8	46.4
Claude Opus 5	96.9	91.2	54.2	47.9	46.7	19.4	62.9	35.8	69.8
Gemini 3.7 Flash	95.2	94.8	60.8	59.6	59.6	45.6	79.0	51.5	78.1
Qwen3.7 Plus	94.0	92.5	10.4	18.8	10.0	2.5	30.4	7.1	50.4
Kimi K3	88.8	87.7	5.0	9.2	6.2	0.0	29.8	3.1	45.8
Qwen3.5-397B-A17B	88.1	85.6	7.5	14.2	6.7	0.6	24.0	3.8	44.9
ChemDFM-X	50.4	46.9	17.5	20.0	13.8	0.0	9.2	4.8	28.6

The criterion-family view shows that TPR remains low because models often fail the structural criteria that require the edit to stay within the annotated receiver region. Ground-S2 is lower than Ground-S1 for many settings because a linker edit must preserve and reconnect two anchors, while S1 requires only one protected attachment anchor.

The hardest family remains explicit donor transfer. Ref.-R1 is low across model families and is especially weak for the domain baselines. This failure is stricter than simply not containing a familiar substructure: the evaluator checks that the donor graph appears with the required receiver-anchor topology. The low R1 rates therefore mean that models frequently use the reference as a loose design hint rather than grafting the specified motif at the specified boundary.

Property direction also remains brittle. Models may produce valid, non-copy molecules while still moving one requested descriptor in the wrong direction, especially when the prompt asks for multiple objectives. This explains why CPR can be high while TPR is still modest: a molecule can pass validity, non-copying, and part of the grounding requirement but fail the final task because one property or operation constraint is missed.

A.2 Instruction-Axis Marginals

Tables 8 and 9 marginalize performance over the three taxonomy axes. Each cell reports TPR/CPR, so the left number measures complete task success and the right number measures partial criterion satisfaction.

Table 8: Marginal 1D results along each instruction axis. Cells report TPR/CPR (%). NR denotes the absence of a reference molecule.

Model	T1	T2	S1	S2	NR	R1	R2	Overall
GPT-5.6 Sol	34.2/73.4	23.8/59.9	35.8/70.5	22.1/62.8	34.4/74.0	18.8/55.2	33.8/70.7	29.0/66.7
Claude Opus 5	47.9/80.2	33.3/69.9	47.9/80.7	33.3/69.4	44.4/79.0	30.0/65.9	47.5/80.2	40.6/75.0
Gemini 3.7 Flash	54.6/83.2	41.7/72.7	53.3/81.2	42.9/74.7	51.2/82.0	36.2/68.3	56.9/83.6	48.1/78.0
Qwen3.7 Plus	21.2/62.7	17.5/54.4	22.5/62.0	16.2/55.1	22.5/64.5	10.0/46.9	25.6/64.3	19.4/58.6
Kimi K3	24.2/63.2	20.4/57.4	24.2/62.9	20.4/57.7	26.2/63.3	11.2/48.9	29.4/68.8	22.3/60.3
GLM-5.2	13.8/52.4	11.7/47.3	14.2/51.6	11.2/48.2	16.2/53.9	5.6/41.8	16.2/54.0	12.7/49.9
GLM-5.3	15.4/28.5	7.9/16.9	20.8/32.2	2.5/13.3	20.0/37.5	1.2/5.1	13.8/25.6	11.7/22.7
Qwen3.5-397B-A17B	24.2/62.2	14.2/50.1	22.1/59.8	16.2/52.5	20.0/59.5	13.8/49.8	23.8/59.1	19.2/56.1
DeepSeek-V4-Pro	15.8/51.4	15.8/47.7	16.2/50.6	15.4/48.5	14.4/51.1	8.1/40.7	25.0/56.9	15.8/49.6
ChemDFM-v1.5-8B	4.2/38.1	4.2/31.5	4.2/37.0	4.2/32.6	6.2/42.1	1.2/24.5	5.0/37.8	4.2/34.8
ChemLLM-20B-Chat-SFT	3.3/40.5	2.1/37.1	1.7/39.6	3.8/37.9	3.8/37.2	0.6/34.1	3.8/45.0	2.7/38.8

Table 9: Marginal 2D results along each instruction axis. Cells report TPR/CPR (%). NR denotes the absence of a reference molecule.

Model	T1	T2	S1	S2	NR	R1	R2	Overall
GPT-5.6 Sol	5.4/50.5	6.2/42.2	3.8/44.4	7.9/48.4	10.6/52.3	1.2/35.5	5.6/51.3	5.8/46.4
Claude Opus 5	37.1/73.4	34.6/66.1	36.2/71.1	35.4/68.4	43.8/75.4	19.4/59.2	44.4/74.8	35.8/69.8
Gemini 3.7 Flash	51.7/81.1	51.2/75.2	52.9/78.3	50.0/78.0	55.6/81.1	45.6/71.9	53.1/81.4	51.5/78.1
Qwen3.7 Plus	9.6/55.9	4.6/45.0	3.8/48.8	10.4/52.1	12.5/59.4	2.5/39.1	6.2/52.8	7.1/50.4
Kimi K3	3.3/50.1	2.9/41.5	2.1/43.7	4.2/47.9	6.2/52.2	0.0/37.9	3.1/47.2	3.1/45.8
Qwen3.5-397B-A17B	5.0/49.5	2.5/40.3	3.3/44.8	4.2/45.0	6.2/52.4	0.6/37.0	4.4/45.3	3.8/44.9
ChemDFM-X	5.0/32.0	4.6/25.2	5.0/29.1	4.6/28.2	14.4/55.7	0.0/14.7	0.0/15.5	4.8/28.6

The 1D marginals show that operation-driven tasks are generally harder than property-only tasks. T2 must satisfy the same validity, non-copy, grounding, and property checks, then additionally match the named operation. S2 linker tasks also remain difficult because the generated edit has to reconnect the same two protected anchors. A molecule that preserves only one side of the context, changes the linker too globally, or creates a plausible but differently connected scaffold fails this scope requirement.

The R-axis is the clearest compositional stress test. R1 is typically below both NR and R2 because it asks for explicit motif transfer, not just reference-inspired design. R2 can sometimes help by providing a broad example without forcing an exact donor graft, while R1 requires boundary-preserving graph alignment between the donor and receiver. This distinction is why R1 remains weak even after output canonicalization removes many avoidable serialization penalties.

In 2D, Gemini 3.7 Flash is the only model with balanced performance across T, S, and R axes. For the other image-conditioned settings, including ChemDFM-X, the diagram often helps less than expected because the model must still convert visual molecular information into a valid graph edit and a valid SMILES output. The low R1 marginals show that visual donor highlighting alone is not enough for reliable site-specific transfer.

A.3 Paired Representation Outcomes

Table 10 partitions the 480 shared task identities for models evaluated in both representations. This table is stricter than comparing averages, because it asks whether the same task is solved under both input formats.

Table 10: Paired task-pass outcomes for models evaluated with both 1D and 2D inputs. The four outcome columns partition the same 480 tasks.

Model	Both pass	1D only	2D only	Neither	Total
GPT-5.6 Sol	10	129	18	323	480
Claude Opus 5	106	89	66	219	480
Gemini 3.7 Flash	141	90	106	143	480
Qwen3.7 Plus	10	83	24	363	480
Kimi K3	6	101	9	364	480
Qwen3.5-397B-A17B	6	86	12	376	480

The paired view shows that 2D does not simply scale 1D performance by a fixed factor. Gemini passes 106 tasks only in 2D and 90 tasks only in 1D, while 141 tasks pass in both representations. These 2D-only successes suggest that highlighting can help the model identify the intended local region. At the same time, models with weaker image-conditioned reasoning still lose many tasks when moving from text to diagrams.

The overlap pattern also explains why both representations are useful for evaluation. A model may understand an ASCII-annotated SMILES instruction but fail to recover the same graph from an image, or it may benefit from a diagram on a particular case while losing another case because the graph edit or reference transfer is not executed correctly. Reporting only the average 1D–2D delta would hide these cross-over cases.

A.4 Multi-Criterion Failure Burden

Table 11 counts how many top-level evaluator checks fail for each task. A zero-failure task is a TPR success; one- and two-failure tasks show near misses; the final column captures broad breakdowns where several constraints fail together.

Table 11: Distribution of the number of failed top-level criterion checks per task. The zero-failure column is the numerator of TPR.

Input	Model	0 failed	1 failed	2 failed	$\geq$3 failed	Mean passed	Mean assigned
1D	GPT-5.6 Sol	139	90	104	147	3.1	4.8
1D	Claude Opus 5	195	101	84	100	3.6	4.8
1D	Gemini 3.7 Flash	231	76	92	81	3.7	4.8
1D	Qwen3.7 Plus	93	91	94	202	2.8	4.8
1D	Kimi K3	107	89	85	199	2.9	4.8
1D	GLM-5.2	61	73	90	256	2.4	4.8
1D	GLM-5.3	56	35	33	356	1.0	4.8
1D	Qwen3.5-397B-A17B	92	80	94	214	2.7	4.8
1D	DeepSeek-V4-Pro	76	62	68	274	2.4	4.8
1D	ChemDFM-v1.5-8B	20	39	78	343	1.6	4.8
1D	ChemLLM-20B-Chat-SFT	13	43	98	326	1.9	4.8
2D	GPT-5.6 Sol	28	75	140	237	2.2	4.8
2D	Claude Opus 5	172	91	86	131	3.3	4.8
2D	Gemini 3.7 Flash	247	75	84	74	3.7	4.8
2D	Qwen3.7 Plus	34	74	127	245	2.4	4.8
2D	Kimi K3	15	59	148	258	2.2	4.8
2D	Qwen3.5-397B-A17B	18	56	135	271	2.1	4.8
2D	ChemDFM-X	23	50	82	325	1.3	4.8

The failure-burden table shows why TPR remains much lower than CPR. Strong models have many partial successes, but they also have nontrivial sets of tasks with three or more failed checks. For

example, Gemini 3.7 Flash in 1D passes 231 tasks completely, yet 81 tasks still fail at least three criteria. In 2D, Gemini improves to 247 full passes, but 74 tasks still have broad failures. These are not one-line formatting mistakes; they usually combine grounding, operation, reference, or property errors. Near misses are useful for development because they identify outputs that satisfy most of the instruction but miss one bottleneck. Broad-failure cases point to a deeper issue: the model has not preserved enough of the protected molecular context or has not executed the requested local edit at all. This is why the benchmark reports both TPR and CPR. TPR reflects whether a user receives a molecule satisfying the full instruction, while CPR reveals whether the model is making partial progress on the individual constraints.

B Benchmark Description

B.1 Release Size and Factorial Coverage

The final release contains 2,400 tasks. It is exactly balanced across the two goal types (T1 and T2), the two edit-location types (S1 and S2), and the three reference settings (No Reference, R1, and R2). Each experimental cell contains 200 tasks, as shown in Table 12. Consequently, T1 and T2 each contain 1,200 tasks, S1 and S2 each contain 1,200 tasks, and the three reference settings each contain 800 tasks. This design supports controlled comparisons along one instruction axis without requiring post-hoc reweighting by the other axes.

Table 12: Composition of the final 2,400-task release. NR denotes tasks with no reference input; it is not an instruction subtype.

Goal	Edit location	No Reference (NR)	R1	R2	Total
T1	S1	200	200	200	600
T1	S2	200	200	200	600
T2	S1	200	200	200	600
T2	S2	200	200	200	600
Total		**800**	**800**	**800**	**2,400**

Furthermore, to ensure diversity, the benchmark contains 1,013 unique starting structures, 939 unique hidden feasible structures, 1,232 unique directed (x_0, x^*) pairs, and 1,242 unique primary localized edits (Table 12). Reference-based tasks draw on 458 unique reference structures. The difference between pair and edit counts arises because a directed pair may support more than one valid local region, whereas paired R1/R2 views intentionally reuse the same transformation to isolate the effect of explicit versus implicit reference guidance.

Table 13: High-level statistics of the final release.

Statistic	Value
Benchmark tasks	2,400
Biological targets	64
Unique x_0 structures	1,013
Unique x^* structures	939
Unique directed pairs	1,232
Unique primary edits	1,242
Unique reference structures	458
Property requirements	5,217
Editable-region size	2–12 heavy atoms
Mean / median region size	5.00 / 4
RDKit version	2024.09.4

B.2 Property and Edit Distributions

Figure 2 summarizes the controlled design. Every task contains one to three property-direction requirements relative to x_0. The final release includes 393 single-property tasks, 1,197 two-property

Table 14: Property-direction requirements in the final release. A dash indicates that the direction is not instantiated.

Property	**Increase**	**Decrease**	**Total**
Quantitative estimate of drug-likeness (QED)	1,153	–	1,153
Topological polar surface area (TPSA)	341	229	570
Number of hydrogen-bond acceptors	282	211	493
Number of heteroatoms	282	209	491
Number of hydrogen-bond donors	285	148	433
Octanol–water partition coefficient (MolLogP)	–	395	395
Fraction of sp^3 carbons	297	–	297
Heavy-atom count	–	289	289
Molecular weight	–	285	285
Number of rotatable bonds	–	255	255
Lipinski-rule violations	–	144	144
Number of amide bonds	–	123	123
Number of aliphatic rings	–	98	98
Number of aromatic rings	–	98	98
Total ring count	–	93	93
All requirements	**2,640**	**2,577**	**5,217**

tasks, and 810 three-property tasks. In total, 2,640 requirements request an increase and 2,577 request a decrease, giving a near-even global split of 50.6% versus 49.4%. This balance prevents a model from obtaining a large systematic advantage by always producing smaller or simpler molecules. Directionality is nevertheless descriptor-dependent: for example, a quantitative estimate of drug-likeness is requested only as an increase, whereas molecular weight and number of rotatable bonds are requested only as decreases. Table 14 reports the complete objective distribution.

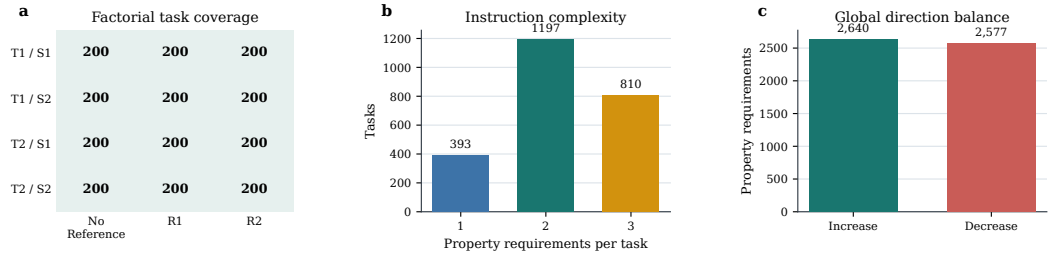


Figure 2: Composition and instruction complexity of the final release. (a) Each combination of goal type, edit-location type, and reference setting contains exactly 200 tasks. No Reference denotes the absence of the reference axis, not an additional R subtype. (b) Distribution of the number of property requirements per task. (c) Aggregate direction counts across all 5,217 property requirements.

All editable regions contain at least two heavy atoms; no single-atom edit remains in the release. There are 1,280 tasks with 2–4 editable atoms, 830 with 5–8 atoms, and 290 with 9–12 atoms. T2 tasks instantiate three operation subtypes derived from the structural delta between x_0 and x^*: 609 local fragment replacements, 379 linker-shortening operations, and 212 ring-removal or ring-opening operations. The operation label is generated jointly with its evaluator parameters, so the prompt and operation-fidelity criterion refer to the same subtype.

Finally, Figure 3 has shown that aggregate direction balance does not imply an artificial symmetric distribution for every descriptor. Some objectives have a chemically meaningful preferred direction in this release: QED and fraction of sp^3 carbons are requested as increases, whereas molecular weight, MolLogP, rotatable bonds, and Lipinski-rule violations are requested as decreases. TPSA, hydrogen-bond counts, and heteroatom counts occur in both directions. Thus, a model cannot pass by applying one universal shrink-or-expand strategy; it must identify the descriptor and requested direction in each instruction.

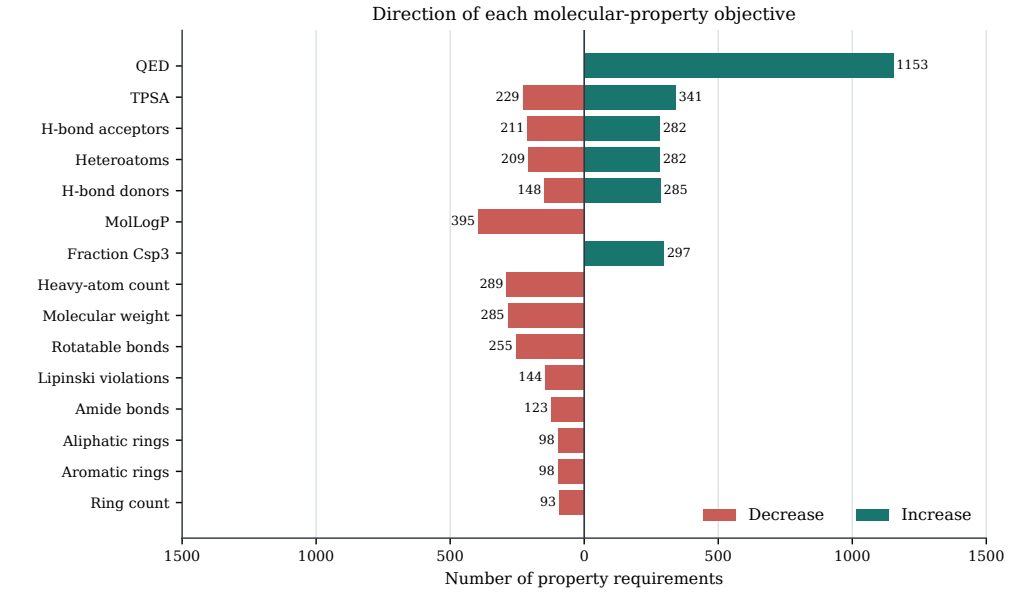


Figure 3: Direction distribution for every molecular-property objective. Bars to the left denote decrease requirements, and bars to the right denote increase requirements. The two sides sum to 5,217 requirements.

B.3 Structural-Edit and Donor Complexity

Figure 4 characterizes the structural scope of the instructions. S1 and S2 regions contain an average of 5.26 and 4.73 heavy atoms, respectively. Their distributions overlap but are not identical: two- and three-atom regions are more common for S2, while S1 retains more medium-sized regions. The operation distribution is also location-dependent. Linker shortening occurs only for S2, while fragment replacement is more common for S1; ring removal or opening is represented in both types.

The R1 donor-transfer graphs contain between one and twelve heavy atoms. Of the 800 R1 tasks, 594 (74.3%) use compact one- to three-atom donor graphs. Compactness does not make these tasks trivial: the evaluator still requires the complete donor atom and bond states, the correct receiver anchors, and the donor boundary topology. Larger transfer graphs are retained when the chemically relevant transferable motif cannot be represented by a smaller connected graph.

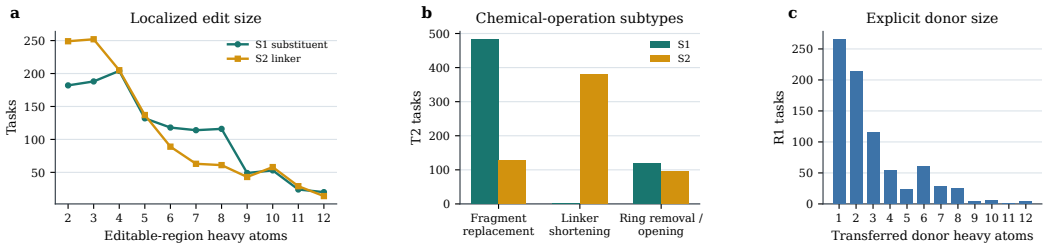


Figure 4: Structural complexity of localized edits and explicit references. (a) Editable-region size for S1 substituent and S2 linker tasks. (b) T2 operation subtypes split by edit-location type. (c) Number of transferred donor heavy atoms in the 800 R1 tasks.

B.4 Biological-Target Diversity

The 2,400 tasks span 64 biological targets and several broad target families. GPCR-associated records contribute 866 tasks across 23 targets, enzyme-associated records contribute 676 tasks across 15 targets, and kinase-associated records contribute 584 tasks across 15 targets. The remaining 274

tasks cover epigenetic, nuclear-receptor, protease, ion-channel liability, transporter, and bacterial targets. No target contributes more than 12 cases to any one of the 12 experimental cells. This per-cell cap reduces domination by a single medicinal-chemistry series while preserving enough within-target examples for controlled analysis.

The family distribution is intentionally broad but not forced to be uniform. GPCRs, enzymes, and kinases provide the largest number of experimentally observed local transformations, while the remaining families test transfer to less frequent settings. Even the largest individual target, HDAC1, contributes only 105 tasks (4.4% of the release), and no target contributes more than 12 tasks to any single experimental cell.

B.5 Molecular Profile of Starting and Feasible Structures

Table 15 summarizes the molecular profile of x_0 and the hidden feasible structures. These statistics describe the selected release rather than universal optimization preferences. On average, x^* has lower molecular weight, MolLogP, heavy-atom count, and rotatable-bond count, and higher QED. However, individual tasks are evaluated only against their stated requirements; an unrequested global trend is never converted into an additional passing criterion. The broad $P10$–$P90$ ranges for x_0 also show that the benchmark is not confined to a narrow molecular-size or polarity regime.

Table 15: Molecular descriptors for the 2,400 starting and hidden feasible structures. $P10$–$P90$ is reported for x_0 to characterize the input range; Δ is the mean of $d(x^*) - d(x_0)$.

Descriptor	x_0 **mean**	x_0 **median**	x_0 $P10$–$P90$	x^* **mean**	x^* **median**	**Mean** Δ
Molecular weight	425.516	413.587	299.399–561.639	397.894	388.396	-27.623
Heavy-atom count	30.207	29.000	22.000–40.000	28.342	28.000	-1.865
Quantitative estimate of drug-likeness	0.514	0.518	0.229–0.786	0.567	0.587	$+0.054$
MolLogP	3.783	3.885	1.689–5.620	3.381	3.430	-0.402
Topological polar surface area	83.099	77.230	35.940–135.750	81.368	75.110	-1.731
Hydrogen-bond acceptors	5.263	5.000	3.000–8.000	5.111	5.000	-0.152
Rotatable bonds	6.254	6.000	3.000–10.000	5.645	5.000	-0.609
Ring count	3.800	4.000	3.000–5.000	3.694	4.000	-0.106
Fraction of sp^3 carbons	0.339	0.333	0.091–0.591	0.322	0.316	-0.017

B.6 Objective Co-occurrence

Property objectives are sampled as sets, not as independent duplicated tasks. The ten most frequent pairs shown in Figure 5 account for recurring medicinal-chemistry trade-offs, but no single pair dominates the release: the most frequent pair, QED with TPSA, occurs in 205 tasks (8.5%). When reporting model performance, we recommend giving both the overall all-criteria pass rate and stratified rates by T, S, reference setting, number of property requirements, edit-region size, and operation subtype. This prevents gains on high-frequency task patterns from obscuring failures on less common compositional instructions.

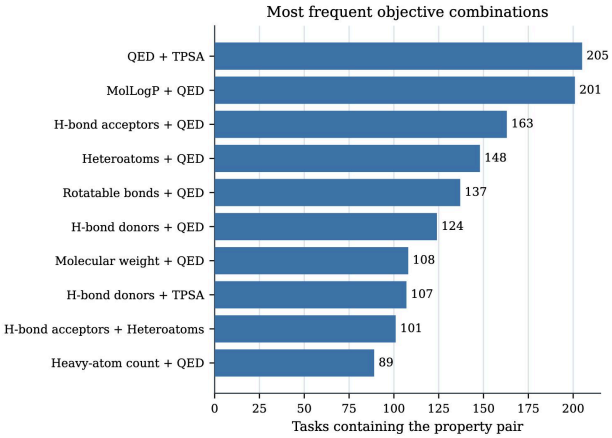


Figure 5: Ten most frequent unordered property pairs among multi-property instructions.

C Task Representation and Packaging

C.1 Structured Record Schema

Each benchmark instance is first represented as a structured record from which the instruction prompt, criteria, and evaluator configuration are generated. Table 16 lists the principal fields. The hidden target x^* is retained for construction, feasibility testing, and evaluator regression tests, but is never included in model input.

Table 16: Main fields in a benchmark task record.

Field	Description
`benchmark_index`	Dense global release index from 0 to 2,399.
`X0`	Starting molecule, canonical SMILES, stored RDKit descriptors, and the internal editable-region atom set.
`X_star`	Hidden feasible molecule and descriptors used only for construction and validation.
`xref`	Optional reference structure. R1 additionally stores the donor-transfer graph; R2 exposes no donor ranks or transfer annotation.
`strategy_taxonomy`	One goal subtype (T1 or T2), one location subtype (S1 or S2), and, only when a reference is supplied, R1, R2, or NR.
`prompts`	An instruction containing canonical SMILES, ASCII region annotation, preservation constraints, and property or operation objectives.
`criteria`	Ordered, machine-readable passing requirements derived from the instruction prompt.
`evaluation`	Ordered evaluation-tool assignments in one-to-one correspondence with `criteria`.

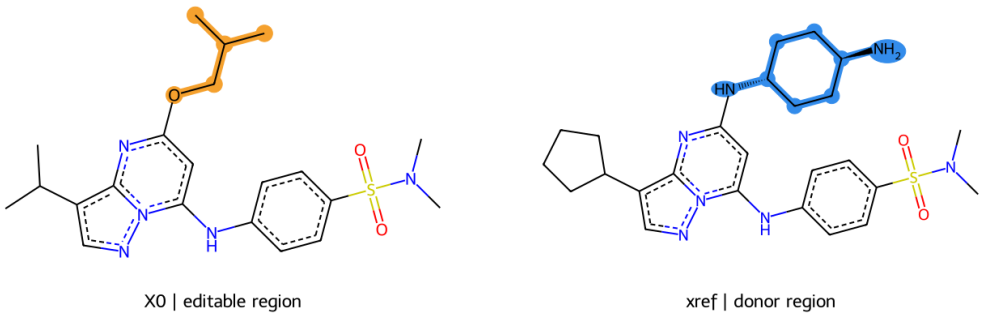
C.2 One-Dimensional and Two-Dimensional Packages

The one-dimensional release stores the instruction prompt, molecular records, criteria, and evaluation-tool parameters in JSON. The two-dimensional package preserves the same global index and task identity while adding rendered molecular diagrams. The model-facing package contains the annotated x_0 image and, for reference-based tasks, an x_{ref} image. In R1, the donor region is highlighted; in R2, no donor region is exposed. The evaluator-only package contains x^* and its structural annotations for regression and feasibility checks. The hidden target image and evaluator metadata are never part of the model input. A release-wide alignment check verifies that all 2,400 one-dimensional records correspond exactly to their two-dimensional directories and indices.

The following one-dimensional excerpt and Figure 6 show the aligned model-facing representations of case 281, a T1/S1/R1 task. The 1D record provides canonical SMILES and SMILES-aligned ASCII annotations, whereas the 2D package communicates the same receiver and donor regions through color highlighting.

One-dimensional example. The complete JSON record additionally stores the criteria and evaluator parameters omitted from this abbreviated model input.

```
taxonomy: T1 / S1 / R1
Starting molecule with the annotated substituent receiver region:
CC(C)COc1cc(Nc2ccc(S(=O)(=O)N(C)C)cc2)n2ncc(C(C)C)c2n1
^^ ^ ^^A
|__ MODIFY THIS REGION
Reference molecule with the annotated donor-transfer region:
CN(C)S(=O)(=O)c1ccc(Nc2cc(N[C@H]3CC[C@H](N)CC3)nc3c(C4CCCC4)cnn23)cc1
                        A ^^^^^^ ^^^^^^^ ^ ^^
                          |__ COPY THIS DONOR REGION
Goals: decrease octanol-water partition coefficient; increase hydrogen
    -bond donors; decrease rotatable bonds.
Output: exactly one RDKit-canonical SMILES and no other text.
```



(a) Annotated receiver region in x_0

(b) Annotated donor region in x_{ref}

Figure 6: Two-dimensional model input for the same case as the one-dimensional excerpt above. Orange marks the editable receiver region and blue marks the explicit R1 donor; the hidden target is not exposed to the model.

C.3 Instruction Realization

The released one-dimensional model-facing prompt uses an ASCII annotation placed directly under the canonical starting-molecule SMILES. Carets mark editable atoms, and `A` marks protected neighboring anchors. This representation avoids asking a model to interpret numeric heavy-atom ranges, RDKit zero-based indices, or SMILES character positions. For S1, one protected attachment anchor is shown; for S2, two protected anchors define the linker endpoints. Internal rank sets are still retained in the structured evaluator record so that the graphical and textual tasks can be checked deterministically.

The prompt then states the localization constraint, the optional reference instruction, and one to three property objectives using explicit property names. T2 prompts additionally state one operation subtype. Every prompt ends with a strict response contract: the model must return exactly one RDKit-canonical, single-component SMILES and no explanatory text.

C.4 Reference Presentation

No-reference tasks contain only the annotated x_0 and the requested objectives. R1 tasks show an annotated receiver region in x_0 and an annotated donor-transfer region in x_{ref}. The instruction explicitly asks the model to graft every donor atom and internal bond and reproduce the donor boundary topology at the receiver anchors. R2 tasks provide an unannotated x_{ref} as a design example. They deliberately omit donor ranks, correspondence sites, and copy instructions, requiring the model to infer a useful local design strategy. R1 and R2 tasks are paired whenever possible: they share x_0,

x^*, x_{ref}, edit location, operation, and property objectives, while differing only in how the reference is presented and evaluated.

D Evaluation Tools and Quality Assurance

D.1 Criterion-Level Pass Rule

A task passes only when every criterion derived from its prompt passes. Table 17 reports both the role and release frequency of each evaluator. All tasks include validity, non-identity, grounding, and property-direction checks. T2 contributes an operation-fidelity check. R1 contributes an explicit donor-graft check, whereas R2 has no reference-copy evaluator because copying a particular hidden motif is not stated in the R2 instruction.

Table 17: Evaluation tools used in the final 2,400-task release.

Tool	Tasks	Passing requirement
`sanity:valid_smiles`	2,400	Output is one SMILES token that RDKit can parse and sanitize; it is normalized to canonical isomeric SMILES and must be single-component without dummy atoms or atom-map annotations.
`sanity:not_identical_x0`	2,400	Canonical generated structure differs from canonical x_0.
`property:rdkit_descriptors_direction`	2,400	Every requested descriptor moves in its stated direction relative to x_0.
`intent:chemical_operation_fidelity`	1,200	The requested T2 operation subtype is realized inside the grounded edit region.
`grounding:linker_anchor_preserved`	1,200	S2 edit forms a nonempty connected linker that reconnects the same two anchors while preserving the protected context.
`grounding:substituent_local_edit`	1,200	S1 edit remains inside the annotated substituent; its attachment anchor and exact protected context are preserved.
`referencing:xref_fragment_graft_explicit`	800	The complete R1 donor-transfer graph and its boundary topology appear at the receiver anchors.

D.2 Anchor-Aware Grounding

Grounding is evaluated by graph decomposition rather than by a global similarity threshold. For S1, the evaluator matches the protected outside context as an exact induced atom-and-bond graph, requires a change inside the editable substituent, preserves the attachment bond state and normalized anchor stereochemistry, and rejects new disconnected edit components. For S2, it additionally requires a nonempty connected edited linker, attachment to both protected anchors, and reconnection of the same anchor pair. New bonds between protected atoms are rejected. Furthermore, to avoid false passes in symmetric or repeated structures, the annotated x_0 edit set must be invariant under relevant automorphisms, and at the same time, all valid protected-context mappings must agree on the edit and anchor decomposition.

D.3 Operation and Reference Fidelity

T2 operations are tested only after the grounding check has identified the generated edit region. Linker shortening passes when the shortest protected-anchor path or the editable-region heavy-atom count decreases. Ring removal/opening passes when the specified local ring contribution is removed or opened. Local fragment replacement requires a nontrivial local graph change while maintaining the prescribed anchor connectivity and outside context.

For R1, donor matching is stricter than substructure occurrence. The evaluator checks the complete transfer atom set, atom states, internal bond states, absence of unmodeled donor neighbors, receiver-region containment, contact with all required receiver anchors, boundary topology, and boundary stereochemistry. R2 deliberately omits this tool: the reference is evidence from which a strategy may be inferred, not a motif that the prompt requires the model to copy.

D.4 Descriptor Contract and Reproducibility

All descriptor baselines and evaluation-time values are computed with RDKit 2024.09.4. In particular, hydrogen-bond acceptors use `rdMolDescriptors.CalcNumHBA`; mixing this definition with `CalcNumLipinskiHBA` is disallowed. Property checks compare the generated descriptor with the stored x_0 baseline under the direction stated in the criterion. The release stores the RDKit version and descriptor contract in every group file and in the manifest.

D.5 Release-Level Filtering and Regression Tests

A candidate task is retained only if the hidden feasible molecule x^* passes the complete tool suite assigned to that task. We additionally reject disconnected or ambiguous edit regions, one-atom regions, invalid S2 linkers, non-invariant symmetric mappings, prompt–criterion mismatches, R2 site leakage, trivial or chemically uninformative R1 donor regions, incomplete donor transfer, and inconsistent stereochemistry or boundary bond order. Reference regions are also screened for chemical relevance to the stated task goal.

Before release, all 2,400 hidden targets are rerun through their assigned tools; all pass. We also verify dense indices 0–2,399, unique case identifiers, one-to-one criterion/evaluation ordering, one-dimensional/two-dimensional alignment, canonical output contracts, stored descriptor recomputation, R1/R2 pairing constraints, and absence of hidden-target information in model-facing files. The final selection is stratified by taxonomy cell, biological target, property direction, property count, edit-region size, property identity, and T2 operation subtype.

E Representative Benchmark Cases

Table 18 lists representative tasks sampled from the final release. We report the model-facing instruction type using *No Reference*, R1, or R2 rather than treating no-reference tasks as an R subtype. The hidden x^* structures are shown here only to demonstrate feasibility; they are not available to evaluated models.

E.1 Case A: Property-Driven Substituent Edit without a Reference

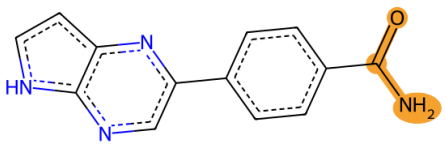
Case 186 is a T1/S1 task for MAPK1. The starting structure is `NC(=O)c1ccc(-c2cnc3[nH]ccc3n2)cc1`; the hidden feasible structure is `Oc1ccc(-c2cnc3[nH]ccc3n2)cc1`. The instruction asks for a change only in the annotated substituent and a decrease in the number of amide bonds. The hidden transformation replaces the terminal carboxamide-like substituent with a hydroxyl group, reducing the RDKit amide-bond count from one to zero while preserving the heteroaromatic parent context.

Table 18: Representative cases from the final release. Indices refer to the dense global benchmark index.

Index	T/S	Reference	Target and pair	Instruction objective	Additional evaluator
186	T1/S1	No Reference	MAPK1: CHEMBL3628252 → CHEMBL3628255	Redesign the annotated substituent and decrease the number of amide bonds.	S1 grounding
695	T1/S2	No Reference	ADRA1A: CHEMBL154023 → CHEMBL155326	Redesign the annotated linker and decrease molecular weight.	S2 grounding
1,267	T2/S1	No Reference	NR3C1: CHEMBL306491 → CHEMBL305417	Replace the local fragment and decrease the number of heteroatoms.	S1 grounding + operation
382	T1/S1	R1	ABL1: CHEMBL13462 → CHEMBL57393	Transfer the explicit phosphonic-acid donor, increase QED, and decrease MolLogP.	Explicit donor graft
582	T1/S1	R2	Same ABL1 transformation as index 382	Infer the useful local design from the unannotated reference under the same property objectives.	No reference-copy criterion
2,193	T2/S2	R1	HRH3: CHEMBL128190 → CHEMBL341491	Transfer the donor, shorten the linker, increase TPSA, and decrease heavy-atom count.	Linker + donor graft
2,215	T2/S2	R2	HDAC1: CHEMBL256440 → CHEMBL255594	Use the reference as an example, replace the local fragment, decrease molecular weight, and increase TPSA.	Linker + operation

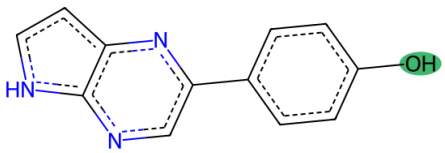
Listing 1 Case 186.

```
Starting molecule with the annotated substituent edit region:
NC(=O)c1ccc(-c2cnc3[nH]ccc3n2)cc1
^^   ^ A
|__ MODIFY THIS REGION
Replace or otherwise redesign this local substructure. Preserve the
    substituent attachment anchor to the parent structure and keep
    structures outside the requested region unchanged. The property
    goal is to decrease the number of amide bonds relative to the
    starting molecule. Return exactly one canonical SMILES in RDKit
    canonical atom order and no other text.
```



X0 | editable region

(a) Model input: annotated x_0



X* | benchmark target

(b) Hidden feasible target x^*

Figure 7: Two-dimensional rendering of case 186. Orange marks the substituent that the model is allowed to edit; green marks the corresponding changed region in the hidden feasible target. The target is shown only for analysis and is not part of the model input.

This case is evaluated by canonical validity, non-identity to x_0, S1 substituent localization, and the amide-bond direction. Passing does not require reproducing x^* exactly; any valid molecule that changes only the marked substituent, preserves its anchor and outside context, and reduces the amide-bond count passes.

E.2 Case B: Operation-Driven Linker Edit without a Reference

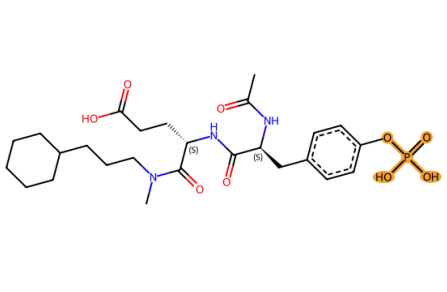
Case 1,862 is a T2/S2 task for CHRM1. Its starting structure is `Nc1ccnc(N2CCc3ccccc3C2)c1`, and the hidden feasible structure is `Nc1cc(N2CCc3ccccc3C2)nc2c1CCC2`. The prompt requests local fragment replacement inside the two-anchor linker region and an increase in the fraction of sp^3 carbons. The hidden transformation raises the fraction from 0.214 to 0.353. This task illustrates that a no-reference T2 instruction still combines three independently evaluated requirements: operation fidelity, S2 localization, and property direction.

E.3 Case C: Paired Explicit and Implicit Reference Guidance

Cases 382 and 582 form a controlled R1/R2 pair for ABL1 (Figure 8). Both use

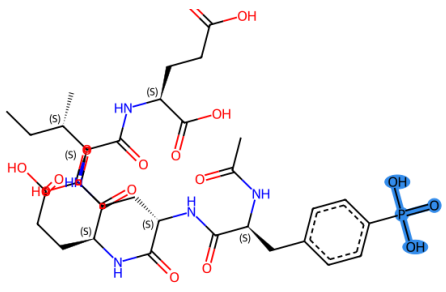
```
x0:   CC(=O)N[C@@H](Cc1ccc(OP(=O)(O)O)cc1)C(=O)N[C@@H](CCC(=O)O)C(=O)N
      (C)CCCC1CCCCC1
xref: CC[C@H](C)[C@H](NC(=O)[C@H](CCC(=O)O)NC(=O)[C@H](CCC(=O)O)NC(=O)
      [C@H](Cc1ccc(P(=O)(O)O)cc1)NC(C)=O)C(=O)N[C@@H](CCC(=O)O)C(=O)O
x*:   CC(=O)N[C@@H](Cc1ccc(P(=O)(O)O)cc1)C(=O)N[C@@H](CCC(=O)O)C(=O)N(
      C)CCCC1CCCCC1
```

The requested edit replaces the aryl phosphate ester linkage with the phosphonic-acid motif found in the reference, creating a direct aryl–phosphorus bond while preserving the peptide-like scaffold and all stereocenters outside the receiver region. The transformation increases QED from 0.198 to 0.217 and decreases MolLogP from 2.374 to 1.705. In the R1 view, the complete phosphonic-acid donor and its attachment boundary are explicitly annotated. In the R2 view, the same reference is supplied without a learning-site annotation, so the model must infer which local motif is relevant. The two tasks otherwise share the starting molecule, reference, hidden feasible target, edit location, and property objectives.



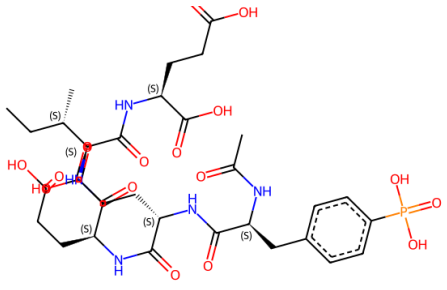
X0 | editable region

(a) Shared annotated starting molecule



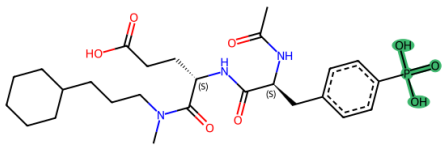
xref | donor region

(b) R1 reference with explicit donor



xref | reference molecule

(c) R2 reference without donor annotation



X* | benchmark target

(d) Shared hidden feasible target

Figure 8: Controlled R1/R2 comparison for cases 382 and 582. R1 exposes the phosphonic-acid donor in blue, whereas R2 presents the same reference without a learning-site annotation. The hidden target shows replacement of the aryl phosphate ester linkage by the directly attached donor motif.

E.4 CASE D: EXPLICIT REFERENCE-GUIDED LINKER SHORTENING

Case 2,193 is a T2/S2/R1 task for HRH3. It starts from `CN1CCN(CCCOc2ccc(CN3CCCCC3)cc2)CC1`, uses `c1cc(OCCN2CCCCC2)ccc1CN1CCCCC1` as the explicit donor reference, and is grounded by the hidden feasible structure `c1cc(OCCCN2CCOCC2)ccc1CN1CCCCC1`. The instruction requires donor transfer, linker shortening, increased TPSA, and decreased heavy-atom count. The evaluator therefore composes six criteria: valid canonical SMILES, non-identity, S2 grounding, linker-shortening fidelity, explicit donor-graft fidelity, and joint property-direction satisfaction. This example demonstrates why target similarity alone is insufficient: a molecule may achieve the property changes yet fail because it modifies the protected aryl or amine context, connects the wrong anchors, or omits part of the donor boundary topology.