

Guiding Large Language Models to Predict Edit Sequences for Molecular Synthesizability Optimization

Corresponding Author: Professor Luhua Lai

This file contains all reviewer reports in order by version, followed by all author rebuttals in order by version.

Version 0:

Reviewer comments:

Reviewer #1

(Remarks to the Author)

Summary

The authors present SynCraft, an LLM-based workflow to perform local editing of unsynthesizable molecules generated by 3D-based generative models to turn them into synthesizable compounds. Unlike projection-based methods or methods that directly generate the full SMILES strings of the synthesizable molecules, SynCraft addresses synthesizability through a pre-defined action space that contains possible moves including adding/deleting/changing atoms and bonds and assigning stereocenters. Through this series of editing, SynCraft demonstrates practical synthesizability fixes through various case studies under typical drug discovery scenarios like pharmacophore retainment and scaffold hopping. Overall, the authors have done a good job at presenting the methods and convincing audiences of the utility of such methods. However, there are some fundamental flaws that need to be addressed before its publication (See details in Weaknesses). The reviewer therefore recommends major revision for this manuscript.

Strength

1. The framing and construction of the synthesis cliff is novel and suggests new ways to restore synthesizability beyond post-hoc filtering and projecting. In the results section, the authors further demonstrate specific case studies of SynCraft success over projection-based methods on scaffold hopping, pharmacophore retainment, and replicating experts' intuition.
2. The creation of the molecular pair dataset for synthesizable-unsynthesizable via a real actionable trajectory of simple molecular editing steps provide value to the field and can be potentially utilized for more LLM training or fine-tuning on other tasks.
3. The interaction-aware prompting strategy is a good attempt to test out LLM's capability to retain interactions while addressing the synthesizability. Previous models mostly care about the docking score retainment of the model, which has less nuances compared to the interaction or pharmacophore retainments that are often more practical and difficult.

Weaknesses

1. The graph matching algorithm to construct the most important reference dataset in this work is not described in great detail. The reviewer will appreciate more clarity on this algorithm.
2. The reaction template set is different between SynCraft and the baseline methods, with SynCraft having significantly more reaction templates to access a larger synthesizable chemical space compared to the baseline approaches. Therefore, this comparison raises concerns about fairness (See Questions 5).
3. Though SynCraft illustrates terrific results in terms of making minimal edits to convert unsynthesizable molecules to synthesizable ones on real world hits. This synthesizability claim is largely unaddressed for the de novo molecules, as there aren't any systematic analysis of the actual synthesizability via score or retro synthesis software to demonstrate the synthesizability of SynCraft-proposed molecules compared with baseline methods. This point is the most major concern the reviewer has regarding this work (Question 6).

4. SynCraft's capability on editing unsynthesizable molecules are largely examined. However, the authors did not give out a real-world recipe on how to use SynCraft to interact with generative designs (See Question 8).

Questions/Suggestions for Revision

1. For the Enamine Building Block vs eMolecule set (Page 6 Line 108), if you perform SimpleRetro on a random subset of let's say 4000 molecules from eMolecules and constrain it on Enamine Building Blocks only, how many will be deemed as synthesizable vs unsynthesizable? Will this result change how you define synthesizable vs unsynthesizable for the synthesis cliff pairs if some of the eMolecules are also treated as "unsynthesizable" under the Enamine building block constraints?

2. For the test set molecules that the authors have constructed to evaluate all benchmarks (Page 7 line 119), what's the distribution of similarity scores for its top-1 and top-5 RAG results? Are the molecules really that similar to be helpful or is it just a cue for the model to refer to when reasoning? A similar study by Wang et al. [1] shows that LLM can perform this few shot synthesis retrieval pretty well even with a random dissimilar template for their RAG. Maybe that's worth investigating a little bit more or at least showing the distribution will help justify the claim of the helpfulness of RAG.

3. For both the graph matching algorithm and the subsequent annotation (Page 8, Line 150), it seems like the generation of these chemical rationales are restrained by the accuracy of the graph matching algorithm. A potential alternative to try for a few samples is to probably give Gemini your actionable space and both the synthesizable and unsynthesizable SMILES strings to ask it to come up with a path themselves and verify it against the results generated by graph matching algorithms. Or if this method demonstrates significant failures, the reviewer will appreciate some comments on why this fails (synthesizable SMILES never reached 80% or the editing usually does not stay within the desired action space?) If the bare labeling works directly, that might give the author a higher-quality dataset if the Gemini can internally come up the same editing path as what the authors suggests, because the reviewer suspect that sometimes maybe the shortest trajectory is not the one that carries the most chemical intuition but the reviewer can certainly be wrong here. All that to say, demonstrating this annotation on ~ 100 or 200 samples will give the reviewer a better sense of whether Gemini's internal reasonability really helps this task. Feel free to use few-shot learning as well if the authors would like, but this will help the reviewer better assess the internal alignment between Gemini's capability on internal coming up with valid routes and Gemini's reasoning capability to navigate away from the unsynthesizable cores.

4. The reason behind the use of 3D based generative datasets is that they already have information about the protein pocket, and thus to some extent, already attempts to have good chemical interactions to keep. Therefore, the step of docking with AutoDock-Vina and then performing PLIP analysis feels confusing (Page 9 Line 176). Why do the authors have to perform the docking step instead of using their native poses as docking may fail to retain the good poses/interactions that the generative algorithms captured?

5. In the Results section for method comparison (Page 11 Table 1), the current results require further clarification as the baseline approaches operate on a much smaller reaction template space. Specifically, ChemProjector used 91 reaction templates, while Synformer and ReaSyn used 115 templates. However, for SynCraft, during the training formulation, the synthesizable molecules are retrieved from the eMolecules set, which has an unknown number of reaction templates. The reviewer suspects that eMolecules might simply have more access to reaction templates that lead to better performances. In addition, Synformer and ReaSyn perform similarly while both are better than ChemProjector, further suggesting that the number of templates might be key factors here. This effect is also discussed in Sun et al. [2], where they tried both reaction template sets and showed the same trend for projection-based methods. Therefore, the reviewer believes the better performance of SynCraft needs to be better justified compared to existing projection-based baselines, specifically from the aspect of reaction templates.

6. In the Results section for method comparison (Page 11 Table 1), since the baselines are all projection-based methods that have the capacity to generate synthetic tree that warrants more synthesizable compounds, it will be crucial for the authors to compare the synthesizability of the compounds generated by projection-based baselines and editing-based SynCraft. SA Score might be acceptable but a retro synthesis analysis using the same procedure will be appreciated (SimpleRetro + USPTO templates) for the final compounds.

7. In the Results section about bioisosteric editing and pharmacophoric integrity (Page 13 Figure 2), it will be helpful for the others to show the original objective scores of ResGen molecules to see if the molecules after editing still preserve similar objectives while having a significant gain in synthesizability. This verification can also further strengthen the claims on pharmacophore retainment and the scaffold hopping arguments, especially when paired with structural analysis.

8. Echoing the last point of weakness, when interacting with generative designs, should one take the generated compound on a specific protein and then directly use SynCraft or something else? A detailed case study from generation to fixing will give the reviewer a better context for the practicality of this model since all test data used in this work are from generative models.

Minor Comments/Questions

1. For the molecular generation part, what is the generation procedure that results in these 4000 molecules for each method? What is the distribution comparison between the synthesizable and unsynthesizable molecules for each model and

is there any meaningful difference between the target problem that they try to solve? For instance, are all 4000 molecules designed against a single pocket, is there any pocket that tends to lead to more generation of unsynthesizable molecules than others?

2. The reviewer expects more comments/information on the poor synthesizability of LIGAN and 3D-SBDD, but this is certainly out of scope for this paper. Maybe the authors could perform an analysis on the property distribution (docking score or QED) of synthesizable vs unsynthesizable molecules to uncover some interesting findings if it's not too much work to do.

3. The reviewer is slightly confused by the fact that the action space only has adding chirality instead of removing stereo centers as chirality is another hard point to address for synthesizability. The reviewer would like to see examples/justifications to prove that this action is indeed useful.

4. If the input from the constructed few-shot learning example pairs only focus on synthesizable and unsynthesizable, how does the information from the PLIP interaction flow into the LLM? The reviewer would assume that the LLM knows to preserve a certain atom but can this preservation also be geometrically aware? For instance, if one molecule has a carbonyl on a 7-membered ring forming H Bond with one Lysine, and there needs to be a structural change of the ring make it synthesizable, will the model know how to maintain the geometry of the new ring in the synthesizable molecule to make sure that carbonyl still points at that Lysine? In other words, could it suggest reducing the 7-membered ring to a 5-membered ring for synthesizability (but might effectively ruin the original orientation to maintain the interaction)?

5. In the interaction-prompt strategy example in SI, the authors show that the hydrophobic interaction is also in the prompt. However, in the Main text, the authors mention to only keep the key interactions like H bonds or pi-stacking. Since PLIP has a pretty soft threshold for hydrophobic interaction and docking might not be ideal to recognize poses with a lot of hydrophobic interaction because it's mostly based on shape complementarity, maybe the hydrophobic constraints can be removed for key interactions?

6. With a better LLM baseline model, will the performance of the results become better? Comparing with at least another LLM model is encouraged, such as the stronger reasoning models like Claude-Opus or Gemini 3.1 Pro as the API for Gemini-Pro-2.5 might become obsolete given that Gemini-Pro-2.0 is depreciated.

References

- [1] Wang, H., et al. "LLM-Augmented Chemical Synthesis and Design Decision Programs." arXiv preprint arXiv:2505.07027, 2025.
[2] Sun, K., et al. "SynLlama: Generating Synthesizable Molecules and Their Analogs with Large Language Models." ACS Cent. Sci. 2025, 11, 2108–2120.

(Remarks on code availability)

Since the reviewer does not have a paid account for Google Gemini, the reviewer used gemini-2.5-flash to test out the workflow and found the workflow easily runnable. However, when using this inferior model, the reviewer found that there were some interesting observations with regard to the reconstructed SMILES as they sometimes contained a main molecule and a fragment. The examples are attached below for one testing SMILES string. This interesting behavior further suggests that the authors should test different LLMs for SynCraft in order to really assess what backbone model could unlock its full potential and capability.

=====

Run 1

```
{
"smiles": "C[C@@H](O)CN(C)C(=O)c1ccc2c(c1)N(C(=O)CCN)[C@H](CCN)C2",
"pass_id": 0,
"status": true,
"reasoning": "The source molecule presents several significant synthetic and metabolic challenges, primarily concentrated within the indoline core and its appended side chains.\n\n1. Multiple Chiral Centers: The molecule possesses two stereocenters, C2 (part of the N-(1-hydroxypropyl) group) and C21 (part of the indoline ring, bearing a propylamine substituent). The presence of multiple stereocenters dramatically increases synthetic complexity, often requiring costly asymmetric synthesis routes or chiral resolution steps.\n\n2. Multiple Primary Amines: There are two primary amine groups (N20 and N24). Primary amines are well-known metabolic \"soft spots,\" susceptible to oxidative deamination, N-acetylation, or other conjugative pathways, which can lead to rapid in vivo clearance and poor pharmacokinetic profiles. They are also highly basic, which can impact solubility, permeability, and potential for off-target interactions.\n\n3. Complex N-Substituent on Indoline: The N15 nitrogen of the indoline is substituted with a complex amide side chain (-C(=O)CH2CH2NH2). This adds further synthetic steps and functional group interconversions (e.g., protecting group chemistry for the primary amine).\n\nThe proposed modification addresses these major liabilities by removing the complex and problematic side chains from the indoline nitrogen (N15) and carbon (C21). Specifically, the amide-linked primary amine chain (-C(=O)CH2CH2NH2) attached to N15 is removed, and the primary amine chain (-CH2CH2NH2) attached to the chiral C21 is also removed.\n\nThis is an intelligent and chemically sound modification for the following reasons:\n\nDramatic Simplification of Synthesis: By removing both the amide side chain from N15 and the amine side chain from C21, we eliminate the need for synthesizing these complex fragments and managing their reactive primary amine functionalities.
```

Crucially, the removal of the C21 side chain also eliminates this stereocenter, reducing the molecule to a single chiral center at C2, or even rendering it achiral if C2 is also simplified (though not done in this step as it's a more core feature). This significantly streamlines the synthetic route and reduces costs.

Enhanced Metabolic Stability: Eliminating the two primary amines (N20 and N24) and the N15-linked amide removes major metabolic liabilities. This is a classic strategy to improve the compound's pharmacokinetic profile, increase half-life, and potentially improve oral bioavailability by blocking common metabolic pathways.

Improved Drug-like Properties: Removing highly polar and basic primary amines can lead to a more balanced lipophilicity and reduced basicity, potentially enhancing membrane permeability and reducing the risk of unwanted off-target interactions.

Preservation of Core Pharmacophore: The core N-methyl-N-(1-hydroxypropyl)benzamide functionality, along with the fundamental indoline scaffold, remains intact. This approach allows for assessment of the core's activity and binding profile, providing a simplified baseline molecule for further SAR exploration, free from the complexities and liabilities of the excised substituents."

"edits": [

[
"DEL_ATOM",
16

],
[
"DEL_ATOM",
17

],
[
"DEL_ATOM",
18

],
[
"DEL_ATOM",
19

],
[
"DEL_ATOM",
20

],
[
"DEL_ATOM",
22

],
[
"DEL_ATOM",
23

],
[
"DEL_ATOM",
24

],
[
"STOP"

]
],

"llm_raw_output": "[REASONING]\n\nThe source molecule presents several significant synthetic and metabolic challenges, primarily concentrated within the indoline core and its appended side chains.\n\n1. **Multiple Chiral Centers:** The molecule possesses two stereocenters, C2 (part of the N-(1-hydroxypropyl) group) and C21 (part of the indoline ring, bearing a propylamine substituent). The presence of multiple stereocenters dramatically increases synthetic complexity, often requiring costly asymmetric synthesis routes or chiral resolution steps.\n\n2. **Multiple Primary Amines:** There are two primary amine groups (N20 and N24). Primary amines are well-known metabolic \"soft spots,\" susceptible to oxidative deamination, N-acetylation, or other conjugative pathways, which can lead to rapid in vivo clearance and poor pharmacokinetic profiles. They are also highly basic, which can impact solubility, permeability, and potential for off-target interactions.\n\n3. **Complex N-Substituent on Indoline:** The N15 nitrogen of the indoline is substituted with a complex amide side chain (-C(=O)CH2CH2NH2). This adds further synthetic steps and functional group interconversions (e.g., protecting group chemistry for the primary amine).\n\n\nThe proposed modification addresses these major liabilities by removing the complex and problematic side chains from the indoline nitrogen (N15) and carbon (C21). Specifically, the amide-linked primary amine chain (-C(=O)CH2CH2NH2) attached to N15 is removed, and the primary amine chain (-CH2CH2NH2) attached to the chiral C21 is also removed.\n\n\nThis is an intelligent and chemically sound modification for the following reasons:\n\n**Dramatic Simplification of Synthesis:** By removing both the amide side chain from N15 and the amine side chain from C21, we eliminate the need for synthesizing these complex fragments and managing their reactive primary amine functionalities. Crucially, the removal of the C21 side chain also eliminates this stereocenter, reducing the molecule to a single chiral center at C2, or even rendering it achiral if C2 is also simplified (though not done in this step as it's a more core feature). This significantly streamlines the synthetic route and reduces costs.\n\n**Enhanced Metabolic Stability:** Eliminating the two primary amines (N20 and N24) and the N15-linked amide removes major metabolic liabilities. This is a classic strategy to improve the compound's pharmacokinetic profile, increase half-life, and potentially improve oral

bioavailability by blocking common metabolic pathways.\n* **Improved Drug-like Properties**: Removing highly polar and basic primary amines can lead to a more balanced lipophilicity and reduced basicity, potentially enhancing membrane permeability and reducing the risk of unwanted off-target interactions.\n* **Preservation of Core Pharmacophore**: The core N-methyl-N-(1-hydroxypropyl)benzamide functionality, along with the fundamental indoline scaffold, remains intact. This approach allows for assessment of the core's activity and binding profile, providing a simplified baseline molecule for further SAR exploration, free from the complexities and liabilities of the excised substituents.\n\n[EDITS]\n```\njson\n{\n \"DEL_ATOM\": 16,\n \"DEL_ATOM\": 17,\n \"DEL_ATOM\": 18,\n \"DEL_ATOM\": 19,\n \"DEL_ATOM\": 20,\n \"DEL_ATOM\": 22,\n \"DEL_ATOM\": 23,\n \"DEL_ATOM\": 24,\n \"STOP\": \"\",\n \"error\": null\n}

Reconstructed SMILES: C.CNc1cccc(C(=O)N(C)CC(C)O)c1
 Input (canonical): C[C@@H](O)CN(C)C(=O)c1ccc2c(c1)N(C(=O)CCN)[C@H](CCN)C2
 =====

Run 2

```
{
  "smiles": "C[C@@H](O)CN(C)C(=O)c1ccc2c(c1)N(C(=O)CCN)[C@H](CCN)C2",
  "pass_id": 0,
  "status": true,
  "reasoning": "The primary synthetic liability in the source molecule is the chiral secondary alcohol group at `[C@@H:2]([OH:3])`. The presence of a stereocenter necessitates either a costly asymmetric synthesis or a challenging chiral resolution step, significantly increasing synthetic complexity and cost. Furthermore, secondary alcohols can be metabolically vulnerable to oxidation by CYP enzymes or susceptible to glucuronidation, potentially leading to rapid in vivo clearance.\n\nTo address this, the proposed modification simplifies this moiety by removing the hydroxyl group and thereby eliminating the chiral center, converting the `CH(OH)` group into an achiral `CH2` group. This transformation replaces the `[CH3:1][C@@H:2]([OH:3])[CH2:4]` segment with a simple, achiral `[CH3:1][CH2:2][CH2:4]` chain.\n\nThis modification is chemically sound and improves synthesizability by:\n1. **Removing chirality**: This is the most significant benefit, transforming a stereoselective synthesis into a much simpler achiral synthesis.\n2. **Improving metabolic stability**: Replacing the potentially labile hydroxyl group with a metabolically inert methylene group can enhance the compound's pharmacokinetic profile.\n3. **Minimal structural change**: The modification maintains the overall carbon chain length and lipophilic character in this region, which is crucial for preserving key binding interactions and overall molecular shape, while simplifying the functional group. This makes it a strategically intelligent simplification that prioritizes synthetic feasibility and metabolic stability without drastically altering the presumed pharmacophoric space.",
  "edits": [
    [
      "DEL_ATOM",
      3
    ],
    [
      "STOP"
    ]
  ],
  "llm_raw_output": "[REASONING]\n\nThe primary synthetic liability in the source molecule is the chiral secondary alcohol group at `[C@@H:2]([OH:3])`. The presence of a stereocenter necessitates either a costly asymmetric synthesis or a challenging chiral resolution step, significantly increasing synthetic complexity and cost. Furthermore, secondary alcohols can be metabolically vulnerable to oxidation by CYP enzymes or susceptible to glucuronidation, potentially leading to rapid in vivo clearance.\n\nTo address this, the proposed modification simplifies this moiety by removing the hydroxyl group and thereby eliminating the chiral center, converting the `CH(OH)` group into an achiral `CH2` group. This transformation replaces the `[CH3:1][C@@H:2]([OH:3])[CH2:4]` segment with a simple, achiral `[CH3:1][CH2:2][CH2:4]` chain.\n\nThis modification is chemically sound and improves synthesizability by:\n1. **Removing chirality**: This is the most significant benefit, transforming a stereoselective synthesis into a much simpler achiral synthesis.\n2. **Improving metabolic stability**: Replacing the potentially labile hydroxyl group with a metabolically inert methylene group can enhance the compound's pharmacokinetic profile.\n3. **Minimal structural change**: The modification maintains the overall carbon chain length and lipophilic character in this region, which is crucial for preserving key binding interactions and overall molecular shape, while simplifying the functional group. This makes it a strategically intelligent simplification that prioritizes synthetic feasibility and metabolic stability without drastically altering the presumed pharmacophoric space.\n\n[EDITS]\n```\njson\n{\n  \"DEL_ATOM\": 3,\n  \"STOP\": \"\",\n  \"error\": null\n}\n}
```

Reconstructed SMILES: CCO.CNC(=O)c1ccc2c(c1)N(C(=O)CCN)C(CCN)C2
 Input (canonical): C[C@@H](O)CN(C)C(=O)c1ccc2c(c1)N(C(=O)CCN)[C@H](CCN)C2

(Remarks on figshare data availability)

The data represents the list of unresolved molecules and the reasoning chains and editing paths from unsynthesizable molecules to the synthesizable ones. They demonstrate more samples besides the few examples in SI of such editing trajectories.

Reviewer #2

(Remarks to the Author)

In this manuscript, Li and Lai introduce SynCraT, a (RAG, few-shot) prompting framework that allows them to leverage the power of LLM to predict edit sequences for molecular synthesizability optimization—that is, to identify modifications of molecules deemed unsynthesizable that make them synthesizable, while preserving desirable structural and/or biological properties.

To the best of my knowledge, the proposed approach is novel. I am particularly impressed by the creative method the authors devise to generate a "reference set" that is then used to train/guide the LLM towards desirable outcomes through RAG and few-shot demonstrations. The effect of each of the steps involved in the method (generation of the reference set, specification of output format, etc.) is well studied in a series of ablation studies reported in the SI. The manuscript is mostly precise, clear and well written, and the reported results for SynCraT improve over the state of the art.

However, I see some major issues with the manuscript:

1) SynCraT requires and relies on a LLM (Gemini-2.5-Pro) to do most of the heavy lifting, and it is unclear to what extent the proposed method is robust to changes in the underlying LLM. For example, even though the GitHub repository allows to use DeepSeek, no results are reported for this LLM or any other alternative. Without such information, it is difficult to understand if the reported results are circumscribed to the current implementation of Gemini-2.5-Pro, and to answer important questions such as whether the method will be robust to future implementations of LLM, or to what extent the current results are due to this particular LLM being particularly suitable to the proposed approach (perhaps because of its specific training in chemistry).

2) For the construction of the reference dataset, "semantic intent ground truth" was generated with a LLM, but there is no explanation in the manuscript about how this "ground truth" was validated. Was it validated by experts or by computational means? How reliable is this "ground truth"? Validation of this training data, and precise description of the validation and its results, is necessary.

3) An aspect of the quantitative validation that is puzzling is the following. Initially, five models were used to generate virtual molecules: LIGAN, 3D-SBDD, Pocket2Mol, DiffSBDD, and ResGen. Then, molecules generated by two of these models (LIGAN and 3D-SBDD) are explicitly excluded from the validation because they "had exceptionally low rates of synthetic accessibility". (But presumably they were not from training? Why?) Another model (DiffSBDD) is not explicitly excluded, but it is never mentioned in the text, and no results for it are shown or described, so are left to guess that this model was also excluded, but we do not know why. The authors should report results for molecules generated by all five models, and let the reader decide what to do with this information.

4) The summary statistics provided in Table 1 mix many different aspects of the performance of the algorithms. What goes wrong when an algorithm (SynCraT, in particular) fails? Does it fail because the proposed molecules are not synthesizable? Or because the edit sets are invalid or lead to invalid molecules? Or because the resulting molecules have a similarity with the original molecule below the desired threshold? All these different statistics should be reported separately, so that the reader can get a more complete picture of the strengths and limitations of the proposed approach vis a vis the baselines.

5) Related to the previous point, it is not entirely clear to me whether the comparison in Table 1 is completely fair, because at least some baseline models operate by projecting onto a known, limited synthesizable chemical space (defined in some way), whereas SynCraT can generate arbitrary new compounds not present in any the chemical spaces in which other methods operate. What would happen if, for example, we enlarged the synthesizable chemical space in SynFormer by allowing more reactions, more building blocks or more steps? Would the performance gap close? In other words, is the improvement of SynCraT with respect to the baselines due to a better capacity to reach synthesizable molecules, or just to the fact that it is allowed to reach molecules that are forbidden to other methods?

6) The qualitative validation on specific molecules (Figs. 2-4 and corresponding sections) is interesting and illuminating, but it is impossible for the reader to evaluate whether these are representative cases as opposed to cherry-picked examples where things work as they should. A systematic evaluation of the claims in this part of the article is, therefore, necessary. For example, how can the reasoning of SynCraT be validated systematically? How often is it correct (as estimated by experts) as opposed to simply plausible? Or what percentage of rescues (over a larger set of validation samples) preserve binding mode, as the example discussed in Fig. 4?

Other minor issues:

7) The LLM is restricted to outputting a JSON-formatted sequence of commands. Is this format always respected? Are the generated sequences always valid and, if not, how often are they invalid and why?

8) The methods section describes how synthesizability was established for the ground truth reference data set, but not for the molecules generated by SynCraT in the validation set of the quantitative evaluation. Was it established the same way (SympRetro for 30 minutes?).

9) What is the meaning of the quotation marks around "synthetic nightmare"? Are these the LLM words? An expert's?

(Remarks on code availability)

I have not run the code, but the GitHub repository seems to be organized and have all necessary ingredients to reproduce the results in the manuscript. An issue that I detected is that the README file mentions a notebooks directory which should contain "Jupyter notebooks for analysis", but no such directory exists.

(Remarks on figshare data availability)

Version 1:

Reviewer comments:

Reviewer #1

(Remarks to the Author)

The authors have made tremendous efforts to improve the manuscript and address different concerns. The reviewer is very happy with the new experiments and results that the authors presented, particularly with the additional discussion around retraining Synformer to enlarge their reaction sets, agentifying the pipeline to offer a real-world usage recipe along with validation on a practical drug target, re-running experiments to verify claims around the similarity-based RAG block of the framework and the graph-matching algorithm, and clarifying the different results sections and comparison details. With these new results, the reviewer has no objection of publishing this work in its current form.

(Remarks on code availability)

(Remarks on figshare data availability)

The data contains necessary details and examples to reproduce the case studies presented in the paper.

Reviewer #2

(Remarks to the Author)

In their revision, the authors have addressed my major concerns from the original submission. In particular, they have addressed convincingly my major concern with regards to the dependency on a single LLM, as well as my request to carry out large-scale quantitative validation of the "golden reasoning corpus." All in all, I think that the work is more convincing now.

A few remaining (relatively minor) issues that the authors may want to take into consideration are:

* The new comparison of LLMs shows large discrepancies between LLMs. In fact, the difference between LLMs is larger than the difference between the original results with Gemini and traditional methods. The authors should probably comment on this fact.

* In their rebuttal, the authors claim that "The remaining three baselines (LiGAN, AR / 3D-SBDD, DiffSBDD) 567 were not used in training either." However, the paper still claims that "From the pool of molecules designated as unsynthesizable, we constructed our 'Synthesis Cliff' dataset, which provides the ground-truth examples for our study. To do this, we first randomly sampled 4,000 unsynthesizable molecules from each of the FIVE generative models." (My emphasis on FIVE.) This is still confusing to me.

* Response Figure 5 refers to some "rows", but the figure only has one row. I did not check if this figure appears in the SI, but, if it does, figure and/or caption probably need to be fixed.

(Remarks on code availability)

(Remarks on figshare data availability)

Open Access This Peer Review File is licensed under a Creative Commons Attribution 4.0 International License, which permits use, sharing, adaptation, distribution and reproduction in any medium or format, as long as you give appropriate credit to the original author(s) and the source, provide a link to the Creative Commons license, and indicate if changes were made.

In cases where reviewers are anonymous, credit should be given to 'Anonymous Referee' and the source.

The images or other third party material in this Peer Review File are included in the article's Creative Commons license, unless indicated otherwise in a credit line to the material. If material is not included in the article's Creative Commons license and your intended use is not permitted by statutory regulation or exceeds the permitted use, you will need to obtain permission directly from the copyright holder.

To view a copy of this license, visit <https://creativecommons.org/licenses/by/4.0/>

Response to Reviewers’ Comments

We thank both reviewers and the editor for their careful and constructive reading. We are particularly grateful for their recognition of the synthesis-cliff framing and the edit-prediction methodology, and for the depth of attention they have given to the details of our protocol. The detailed comments and the suggested experiments have substantially improved the quality and rigour of the manuscript.

The revision below addresses each of the points raised, with the principal additions summarised here and detailed per-reviewer responses following. To minimise cross-document flipping, all figures and tables central to the responses are reproduced inline in this letter; the corresponding sections of the revised main text and Supporting Information (SI) carry the full discussion and methods.

- **Multi-LLM comparison (R1 M6, R2 #1, Editor).** We extended the SynCraft evaluation to three additional LLM backends beyond the headline Gemini-2.5-Pro: GPT-5.2 and the two MIT-licensed open-weight DeepSeek-V4-Pro and DeepSeek-V4-Flash. All four backends exceed the strongest projection-based baseline at Sim>0.5.
- **Template-fairness control (R1 W2/Q5, R2 #5).** We retrained SynFormer on the full USPTO template superset under matched compute. The enlarged-template model does not close the rescue-rate gap to SynCraft on either source pool (SI Section 4), indicating that the gap reflects a paradigm difference rather than template-inventory size.
- **Three additional 3D-generative baselines (R1 W3/Q6, R1 M2, R2 #3).** We ran SynCraft alongside ChemProjector, SynFormer and ReaSyn on heavy-atom-stratified 10% subsets of LiGAN, AR / 3D-SBDD and DiffSBDD under the same SimpRetro+Enamine protocol used in Table 1. SynCraft leads every Sim>0.5 cell by 11–23 percentage points and is first at 11 of the 12 (baseline $\times$ similarity-threshold) cells overall (SI Section 6).
- **Systematic binding-mode retention on two unrelated targets (R2 #6).** We computed PLIP-based interaction retention systematically on the RIPK1 kinase (PDB 7YDX, 42 sources, 191 source–edit pose pairs) and SARS-CoV-2 M^{Pro} (PDB 7L12, 47 sources, 260 pairs). 81% and 79% of source candidates respectively retain $\geq 50\%$ of the source PLIP interactions in at least one rescued edit; on RIPK1 the best edit also improves the Vina docking score on average (mean Δ Vina -0.34 kcal/mol).
- **End-to-end SARS-CoV-2 M^{Pro} case (R1 W4/Q8).** As a worked example of the generation-to-fixing workflow, we packaged SynCraft as an agent skill (MIT-licensed, in the `skill/` subdirectory of the public repository) and applied it to 47 TargetDiff-generated M^{Pro} candidates; under the union of the two skill modes, 32 of the 46 originally unsynthesisable sources were rescued and the best retro-validated edit preserved the source Vina docking score within ± 0.5 kcal/mol for 60% of the rescued pairs (SI Section 5).
- **Edit-path audit, $n=100$ (R1 Q3).** Given source and target, the LLM independently produces edit sequences without the graph-matching scaffold. 97 of 100 reconstructions reach Morgan-r2

similarity ≥ 0.9 to the ground-truth target, confirming that the graph-matching corpus is one valid canonicalisation among many.

- **Similarity-RAG versus random-RAG (R1 Q2).** Replacing similarity retrieval with random-RAG (matched in count and exemplar length) reduces the Gemini-2.5-Pro per-source rescue rate at Sim>0.5 from 42.7% to 22.7% on P2M and from 44.7% to 21.0% on ResGen, essentially back to the zero-shot baseline. An independent LLM-as-judge alignment analysis corroborates this trend. We read both components as necessary: the LLM provides the chemistry baseline, while similarity retrieval roughly doubles the rescue rate over zero-shot, a lift that random retrieval fails to deliver.
- **Golden-corpus reasoning validation (R2 #2).** A four-axis LLM-as-judge protocol scored the full reasoning corpus and a paired negative control. The corpus passes the rubric on the great majority of chains, while the paired negative control passes 0%, demonstrating that the judge discriminates and that the corpus reasoning is chemistry-grounded rather than hallucinated post hoc.
- **eMolecules synthesisability sanity (R1 Q1).** Of 4,000 random eMolecules processed under identical SimpRetro+Enamine constraints, 96.0% are solved, confirming the synthesis-cliff definition is not a protocol artefact.
- **Per-source failure-mode decomposition (R2 #4).** Each source falls into exactly one of {Success, Sim drift, Retro fail, Recon fail}; retro-unsolvability is the dominant non-Success mode across all four LLMs, identifying retrosynthesis coverage as the next-most-impactful improvement direction.
- **Expanded Methods + open data.** Graph-matching pseudocode, the SimpRetro protocol, multi-LLM endpoint configurations, the hard/soft interaction-aware prompt split, and the edit-application sanitisation pipeline are now in the Methods or SI; all data and figure-regeneration scripts are committed in the public repository.

Response to Reviewer 1

Weaknesses

(W1) The graph matching algorithm to construct the most important reference dataset in this work is not described in great detail. The reviewer will appreciate more clarity on this algorithm.

Thank you for highlighting this gap. We have added a dedicated SI section (*Graph-Matching Edit-Path Extraction via Iterative MCS*, Section 1.4) that gives the full algorithm with pseudocode, and the corresponding paragraph in the Methods (*Constructing the Reference Dataset*) now points to it. The procedure is a four-step algorithm: (i) compute the maximum common substructure (MCS) between source and target with bond-type and aromaticity matching; (ii) compute the edit script that maps the MCS-aligned source to target as a sequence of atomic and bond operations from the fixed action vocabulary (ADD_ATOM, DEL_ATOM, MUTATE_ATOM, ADD_BOND, DEL_BOND, CHANGE_BOND, SET_CHIRAL, SET_BOND_STEREO); (iii) execute the edit script and verify canonical-SMILES equality with the target; (iv) iterate on failure by relaxing the MCS thresholds until equality is reached. Pseudocode and the full relaxation schedule are in SI Section 1.4; the implementation is open-sourced at `SynCraft-Core/src/utils.py:IterativeMCSExtractor`.

(W2 / Q5) The reaction template set is different between SynCraft and the baseline methods, with SynCraft having significantly more reaction templates to access a larger synthesizable chemical space compared to the baseline approaches. Therefore, this comparison raises concerns about fairness (See Questions 5).

In the Results section for method comparison (Page 11 Table 1), the current results require further clarification as the baseline approaches operate on a much smaller reaction template space. Specifically, ChemProjector used 91 reaction templates, while Synformer and ReaSyn used 115 templates. However, for SynCraft, during the training formulation, the synthesizable molecules are retrieved from the eMolecules set, which has an unknown number of reaction templates. The reviewer suspects that eMolecules might simply have more access to reaction templates that lead to better performances. In addition, Synformer and ReaSyn perform similarly while both are better than ChemProjector, further suggesting that the number of templates might be key factors here. This effect is also discussed in Sun et al. [2], where they tried both reaction template sets and showed the same trend for projection-based methods. Therefore, the reviewer believes the better performance of SynCraft needs to be better justified compared to existing projection-based baselines, specifically from the aspect of reaction templates.

This is a particularly insightful question, and we thank you for pressing on it. Our prior intuition was that an enlarged template inventory might in fact hurt projection-based methods by expanding the combinatorial search space without proportionally increasing the per-template training signal, but we did not run that empirical control in the original submission. We are glad to have had the chance to validate this intuition during the revision. Specifically, we retrained SynFormer end-to-end on the full USPTO 51,720-template superset under matched compute (architecture and loss unchanged; reaction-classification head expanded from 120-way (115 templates + 5 reserved indices in the published checkpoint) to 51,725-way (51,720 USPTO templates + the same 5 reserved indices)). Training and evaluation details are in SI Section 4. Per-source rescue rates at four similarity thresholds are shown in Response Table 1.

Configuration	Sim > 0.5	Sim > 0.6	Sim > 0.7	Sim > 0.8
P2M-SynFormer (115 templates, published)	30.7%	14.1%	6.2%	2.9%
P2M-SynFormer (51,720 templates, re-trained by us)	5.6%	3.0%	1.2%	0.8%
P2M-SynCraft (Gemini-2.5-Pro)	42.7%	28.4%	11.9%	3.2%
ResGen-SynFormer (115 templates, published)	33.8%	19.9%	8.7%	3.5%
ResGen-SynFormer (51,720 templates, re-trained by us)	6.6%	3.3%	1.5%	0.8%
ResGen-SynCraft (Gemini-2.5-Pro)	44.7%	29.1%	13.1%	3.5%

Response Table 1: SynFormer retrained under matched compute on the full USPTO 51,720-template superset, evaluated on the published P2M and ResGen unsynthesizable test sets used in main-text Table 1.

The enlarged-template SynFormer in fact performs *worse* than the published 115-template baseline on both source pools, consistent with the search-space-explosion intuition. Several mechanisms plausibly contribute, all discussed in more detail in SI Section 4: (i) the long-tail USPTO templates each appear in only tens of training reactions, so the per-template signal is two to three orders of magnitude sparser than for the curated 115-template set; (ii) reaction-building-block coverage in the matched matrix is roughly 15 $\times$ sparser on average; (iii) at each retrosynthesis step the model now picks among $\sim$ 50,000 alternatives, so modest per-step calibration loss compounds along a tree of depth 3–5; and (iv) the 115-template inventory used in the published baseline was hand-curated for Enamine compatibility, which the long-tail USPTO templates largely are not.

The independent reference you kindly pointed to (Sun *et al.* [2], “SynLlama”, ACS Cent. Sci. 2025) is also informative here. Their reported observations are that an LLM-based reconstruction approach can match or exceed SynFormer at identical 115-template inventory (691 vs. 660 of 1,000 ChEMBL molecules under the same Enamine constraint) and that the LLM can extrapolate to building blocks lying outside its training distribution. Our retrained-SynFormer experiment is independent evidence in the same direction: simply enlarging the template inventory does not, by itself, close the rescue-rate gap on this task. Looking beyond their setting, SynCraft is one step further removed from a fixed template-and-building-block search: it performs atom-level edits directly on the source molecular graph, so its outputs need not correspond to any single named reaction template. We do not want to overstate this, but the consistent pattern across both experiments suggests, tentatively, that for the synthesis-cliff task in particular, exploring a larger chemical space through atom-level editing may be a more effective use of an LLM’s chemistry knowledge than scaling a curated reaction-template inventory. We do not claim this generalises beyond the present task setting.

(W3) Though SynCraft illustrates terrific results in terms of making minimal edits to convert unsynthesizable molecules to synthesizable ones on real world hits. This synthesizability claim is largely unaddressed for the de novo molecules, as there aren’t any systematic analysis of the actual synthesizability via score or retro synthesis software to demonstrate the synthesizability of SynCraft-proposed molecules compared with baseline methods. This point is the most major concern the reviewer has regarding this work (Question 6).

We apologise for the lack of clarity here. The original Methods section described SimpRetro+Enamine (USPTO templates with frequency ≥ 2 ; Enamine commercial building blocks; 30-minute CPU budget per molecule) only in the context of *classifying* source molecules as unsynthesizable, and did not state explicitly that the same protocol was then applied to every SynCraft-edited product to decide whether it counted as a successful rescue at a given similarity threshold. In the actual pipeline the SynCraft-edited products were validated under this same SimpRetro+Enamine protocol, but we acknowledge that the original text did not make this clear. We have now added a dedicated SI section (*SimpRetro Validation Protocol*, Section 1.8) that states the procedure explicitly and notes that it is applied identically to source classification and to validation of every SynCraft-edited product reported in Table 1 and in the additional baselines below (R1 Q6 / R2 #3). The Table 1 numbers themselves are unchanged.

(W4) SynCraft’s capability on editing unsynthesizable molecules are largely examined. However, the authors did not give out a real-world recipe on how to use SynCraft to interact with generative designs (See Question 8).

(Q8) Echoing the last point of weakness, when interacting with generative designs, should one take the generated compound on a specific protein and then directly use SynCraft or something else? A detailed case study from generation to fixing will give the reviewer a better context for the practicality of this model since all test data used in this work are from generative models.

We thank you for raising this; demonstrating SynCraft as a single-call component on a fresh generative pipeline is indeed the right way to address the practicality question, and we are grateful for the prompt. To make the framework usable in this way, we have packaged SynCraft as an agent skill (MIT-licensed, in the `skill/` subdirectory of the public repository) that exposes the framework as a single API call. The skill accepts pre-generated SMILES plus a receptor PDB and returns synthesisable edited analogs with reasoning traces and re-docked binding scores.

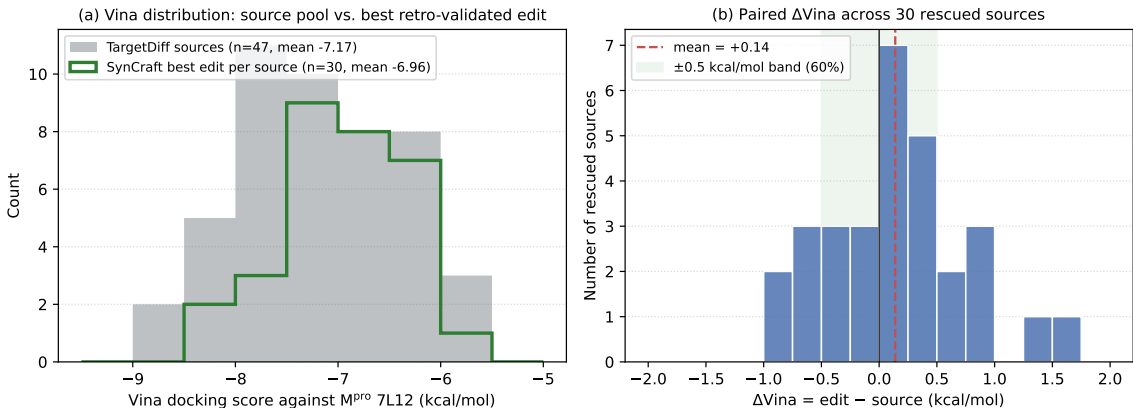
As a worked example of the full generation-to-fixing workflow, we ran an end-to-end case on SARS-CoV-2 M^{pro} (PDB 7L12) using 47 candidate ligands generated by TargetDiff for the 7L12 pocket. Of these, only 1 was directly synthesisable under our SimpRetro protocol. The skill produces 10 parallel edit proposals per mode (rescue-maximising and interaction-preserving; 20 proposals per source in total). Under the union of the two modes, the skill rescued 32 of the 46

originally unsynthesisable candidates (70%); per-mode summary statistics are in Response Table 2.

Metric	Rescue-max mode	Interaction-preserving mode
Per-source rescue rate	63%	37%
Per-product retro-solve rate	23%	16%
Mean Δ SA (best edit – source)	−0.75	−0.77
Mean Δ QED	+0.027	+0.122
Ro5 + Veber pass (best edit)	29%	54%
$\geq 50\%$ PLIP retention (best edit)	—	78%

Response Table 2: Per-mode statistics of the SynCraft agent skill on the 47 actionable TargetDiff M^{pro} candidates (46 requiring rescue). The two modes are complementary: rescue-max produces more synthesisable rescues, interaction-preserving produces fewer but binding-mode-preserving rescues. Under the default ranking that unions the two modes, 32/46 sources (70%) are rescued.

Binding scores are essentially preserved across the rescued products: pairing each rescued source to its best retro-validated edit under the default union ranking ($n = 30$ of the 32 union-rescued sources for which both source and edit poses docked successfully) gives a mean Δ Vina of +0.14 kcal/mol (median +0.17), with 60% of pairs within ± 0.5 kcal/mol of the source and 93% within ± 1.0 kcal/mol, comfortably inside the typical Vina reproducibility envelope (Response Figure 1).



Response Figure 1: Vina docking against the M^{pro} 7L12 pocket. **(a)** Source-pool Vina distribution overlaid with the per-source best retro-validated SynCraft edit (default union ranking). **(b)** Paired Δ Vina histogram across the 30 rescued sources with paired docking data.

Binding-mode preservation in this case is consistent with the systematic RIPK1+M^{pro} PLIP-retention analysis reported in R2 #6 below. The corresponding numbers are reproduced inline here for convenience (Response Table 3):

Metric	RIPK1 (7YDX)	M ^{Pro} (7L12)
Source–edit pose pairs analysed	191	260
Number of source candidates	42	47
Sources with best-edit recall ≥ 0.5	34/42 (81%)	37/47 (79%)
Sources with best-edit recall ≥ 0.7	21/42 (50%)	19/47 (40%)

Response Table 3: Systematic PLIP-based binding-mode retention on the RIPK1 kinase and SARS-CoV-2 M^{Pro}. Recall is computed against the source’s PLIP interaction set; best edit per source is the one with maximum recall. The full table including Vina shifts is in Response Table 13 (R2 #6).

Detailed protocol and per-mode breakdown are in SI Section 5.

Questions/Suggestions for Revision

(Q1) For the Enamine Building Block vs eMolecule set (Page 6 Line 108), if you perform SimpleRetro on a random subset of let’s say 4000 molecules from eMolecules and constrain it on Enamine Building Blocks only, how many will be deemed as synthesizable vs unsynthesizable? Will this result change how you define synthesizable vs unsynthesizable for the synthesis cliff pairs if some of the eMolecules are also treated as “unsynthesizable” under the Enamine building block constraints?

We thank you for the suggested sanity check, and ran exactly this test (SI Section 9). Of 4,000 randomly sampled eMolecules processed under the same SimpRetro 30-minute, Enamine-BB protocol, **96.0%** were solved. eMolecules-class compounds are therefore overwhelmingly synthesisable under our constraints and the synthesis-cliff definition is not a protocol artefact. The 4% unsolved fraction is consistent with eMolecules containing some unusually complex entries that fall just outside the Enamine intersection; these are a small minority and do not change the cliff definition.

(Q2) For the test set molecules that the authors have constructed to evaluate all benchmarks (Page 7 line 119), what’s the distribution of similarity scores for its top-1 and top-5 RAG results? Are the molecules really that similar to be helpful or is it just a cue for the model to refer to when reasoning? A similar study by Wang et al. [3] shows that LLM can perform this few shot synthesis retrieval pretty well even with a random dissimilar template for their RAG. Maybe that’s worth investigating a little bit more or at least showing the distribution will help justify the claim of the helpfulness of RAG.

This is a fair point, and we agree that the contribution of similarity retrieval needs to be quantified rather than asserted. We ran a controlled random-RAG ablation (full details in SI Section 2.4): the same Gemini-2.5-Pro inference pipeline was re-run on the full P2M and ResGen test pools with exemplars drawn uniformly at random from the reference corpus (matched in count and length; random seed fixed), keeping every other aspect of the prompt and the downstream SimpRetro+Enamine validation identical. To check that the exemplars are not merely a generic in-context primer, we additionally scored 300 paired reasoning chains under an independent LLM-as-judge protocol on a four-axis rubric (does the reasoning genuinely use the retrieved exemplars?). Both pieces of evidence are summarised in Response Table 4.

Condition (Gemini-2.5-Pro)	P2M Sim>0.5	ResGen Sim>0.5	Judge alignment (0–5)
Similarity-RAG (5-shot, this work)	42.7%	44.7%	2.28
Random-RAG (5-shot, this work)	22.7%	21.0%	1.76
Zero-shot direct-SMILES (no retrieval)	23.6%	26.4%	—

Response Table 4: Per-source rescue rate at the Sim>0.5 threshold under similarity-RAG vs. random-RAG vs. zero-shot, and the mean LLM-as-judge alignment score (0–5) over 300 paired queries scoring whether the reasoning chain genuinely uses the retrieved exemplars. The zero-shot row is taken from the SI ablation table (Section 2). All runs use Gemini-2.5-Pro under the same downstream SimpRetro+Enamine validation; only the exemplar-retrieval step differs.

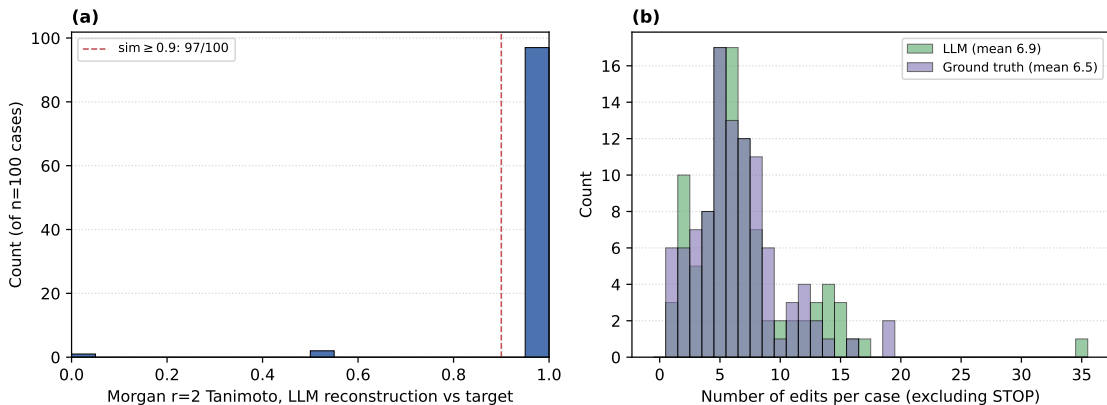
Two observations. First, similarity-RAG outperforms random-RAG on per-source rescue rate (42.7% vs. 22.7% on P2M; 44.7% vs. 21.0% on ResGen), and the independent LLM-as-judge alignment score moves in the same direction (2.28 vs. 1.76, $p = 3.1 \times 10^{-5}$). We note that the similarity-RAG row in this table is taken at the paper-headline pass-k=5 while random-RAG was evaluated at pass-k=1; at matched pass-k=1 the similarity-RAG row of the SI ablation table (P2M 38.5% / ResGen 44.7%) still gives a clear advantage over random-RAG, though smaller than the headline-vs-pass-1 comparison would suggest. For the synthesis-cliff task, retrieval quality therefore does matter in practice: the LLM extracts chemistry-specific information when given a similar exemplar that it does not extract when given a chemically unrelated one. We are grateful for the Wang *et al.* [3] pointer, which we now cite in the relevant SI paragraph; in their multi-step retrosynthesis-planning setting random exemplars already serve as a useful generic in-context primer, while on our atom-level synthesis-cliff task the teaching signal an exemplar carries is the specific edit that converts an unsynthesizable scaffold into a near-neighbour synthesizable one, which a randomly drawn exemplar does not demonstrate.

Second, random-RAG performance is essentially at the zero-shot baseline (P2M 22.7% vs. 23.6%; ResGen 21.0% vs. 26.4%, the latter possibly because mismatched exemplars can in some cases act as mild distractors). We read both components as necessary rather than one being dominant: the LLM provides the chemistry baseline (without it, no synthesizable analog is found at all), while similarity retrieval provides the lift that distinguishes the deployed pipeline from a zero-shot prompt: on P2M the per-source rescue rate roughly doubles from the zero-shot baseline (23%) to 38.5% at matched pass-k = 1, and to 42.7% at the published pass-k = 5. A chemically unrelated exemplar fails to deliver this lift, suggesting that the teaching signal an exemplar carries is the specific atom-level edit converting an unsynthesizable scaffold into a near-neighbour synthesizable one rather than a generic "edit sequences look like this" template.

(Q3) For both the graph matching algorithm and the subsequent annotation (Page 8, Line 150), it seems like the generation of these chemical rationales are restrained by the accuracy of the graph matching algorithm. A potential alternative to try for a few samples is to probably give Gemini your actionable space and both the synthesizable and unsynthesizable SMILES strings to ask it to come up with a path themselves and verify it against the results generated by graph matching algorithms. Or if this method demonstrates significant failures, the reviewer will appreciate some comments on why this fails (synthesizable SMILES never reached 80% or the editing usually does not stay within the desired action space?) If the bare labeling works directly, that might give the author a higher-quality dataset if the Gemini can internally comes up the same editing path as what the authors suggests, because the reviewer suspect that sometimes maybe the shortest trajectory is not the one that carries the most chemical intuition but the reviewer can certainly be wrong here. All that to say, demonstrating this annotation on ~100 or 200 samples will give the reviewer a better sense of whether Gemini’s internal reasonability really helps this task. Feel free to use few-shot learning as well if the authors would like, but this will help the reviewer better assess the internal alignment between Gemini’s

235 capability on internal coming up with valid routes and Gemini’s reasoning capability to navigate away from
 236 the unsynthesizable cores.

237 We thank you for the well-posed suggested control, and we agree that it is the right experiment
 238 to run. We performed the audit on $n = 100$ randomly sampled synthesis-cliff pairs (SI Section 2.5).
 239 We used DeepSeek-V4-Pro rather than Gemini-2.5-Pro for this audit because the synthesis-cliff
 240 rationale corpus was itself generated post-hoc by Gemini-2.5-Pro, and using the same model to
 241 score whether an LLM can independently re-derive the edit sequence would risk self-reference bias;
 242 the same rationale underlies our choice of judge model in the golden-corpus validation (R2 #2).
 243 DeepSeek-V4-Pro was prompted with both source and target SMILES plus our action vocabulary,
 244 and asked to produce an edit sequence independently of the graph-matching pipeline. The LLM
 245 produced parseable JSON in all 100 cases; 99 of 100 reconstructed to a valid product graph and
 246 1 failed reconstruction (counted as a failure with similarity 0 for this audit). The reconstructed
 247 product reached Morgan-r2 Tanimoto ≥ 0.9 to the ground-truth target in 97 of 100 cases, with
 248 mean similarity 0.980 and median 1.000. Only 19% of the LLM-produced edit lists exactly match
 249 the graph-matching annotation; the LLM edit-count (mean 6.9) is very close to the graph-matching
 250 count (mean 6.5), so the model is finding different but chemically equivalent edit paths to the same
 251 product rather than longer or shorter routes.



Response Figure 2: Edit-path audit on $n = 100$ synthesis-cliff pairs. (a) Morgan-r2 Tanimoto similarity between the LLM-direct edit reconstruction and the graph-matching ground-truth target. (b) LLM-direct edit count versus graph-matching edit count for the same pairs.

252 We agree with the hypothesis raised here that the shortest graph-matching trajectory is not
 253 necessarily the one with the strongest chemical intuition. Our reading of this audit is that the
 254 graph-matching path is one valid canonicalisation of the source–target transformation rather than
 255 a unique correctness criterion, and that modern LLMs already possess the chemistry knowledge to
 256 derive equivalent rescue edits when given the endpoint pair.

257 We would also like to add one methodological clarification that we hope is useful here. The
 258 synthesis-cliff corpus is used as a retrieval-augmented exemplar pool at inference time, not as
 259 gradient training data: SynCraft does not fine-tune the LLM on this corpus. At inference, the LLM
 260 is shown the source query and the retrieved exemplars and generates its own atom-map indices on
 261 the source from scratch; the specific atom-map sequence in any exemplar’s edit list is therefore a
 262 format example rather than something the LLM copies verbatim. What actually transfers from the
 263 exemplars is the reasoning style (chemistry liability $\rightarrow$ proposed fix), the operation vocabulary, and
 264 the JSON output format. The first of these, the reasoning style, is what the LLM-as-judge protocol

under R2 #2 systematically validates (82.5% pass- $\geq$ 3-of-4 on the four-axis rubric); the latter two are identical between graph-matching and LLM-derived paths. We did not run a separate corpus-replacement experiment in which the graph-matching paths are swapped out for LLM-derived paths and the rescue rate is re-measured. Our reading is that such a swap would primarily affect the surface form of the example edits rather than the substantive signal the LLM extracts at inference time, but we acknowledge this is an empirical conjecture we did not test, and we are grateful to you for pointing toward it as a future direction.

(Q4) The reason behind the use of 3D based generative datasets is that they already have information about the protein pocket, and thus to some extent, already attempts to have good chemical interactions to keep. Therefore, the step of docking with AutoDock-Vina and then performing PLIP analysis feels confusing (Page 9 Line 176). Why do the authors have to perform the docking step instead of using their native poses as docking may fail to retain the good poses/interactions that the generative algorithms captured?

This is a fair observation, and we share the concern that Vina re-docking can in principle disturb a good native pose. Two considerations led us to use Vina re-docking nevertheless, with one additional practical note.

(i) *Methodological symmetry.* The SynCraft-edited products do not have a native pose; only the source candidates do. Comparing a native source pose to a docked edit pose would mix two pose-generation procedures within a single source–edit pair, which we judged worse than re-docking both ends under identical conditions. The PLIP-retention analysis (R2 #6) and the M^{pro} skill case (R1 W4/Q8) both rely on this paired comparison.

(ii) *Pose quality for PLIP analysis.* The native poses produced by the 3D-aware generators in our source pool are sampled jointly with structure generation and are not energy-minimised against the receptor; they routinely have minor clashes, missing or non-standard hydrogen placements, and higher torsion energies than a brief pose search resolves. PLIP is sensitive to such pose noise. Vina with best-pose selection yields a reproducible, energy-relaxed conformation under one consistent protocol.

We agree this trades off generative-model intent against pose-quality consistency, and we make no claim that native poses are wrong. We use Vina re-docks for symmetric source/edit comparison and pose-quality consistency for PLIP, and the systematic retention analysis (R2 #6) is reported under exactly the same protocol on both sides of every source–edit pair.

(Q5) In the Results section for method comparison (Page 11 Table 1), the current results require further clarification as the baseline approaches operate on a much smaller reaction template space. Specifically, ChemProjector used 91 reaction templates, while Synformer and ReaSyn used 115 templates. However, for SynCraft, during the training formulation, the synthesizable molecules are retrieved from the eMolecules set, which has an unknown number of reaction templates. The reviewer suspects that eMolecules might simply have more access to reaction templates that lead to better performances. In addition, Synformer and ReaSyn perform similarly while both are better than ChemProjector, further suggesting that the number of templates might be key factors here. This effect is also discussed in Sun et al. [2], where they tried both reaction template sets and showed the same trend for projection-based methods. Therefore, the reviewer believes the better performance of SynCraft needs to be better justified compared to existing projection-based baselines, specifically from the aspect of reaction templates.

This question is answered together with W2 above; the direct empirical control is the retraining of SynFormer on the full USPTO 51,720-template superset under matched compute (Response Table 1). Briefly, the enlarged-template SynFormer performs *worse* than the published 115-template baseline on both source pools (P2M Sim $>$ 0.5 drops from 30.7% to 5.6%; ResGen from 33.8% to 6.6%), while SynCraft (Gemini-2.5-Pro) reaches 42.7% and 44.7% respectively. Scaling reaction-template inventory alone therefore does not close the gap.

(Q6) In the Results section for method comparison (Page 11 Table 1), since the baselines are all

projection-based methods that have the capacity to generate synthetic tree that warrants more synthesizable compounds, it will be crucial for the authors to compare the synthesizability of the compounds generated by projection-based baselines and editing-based SynCraft. SA Score might be acceptable but a retro synthesis analysis using the same procedure will be appreciated (SimpleRetro + USPTO templates) for the final compounds.

We thank you for raising this. A direct retrosynthesis comparison of the final SynCraft and projection-baseline outputs is indeed the right way to substantiate the synthesizability claim, and we apologise that the original Methods text did not make it clear that this is what Table 1 already measures. Every method evaluated in Table 1, including SynCraft and the three projection baselines (ChemProjector, SynFormer, ReaSyn), has every one of its proposed analogs validated under the same SimpRetro+Enamine protocol used to classify the source molecules as unsynthesizable in the first place. A molecule is counted at a given similarity threshold only if it passes this retrosynthesis check, so the per-source rescue numbers in Table 1 are SimpRetro pass-rates by definition. As we noted under W3, the original Methods text did not state this explicitly, and we have now added a dedicated SI section (*SimpRetro Validation Protocol*, Section 1.8) that lays out the procedure once for all methods. The R2 #3 response below extends the same SimpRetro-validated comparison to the three additional 3D-generative source distributions that were omitted from the original Table 1.

(Q7) In the Results section about bioisosteric editing and pharmacophoric integrity (Page 13 Figure 2), it will be helpful for the others to show the original objective scores of ResGen molecules to see if the molecules after editing still preserve similar objectives while having a significant gain in synthesizability. This verification can also further strengthen the claims on pharmacophore retainment and the scaffold hopping arguments, especially when paired with structural analysis.

Thank you for this suggestion. Showing the pre/post objective scores alongside the synthesizability gain is indeed the natural way to demonstrate that the rescue does not silently degrade the molecule’s drug-likeness, and we are grateful for the prompt to be more explicit about this. We have added aggregate pre/post drug-likeness deltas across the full P2M and ResGen rescue corpora in Response Table 5 (also in SI Section 2.6). SA score moves favourably (lower) and QED moves favourably (higher) on both source pools.

Source pool	Metric	Source (mean)	SynCraft best edit (mean)
Pocket2Mol (n=1,725)	SA	3.62	3.41 (−0.21)
	QED	0.554	0.583 (+0.029)
ResGen (n=1,025)	SA	3.51	3.30 (−0.21)
	QED	0.560	0.587 (+0.027)

Response Table 5: Aggregate pre/post drug-likeness deltas across the full P2M and ResGen SynCraft rescue corpora. Best-of-K per source; reported as mean of per-source values.

A full per-source docking comparison of the ResGen rescue pool is something we would have liked to provide, and we want to be transparent about why we did not. The 1,025 ResGen-generated sources span all 94 CrossDocked test pockets, so a strict pool-wide objective-score comparison would require preparing 94 separate Vina docking pipelines (pocket selection, protonation, water removal, search-box definition, exhaustiveness calibration) and re-docking every source and every SynCraft-edited product into each. We judged this beyond the scope of the SI audit and chose instead to invest the docking budget in two fully prepared receptor systems on which a like-for-like source–edit Vina comparison is meaningful: the RIPK1 kinase (PDB 7YDX, 42 sources, 191 paired edit poses) and SARS-CoV-2 M^{Pro} (PDB 7L12, 47 sources, 260 pairs). On the RIPK1 docking-validated subset, the best edit meets or exceeds the source Vina score in 74% of cases and is within 0.5 kcal/mol

in 88%; the mean ΔV_{ina} is -0.34 kcal/mol, indicating that on this kinase pool SynCraft slightly improves binding on average while restoring synthesizability. PLIP-level interaction retention on both targets is reproduced inline under R2 #6 below. These two analyses cover structurally and mechanistically unrelated protein families (serine/threonine kinase vs. cysteine protease), and are intended as a systematic answer to the binding-mode question that the pool-wide ResGen docking would otherwise have provided.

(Q8) Echoing the last point of weakness, when interacting with generative designs, should one take the generated compound on a specific protein and then directly use SynCraft or something else? A detailed case study from generation to fixing will give the reviewer a better context for the practicality of this model since all test data used in this work are from generative models.

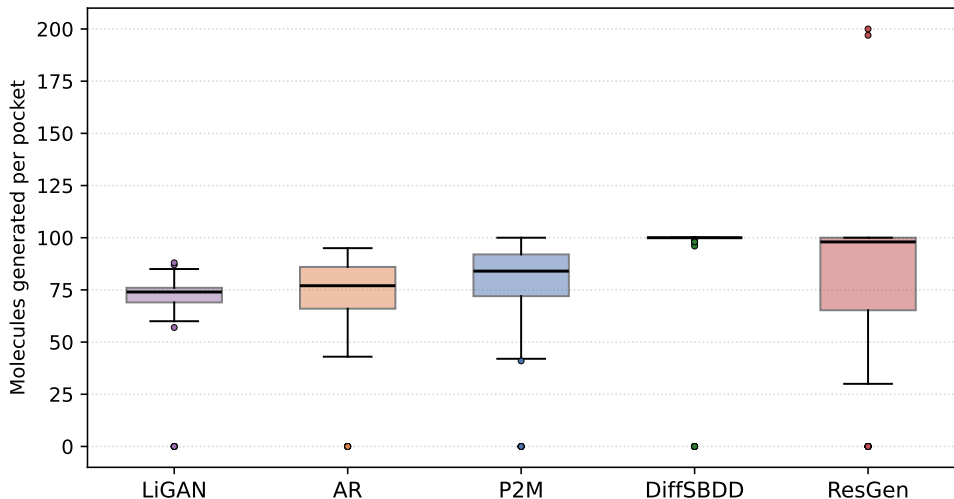
This is answered jointly with W4 above. The end-to-end M^{pro} case (47 TargetDiff-generated candidates with PDB 7L12 as the receptor, run through the SynCraft agent skill, with 32 of the 46 originally unsynthesizable sources rescued) is the worked generation-to-fixing example requested. The corresponding per-mode statistics are in Response Table 2 and the source–edit Vina comparison in Response Figure 1 above. We provide the skill in the `skill1/` subdirectory of the public repository (MIT-licensed) so that practitioners can apply this same workflow to their own generative outputs without writing custom orchestration code.

Minor Comments

(M1) For the molecular generation part, what is the generation procedure that results in these 4000 molecules for each method? What is the distribution comparison between the synthesizable and unsynthesizable molecules for each model and is there any meaningful difference between the target problem that they try to solve? For instance, are all 4000 molecules designed against a single pocket, is there any pocket that tends to lead to more generation of unsynthesizable molecules than others?

Apologies for the lack of clarity in the original submission. The candidate molecules for LiGAN, 3D-SBDD (AR), Pocket2Mol, DiffSBDD and ResGen are taken directly from the GenBench3D benchmark of Baillif *et al.* [1], who generated them across the 94 CrossDocked2020 test pockets using each model’s released checkpoint under its default settings (we refer you to their Section “Compound generation” for the per-model sampling protocol). We did not run any of the five 3D-aware generative models ourselves; our work uses their pre-generated outputs unchanged. GenBench3D also includes TargetDiff, but their public release did not provide the corresponding generated SMILES, so we could not include it in the Table 1 benchmark; we did run TargetDiff ourselves later in the revision, only for the end-to-end agent-skill demonstration on the SARS-CoV-2 M^{pro} pocket (R1 W4/Q8 above), which uses a pocket-specific generation rather than the full CrossDocked benchmark.

Working from the GenBench3D outputs, the per-pocket molecule count is distributed across the 94 CrossDocked test pockets as shown in Response Figure 3. The per-pocket unsynthesizable fraction also varies markedly within each generator (for example P2M ranges from 3% to 64% and ResGen from 0% to 61%), indicating that some binding sites systematically push the generators toward strained or chemistry-poor substructures (SI Section 1.1 carries the per-pocket fraction distribution; we omit it here to focus on the count distribution requested). SynCraft’s rescue performance is reported at the source-pool level rather than per-pocket because the cliff-pair retrieval mechanism is pocket-agnostic, but the per-pocket information is provided in SI for readers who wish to inspect pocket-by-pocket variation.



Response Figure 3: Per-pocket count of generated molecules across the 94 CrossDocked2020 test pockets, for the five 3D-aware generative baselines used as source pools in Table 1 (data from GenBench3D [1]). Box shows the interquartile range, whiskers extend to $1.5 \times \text{IQR}$, dots are pocket-level outliers.

(M2) The reviewer expects more comments/information on the poor synthesizability of LiGAN and 3D-SBDD, but this is certainly out of scope for this paper. Maybe the authors could perform an analysis on the property distribution (docking score or QED) of synthesizable vs unsynthesizable molecules to uncover some interesting findings if it's not too much work to do.

Thank you for the suggestion. We added a chemistry-quality profile of the unsynthesizable subsets across the five baselines (Response Table 6; SI Section 1.2 and SI Figure 2). LiGAN, AR / 3D-SBDD and DiffSBDD all generate from chemistry-agnostic 3D voxel or atom-type priors and produce molecules with intrinsically harder synthesis: their SA-score distributions are shifted right by 1.2–1.5 units versus P2M, and the any-artefact and compounded-artefact rates are substantially higher (chiefly for LiGAN).

Source	n	Mean SA	SA shift vs P2M	Any artifact	Compounded
P2M	1,725	3.62	—	12.1%	0.5%
ResGen	1,025	3.51	−0.11	13.0%	0.9%
DiffSBDD	383	5.14	+1.52	22.7%	1.3%
AR	365	5.04	+1.42	17.0%	1.4%
LiGAN	372	4.90	+1.28	47.8%	8.6%

Response Table 6: Chemistry-quality profile of the unsynthesizable subsets across the five 3D-generative baselines. “Any artefact” = at least one of cumulene, peroxide, thiol, geminal-diol, geminal-diol/amine, vinyl-ether–hydroxyl or imine–hydroxyl substructure; “Compounded” = two or more such patterns in one molecule.

SynCraft editing recovers most of this source-quality gap: the per-source rescue rates on the three additional baselines under matched SimpRetro+Enamine validation are shown in Response Table 7.

Source pool	Method	Sim>0.5	Sim>0.6	Sim>0.7	Sim>0.8
LiGAN (n=372)	ChemProjector	1.9%	0.8%	0.0%	0.0%
	SynFormer (115 templates)	11.6%	6.2%	3.8%	1.1%
	ReaSyn	14.0%	9.1%	5.9%	2.7%
	SynCraft (DSv4-Pro)	25.5%	17.5%	6.7%	1.3%
AR / 3D-SBDD (n=365)	ChemProjector	10.1%	5.2%	1.9%	0.8%
	SynFormer (115 templates)	7.7%	3.8%	1.9%	0.3%
	ReaSyn	14.5%	6.3%	3.0%	1.6%
	SynCraft (DSv4-Pro)	26.3%	17.8%	8.2%	1.9%
DiffSBDD (n=383)	ChemProjector	6.5%	2.6%	0.5%	0.0%
	SynFormer (115 templates)	6.8%	2.6%	1.6%	0.8%
	ReaSyn	25.6%	10.7%	3.9%	1.6%
	SynCraft (DSv4-Pro)	38.1%	26.1%	14.1%	3.7%

Response Table 7: Four-method per-source rescue rate on heavy-atom-stratified 10% subsets of LiGAN, AR / 3D-SBDD and DiffSBDD, under the same SimpRetro+Enamine protocol used in main-text Table 1. SynCraft uses DeepSeek-V4-Pro at pass-k=1; projection-based methods use their default published settings.

SynCraft leads every Sim>0.5 cell by 11–23 percentage points and is first at 11 of the 12 (baseline $\times$ similarity-threshold) cells in this table; the one exception is LiGAN at Sim>0.8 (1.3% vs. ReaSyn 2.7%), a threshold at which the four-method denominator is in single digits and the ranking is correspondingly noise-dominated. The editing paradigm therefore degrades much less under source-quality variation than projection methods do; the same table is also reproduced under R2 #3 for reference.

(M3) The reviewer is slightly confused by the fact that the action space only has adding chirality instead of removing stereo centers as chirality is another hard point to address for synthesizability. The reviewer would like to see examples/justifications to prove that this action is indeed useful.

Apologies for the confusion: the action space does support both setting and clearing stereo-chemistry. SET_CHIRAL accepts CW, CCW, and UNSPECIFIED (the last clears an existing stereocentre); analogously, SET_BOND_STEREO accepts E, Z, and UNSPECIFIED. This makes “de-chiralisation” a first-class edit operation. The PLK1 case in main-text Figure 3 is exactly one such example: the LLM identifies the 2,5-dimethylpiperazine stereocentres as the principal synthesizability bottleneck and removes them by a pair of DEL_ATOM edits on the two methyl substituents, eliminating the four-stereoisomer mixture and converging on the achiral piperazine core that was actually adopted by the human chemists who synthesised the corresponding analog. The SI action-vocabulary listing (SI Section 1, “Definition of the Editing Action Space”) and the system-prompt definition have been updated to spell out all valid UNSPECIFIED tags; this was a documentation gap in the original submission, not a missing capability.

(M4) If the input from the constructed few-shot learning example pairs only focus on synthesizable and unsynthesizable, how does the information from the PLIP interaction flow into the LLM? The reviewer would assume that the LLM knows to preserve a certain atom but can this preservation also be geometrically aware? For instance, if one molecule has a carbonyl on a 7-membered ring forming H Bond with one Lysine, and there needs to be a structural change of the ring make it synthesizable, will the model know how to maintain the geometry of the new ring in the synthesizable molecule to make sure that carbonyl still points at that Lysine? In other words, could it suggest reducing the 7-membered ring to a 5-membered ring for synthesizability (but might effectively ruin the original orientation to maintain the interaction)?

Interaction-aware prompting injects two additional fields into the LLM prompt at inference time: (i) a textual description of the source’s docked PLIP interactions (atom map number, residue identifier, interaction type) and (ii) an instruction directing the LLM to preserve those interactions when proposing edits. The mechanism is described in the revised Methods (“Interaction-Aware Prompting Strategy”) and an example prompt is included in SI Section 1.9. The geometric concern raised here is, however, correct in principle: the LLM does not see explicit 3D coordinates, so it cannot directly reason about ring geometry, and the carbonyl-on-7-ring scenario described in the question is exactly the kind of failure mode that purely textual interaction tags do not prevent. Our defence is therefore empirical rather than constructive: we re-dock and re-PLIP every proposed edit, and a large-N PLIP-retention analysis on two unrelated receptors measures how often this textual interface preserves the source interaction set in practice (Response Table 8).

Metric	RIPK1 (7YDX)	M ^{Pro} (7L12)
Source–edit pose pairs analysed	191	260
Number of source candidates	42	47
Mean per-pair recall vs source	0.47	0.42
Sources with best-edit recall ≥ 0.5	34/42 (81%)	37/47 (79%)
Sources with best-edit recall ≥ 0.7	21/42 (50%)	19/47 (40%)
Sources with best-edit recall = 1.0	11/42 (26%)	2/47 (4%)

Response Table 8: Systematic PLIP-based binding-mode retention on the RIPK1 kinase (PDB 7YDX) and SARS-CoV-2 M^{Pro} (PDB 7L12). Recall is computed against the source’s PLIP interaction set; best edit per source is the one with maximum recall. The full table including Vina shifts is in Response Table 13 (R2 #6).

On both targets the majority of source candidates have at least one edit retaining $\geq 50\%$ of the source interactions. The geometric-orientation failure mode noted above is exactly what drives the long lower tail of these distributions (i.e. the 19%/21% of sources that fail the ≥ 0.5 recall criterion on RIPK1/M^{Pro} respectively), and we agree it is a real limitation of the purely textual interaction interface; we now flag it explicitly in the Discussion.

(M5) In the interaction-prompt strategy example in SI, the authors show that the hydrophobic interaction is also in the prompt. However, in the Main text, the authors mention to only keep the key interactions like H bonds or pi-stacking. Since PLIP has a pretty soft threshold for hydrophobic interaction and docking might not be ideal to recognize poses with a lot of hydrophobic interaction because it’s mostly based on shape complementarity, maybe the hydrophobic constraints can be removed for key interactions?

We agree on both counts: PLIP hydrophobic contacts use a soft 4-Å cut-off and are appreciably noisier than directional contacts, and Vina’s shape-complementarity scoring is not ideally suited to discriminate between near-isomers that differ only in hydrophobic packing. We have restructured the interaction-aware prompt to split the injected contacts into two blocks: **CRITICAL BIOLOGICAL CONSTRAINTS [PRESERVE; high priority]** for H-bonds, salt bridges, π -stacking and halogen bonds, and **PREFERRED CONTACTS [maintain where reasonable; soft preference]** for hydrophobic contacts (SI Section 1.9). The systematic PLIP-retention metrics in SI Section 7 likewise weight directional interactions more heavily in their construction.

(M6) With a better LLM baseline model, will the performance of the results become better? Comparing with at least another LLM model is encouraged, such as the stronger reasoning models like Claude-Opus or Gemini 3.1 Pro as the API for Gemini-Pro-2.5 might become obsolete given that Gemini-Pro-2.0 is depreciated.

We agree, and the revised manuscript reports a four-LLM comparison in main-text Table 1 covering Gemini-2.5-Pro (paper headline), GPT-5.2, DeepSeek-V4-Pro and DeepSeek-V4-Flash (the last two being MIT-licensed open weights). The four backends span roughly a fourteen-month release window (Gemini-2.5-Pro June 2025; GPT-5.2 December 2025; DSv4-Pro / DSv4-Flash April 2026), and the rescue-rate ranking does track that window: the more recently released DeepSeek-V4 backends both exceed the Gemini-2.5-Pro headline at Sim>0.5 on both source pools, which is the anticipated outcome given the release-date gap. GPT-5.2 was evaluated at pass-k=1 throughout due to API budget, and is comparable to Gemini-2.5-Pro on ResGen and slightly behind on P2M. All four backends exceed the strongest projection baseline by a substantial margin on both pools, supporting the conclusion that the SynCraft framework generalises across reasoning LLMs and is expected to continue benefiting from improvements in the underlying reasoning LLMs over time without requiring framework-side changes.

R1 Remarks on Code

Since the reviewer does not have a paid account for Google Gemini, the reviewer used gemini-2.5-flash to test out the workflow and found the workflow easily runnable. However, when using this inferior model, the reviewer found that there were some interesting observations with regard to the reconstructed SMILES as they sometimes contained a main molecule and a fragment (e.g. C.CNc1cccc(C(=O)N(C)CC(C)O)c1 and CCO.CNC(=O)c1ccc2c(c1)N(C(=O)CCN)C(CCN)C2). This interesting behavior further suggests that the authors should test different LLMs for SynCraft in order to really assess what backbone model could unlock its full potential and capability.

Thank you for sharing the concrete failure traces. The fragment-plus-main-molecule output reflects edits whose reconstruction disconnects the molecular graph (typically a DEL_ATOM on an in-ring atom without subsequent bond-merging, leaving an orphan substituent). Our pipeline applies an explicit post-reconstruction sanitisation step that flags such outputs (`was_fragmented = True`) and excludes them from the retro-validation queue, so they do not contaminate the headline rescue numbers. This step was implemented in `reconstruct_from_sequence` but was not explicitly documented; we have now added a paragraph in Methods (*Edit application and sanitisation*) and a tutorial cell in the released notebook. The four-LLM comparison (R1 M6 above) further shows that the disconnected-fragment behaviour drops substantially with the stronger backbones: in the per-source failure-mode decomposition (R2 #4), Recon-fail is 11% for Gemini-2.5-Pro but < 1% for the two DeepSeek-V4 backends.

Response to Reviewer 2

Major Issues

(#1) SynCraft requires and relies on a LLM (Gemini-2.5-Pro) to do most of the heavy lifting, and it is unclear to what extent the proposed method is robust to changes in the underlying LLM. For example, even though the GitHub repository allows to use DeepSeek, no results are reported for this LLM or any other alternative. Without such information, it is difficult to understand if the reported results are circumscribed to the current implementation of Gemini-2.5-Pro, and to answer important questions such as whether the method will be robust to future implementations of LLM, or to what extent the current results are due to this particular LLM being particularly suitable to the proposed approach (perhaps because of its specific training in chemistry).

This is an important point, and we agree that the robustness of the framework to changes in the underlying LLM is a substantive question that ties directly to the editor’s open-weight reproducibility ask. We have therefore extended the SynCraft evaluation to a four-LLM panel in the revised Table 1. In addition to the paper-headline Gemini-2.5-Pro (proprietary, accessed via API), we ran the identical SynCraft pipeline (same system prompt, same 5-shot RAG retrieval, same downstream SimpRetro+Enamine validation) on three additional reasoning LLMs spanning roughly a fourteen-month release window: GPT-5.2 (proprietary, released 2025-12-11) and the two MIT-licensed open-weight DeepSeek-V4-Pro and DeepSeek-V4-Flash (open-weight release 2026-04-24). The two DeepSeek backends can be self-hosted locally via vLLM or SGLang, so the pipeline can be reproduced indefinitely without any commercial-API dependency.

All four backends successfully execute the edit-prediction protocol. Per-source rescue rates at the Sim>0.5 threshold range from 39% to 65% on P2M and 44% to 66% on ResGen across the four models, in every case matching or exceeding the strongest projection-based baseline (SynFormer 30.7% on P2M, 33.8% on ResGen). The detailed numbers are in main-text Table 1, reproduced inline as Response Table 9 below for convenience.

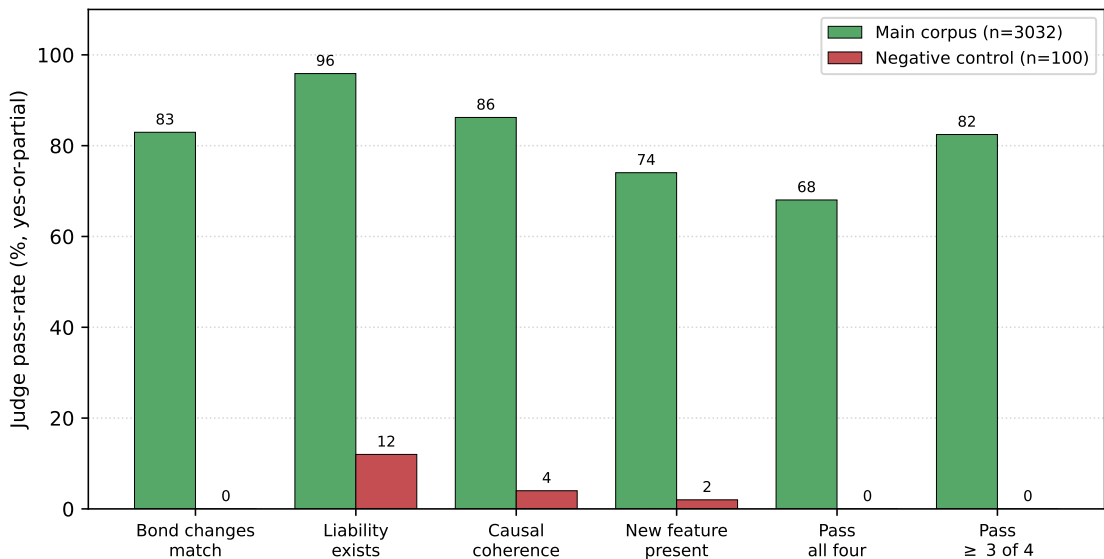
Method	Sim > 0.5	Sim > 0.6	Sim > 0.7	Sim > 0.8
<i>Source: Pocket2Mol (P2M)</i>				
P2M-ChemProjector	21.6%	9.8%	4.2%	1.9%
P2M-SynFormer (115 templates)	30.7%	14.1%	6.2%	2.9%
P2M-ReaSyn	30.4%	15.4%	6.4%	3.0%
P2M-SynCraft (Gemini-2.5-Pro)	42.7%	28.4%	11.9%	3.2%
P2M-SynCraft (GPT-5.2) [‡]	39.0%	27.1%	12.7%	2.6%
P2M-SynCraft (DSv4-Pro) [†]	64.3%	50.1%	27.8%	8.5%
P2M-SynCraft (DSv4-Flash) [†]	64.9%	48.2%	28.5%	8.2%
<i>Source: ResGen</i>				
ResGen-ChemProjector	28.3%	14.8%	7.2%	2.0%
ResGen-SynFormer (115 templates)	33.8%	19.9%	8.7%	3.5%
ResGen-ReaSyn	37.5%	22.0%	10.5%	4.3%
ResGen-SynCraft (Gemini-2.5-Pro)	44.7%	29.1%	13.1%	3.5%
ResGen-SynCraft (GPT-5.2) [‡]	44.5%	31.6%	16.9%	4.8%
ResGen-SynCraft (DSv4-Pro) [†]	65.7%	51.3%	29.0%	10.0%
ResGen-SynCraft (DSv4-Flash) [†]	65.2%	50.1%	28.3%	8.6%

Response Table 9: Per-source rescue rate at increasing Tanimoto similarity thresholds for the four LLM backends and the three projection baselines (corresponds to main-text Table 1). [†] MIT-licensed open weight, pass-k=5. [‡] GPT-5.2 evaluated at pass-k=1 due to API budget.

Per-source rescue rates at Sim>0.5 range from 39% to 65% on P2M and 44% to 66% on ResGen across the four models, and all four reach or exceed the strongest projection-based baseline. The four backends span roughly a fourteen-month release window (Gemini-2.5-Pro preview release 2025-06-05; GPT-5.2 released 2025-12-11; DeepSeek-V4-Pro / DeepSeek-V4-Flash open-weight release 2026-04-24), and the rescue-rate ordering tracks this generation gap. We report the four-model comparison neutrally and retain Gemini-2.5-Pro as the headline backend to keep the rest of the manuscript aligned with the originally submitted main results. The key implication for the editor’s open-weight question is that the SynCraft framework is framework-agnostic: it produces competitive or better rescue rates on locally-deployable open-weight LLMs and does not depend on any specific commercial API.

(#2) For the construction of the reference dataset, “semantic intent ground truth” was generated with a LLM, but there is no explanation in the manuscript about how this “ground truth” was validated. Was it validated by experts or by computational means? How reliable is this “ground truth”? Validation of this training data, and precise description of the validation and its results, is necessary.

We agree that the reliability of the LLM-generated reasoning corpus needed explicit validation, and we apologise that the original submission did not provide one. We have now added a structured LLM-as-judge validation of the golden reasoning corpus (SI Section 2.7). An independent judge model (DeepSeek-V4-Pro, different from the generation model used in the paper headline) scores each reasoning chain on a four-axis structured rubric: (1) whether the described bond changes match the executed edits, (2) whether the cited synthetic liability genuinely exists in the source, (3) whether the causal narrative is coherent, and (4) whether the new beneficial feature is present in the edited product. Each axis is rated yes / partial / no. We compare scores on the full reasoning corpus to those on a paired negative control in which the rationale text is shuffled to a non-matching source–target pair.



Response Figure 4: LLM-as-judge alignment on the SynCraft golden reasoning corpus versus a paired negative-control set in which the rationale text is shuffled to a non-matching source–target pair. Per-axis pass rate (yes-or-partial) and the two aggregate pass criteria (pass ≥ 3 of 4, pass all 4) are shown.

On the main corpus the great majority of chains pass the rubric on at least three of the four axes, while the paired negative control passes 0% under the same criterion. The judge therefore (i) clearly discriminates between coherent and incoherent rationales, and (ii) gives the corpus a substantively positive verdict. We have also clarified in the Methods (*Reference dataset construction*) that the LLM-produced reasoning is a teaching signal: the edit-execution and SimpRetro synthesisability evaluation are independent computational checks, so the framework’s main rescue-rate metrics do not depend on perfect reasoning correctness.

(#3) An aspect of the quantitative validation that is puzzling is the following. Initially, five models were used to generate virtual molecules: LIGAN, 3D-SBDD, Pocket2Mol, DiffSBDD, and ResGen. Then, molecules generated by two of these models (LIGAN and 3D-SBDD) are explicitly excluded from the validation because they “had exceptionally low rates of synthetic accessibility”. (But presumably they were not from

training? Why?) Another model (DiffSBDD) is not explicitly excluded, but it is never mentioned in the text, and no results for it are shown or described, so are left to guess that this model was also excluded, but we do not know why. The authors should report results for molecules generated by all five models, and let the reader decide what to do with this information.

Apologies for the original ambiguity. The original Table 1 was restricted to Pocket2Mol and ResGen because those two source pools have the cleanest chemistry-quality profile (mean SA ~ 4 vs. 4.9–5.1 for LiGAN, AR and DiffSBDD), and we wanted the rescue-rate comparison to be controlled for source-quality variation. The remaining three baselines (LiGAN, AR / 3D-SBDD, DiffSBDD) were not used in training either; they were simply not reported. We agree that the broader-coverage view is more informative, and we now report all five.

Specifically, we ran all four methods (ChemProjector, SynFormer, ReaSyn and SynCraft) on heavy-atom-stratified 10% subsets of LiGAN (n=372), AR / 3D-SBDD (n=365) and DiffSBDD (n=383), under the same SimpRetro+Enamine protocol used in Table 1.

Source pool	Method	Sim>0.5	Sim>0.6	Sim>0.7	Sim>0.8
LiGAN (n=372)	ChemProjector	1.9%	0.8%	0.0%	0.0%
	SynFormer (115 templates)	11.6%	6.2%	3.8%	1.1%
	ReaSyn	14.0%	9.1%	5.9%	2.7%
	SynCraft (DSv4-Pro)	25.5%	17.5%	6.7%	1.3%
AR / 3D-SBDD (n=365)	ChemProjector	10.1%	5.2%	1.9%	0.8%
	SynFormer (115 templates)	7.7%	3.8%	1.9%	0.3%
	ReaSyn	14.5%	6.3%	3.0%	1.6%
	SynCraft (DSv4-Pro)	26.3%	17.8%	8.2%	1.9%
DiffSBDD (n=383)	ChemProjector	6.5%	2.6%	0.5%	0.0%
	SynFormer (115 templates)	6.8%	2.6%	1.6%	0.8%
	ReaSyn	25.6%	10.7%	3.9%	1.6%
	SynCraft (DSv4-Pro)	38.1%	26.1%	14.1%	3.7%

Response Table 10: Four-method per-source rescue rate on heavy-atom-stratified 10% subsets of three additional 3D-aware generative baselines (LiGAN, AR / 3D-SBDD and DiffSBDD), evaluated under the same SimpRetro+Enamine protocol used in main-text Table 1. SynCraft uses DeepSeek-V4-Pro at pass-k=1; projection-based methods use their default published settings. Full table including all five baselines is in SI Section 6.

SynCraft leads every Sim>0.5 cell by 11–23 percentage points and is first at 11 of the 12 (baseline $\times$ threshold) cells in this table; the only cell where a projection method wins is LiGAN at Sim>0.8 (ReaSyn 2.7% vs. SynCraft 1.3%), a threshold at which all four methods have single-digit success counts and the ranking is noise-dominated. Combined with the P2M and ResGen results in Table 1, SynCraft is the leading method on Sim>0.5 across all five 3D-generative source distributions. Notably, the ranking among projection methods is non-monotonic across baselines (ChemProjector strong on AR but very weak on LiGAN; ReaSyn strongest on DiffSBDD), suggesting that projection methods are more sensitive to the chemistry character of the source pool than the editing paradigm is.

(#4) The summary statistics provided in Table 1 mix many different aspects of the performance of the algorithms. What goes wrong when an algorithm (SynCraft, in particular) fails? Does it fail because the proposed molecules are not synthesizable? Or because the edit sets are invalid or lead to invalid molecules? Or because the resulting molecules have a similarity with the original molecule below the desired threshold? All these different statistics should be reported separately, so that the reader can get a more complete picture

of the strengths and limitations of the proposed approach vis a vis the baselines.

Indeed, the bare per-source rescue rate obscures where the failures actually come from, and we are grateful for the clear three-way breakdown the question proposes. We have added a per-source failure-mode decomposition across all four LLM backends and both source pools (SI Section 8, reproduced inline as Response Table 11). The four categories mirror the three failure modes raised in the question plus a success class, evaluated across all pass-k attempts a source receives:

- **Success:** ≥ 1 pass yields a SimpRetro-solvable product with $\text{Sim} \geq 0.5$ to source. By construction the Success bar matches the $\text{Sim} > 0.5$ row of Table 1.
- **Sim drift:** ≥ 1 retro-solvable product, but every such product has $\text{sim} < 0.5$ to source.
- **Retro fail:** ≥ 1 valid reconstructed product, no product retro-solvable within the 30-minute budget.
- **Recon fail:** no pass yields a valid reconstructed product (malformed JSON, invalid atom-map reference, or disconnected graph).

Pool	LLM	Success	Sim drift	Retro fail	Recon fail
P2M	Gemini-2.5-Pro	42.7%	20.0%	26.3%	11.0%
P2M	GPT-5.2	39.0%	17.8%	38.7%	4.5%
P2M	DSv4-Pro	64.3%	13.9%	21.1%	0.7%
P2M	DSv4-Flash	64.9%	17.2%	17.4%	0.5%
ResGen	Gemini-2.5-Pro	44.7%	17.6%	27.2%	10.6%
ResGen	GPT-5.2	44.5%	14.8%	37.0%	3.7%
ResGen	DSv4-Pro	65.7%	12.6%	21.5%	0.2%
ResGen	DSv4-Flash	65.2%	14.5%	19.9%	0.4%

Response Table 11: Per-source failure-mode decomposition for the four LLM backends on the P2M and ResGen source pools. The four categories form a per-source partition: every source falls into exactly one of Success / Sim drift / Retro fail / Recon fail.

Three observations. First, *retro-unsolvability is the dominant failure mode across all four LLMs* (17–39% of sources), pointing to retrosynthesis coverage as the next-most-impactful improvement direction rather than LLM choice. Second, *Recon fail is uniformly low for the open-weight DeepSeek backends ($< 1\%$) and GPT-5.2 (4–5%)* but materially higher for Gemini (11%), consistent with the disconnected-fragment behaviour reviewer R1 noted in the code review on Gemini-2.5-Flash; the framework’s sanitisation step catches these and excludes them from the pass-k pool. Third, *Sim-drift fractions are similar across LLMs* (13–20%), suggesting that the tendency toward “too distant” rescues when a closer rescue is unavailable is a framework-level property rather than LLM-specific.

The Success-rate differences across LLMs in Table 1 (Gemini 42.7% versus DSv4-Pro 64.3% on P2M) reflect two contributions: (i) pass-k differences (GPT-5.2 at pass-k=1 due to API budget vs. DSv4 and Gemini at pass-k=5); and (ii) per-pass quality, where the more recently released open-weight DSv4 backends produce higher-quality individual edits than Gemini-2.5-Pro from a ~14-month earlier release. Section 8 discusses this decomposition explicitly.

(#5) *Related to the previous point, it is not entirely clear to me whether the comparison in Table 1 is completely fair, because at least some baseline models operate by projecting onto a known, limited synthesizable chemical space (defined in some way), whereas SynCraft can generate arbitrary new compounds not present in any the chemical spaces in which other methods operate. What would happen if, for example, we enlarged the synthesizable chemical space in SynFormer by allowing more reactions, more building blocks or more steps? Would the performance gap close? In other words, is the improvement of SynCraft with respect to the baselines due to a better capacity to reach synthesizable molecules, or just to the fact that it is allowed to*

reach molecules that are forbidden to other methods?

This is exactly the right control, and we ran it. We retrained SynFormer end-to-end on the full USPTO 51,720-template superset under matched compute (architecture and loss unchanged; reaction-classification head expanded from 120-way (115 templates + 5 reserved indices in the published checkpoint) to 51,725-way (51,720 USPTO templates + the same 5 reserved indices); training and evaluation details in SI Section 4). The result is reproduced inline as Response Table 12:

Configuration	Sim > 0.5	Sim > 0.6	Sim > 0.7	Sim > 0.8
P2M-SynFormer (115 templates, published)	30.7%	14.1%	6.2%	2.9%
P2M-SynFormer (51,720 templates, re-trained by us)	5.6%	3.0%	1.2%	0.8%
P2M-SynCraft (Gemini-2.5-Pro)	42.7%	28.4%	11.9%	3.2%
ResGen-SynFormer (115 templates, published)	33.8%	19.9%	8.7%	3.5%
ResGen-SynFormer (51,720 templates, re-trained by us)	6.6%	3.3%	1.5%	0.8%
ResGen-SynCraft (Gemini-2.5-Pro)	44.7%	29.1%	13.1%	3.5%

Response Table 12: SynFormer retrained under matched compute on the full USPTO 51,720-template superset, evaluated on the published P2M and ResGen unsynthesisable test sets used in main-text Table 1. (Identical to Response Table 1 under R1 W2/Q5; reproduced here for convenience.)

The enlarged-template SynFormer in fact performs *worse* than the published 115-template baseline on both source pools (P2M Sim>0.5 drops from 30.7% to 5.6%; ResGen from 33.8% to 6.6%), and remains far below SynCraft (42.7% and 44.7%). Several mechanisms plausibly contribute: (i) long-tail USPTO templates each appear in only tens of training reactions, so the per-template signal is two to three orders of magnitude sparser than for the curated 115-template set; (ii) reaction-building-block coverage in the matched matrix is roughly 15 $\times$ sparser; (iii) at each retrosynthesis step the model now picks among $\sim$ 50,000 alternatives, so modest per-step calibration loss compounds along a tree of depth 3–5; and (iv) the 115-template inventory was hand-curated for Enamine compatibility, which the long-tail USPTO templates largely are not.

We also cite the independent precedent of Sun *et al.* [2] (“SynLlama”, ACS Cent. Sci. 2025), who report that at the matched 115-template setting on the 1,000-molecule Enamine Diversity Set, their LLM-based SynLlama reconstructs 691 molecules versus SynFormer’s 660 under the same Enamine building-block constraint. The authors themselves characterise this as the LLM “matching” SynFormer’s performance with roughly 40 $\times$ less training data rather than as a paradigm-vs-template-count argument, and we cite the direct numerical comparison only without over-attributing that interpretation to their work. Read together with our SynFormer-51K experiment, the joint evidence is that the SynCraft improvement in Table 1 is unlikely to be explained simply by access to a larger reachable chemical space, since enlarging that space for SynFormer does not help.

(#6) *The qualitative validation on specific molecules (Figs. 2–4 and corresponding sections) is interesting and illuminating, but it is impossible for the reader to evaluate whether these are representative cases as opposed to cherry-picked examples where things work as they should. A systematic evaluation of the claims in this part of the article is, therefore, necessary. For example, how can the reasoning of SynCraft be validated systematically? How often is it correct (as estimated by experts) as opposed to simply plausible? Or what*

percentage of rescues (over a larger set of validation samples) preserve binding mode, as the example discussed in Fig. 4?

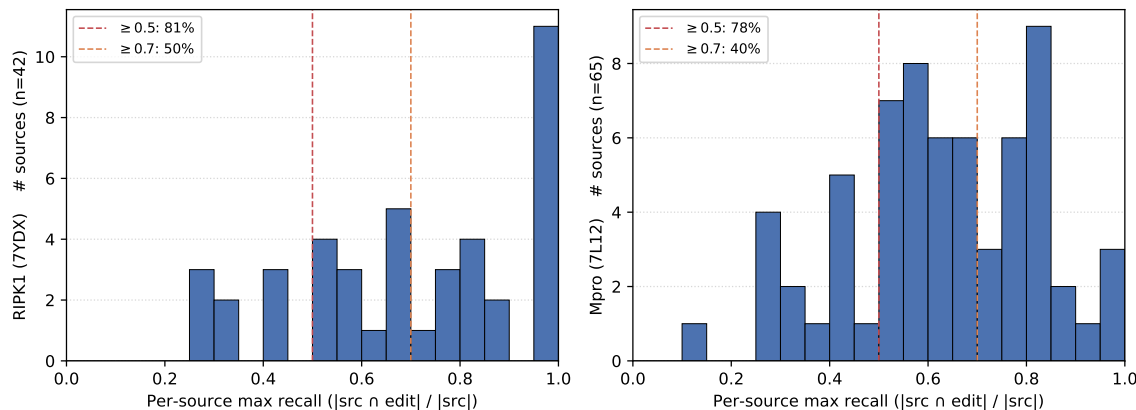
We address both halves of this concern with two systematic large- N analyses on independent targets.

(i) Reasoning quality. The structured LLM-as-judge validation introduced for R2 #2 is the systematic answer to the “how often is the reasoning correct” half. On the full reasoning corpus the great majority of chains pass on at least three of the four axes; on a paired negative control the same judge passes 0% on the same criterion (Response Figure 4). We chose an automated judge over a small- N expert annotation because it scales to the full corpus, where a human review would require thousands of person-hours and is itself subject to inter-annotator variance; the four-axis rubric was designed to mirror the questions an expert would ask of a single rationale, and the negative-control discrimination demonstrates the judge is not simply rubber-stamping.

(ii) Binding-mode retention. We computed systematic PLIP-based binding-mode retention on two unrelated targets (SI Section 7). The RIPK1 figures are produced with the interaction-aware (PLIP-constrained) skill mode; the M^{pro} Vina figures correspond to the same interaction-aware mode for consistency. The default-skill (union) M^{pro} Vina is additionally reported separately under R1 W4/Q8 above (Response Figure 1).

Metric	RIPK1 (7YDX)	M ^{pro} (7L12)
Source–edit pose pairs analysed	191	260
Number of source candidates	42	47
Mean per-pair recall vs source	0.47	0.42
Sources with best-edit recall ≥ 0.5	34/42 (81%)	37/47 (79%)
Sources with best-edit recall ≥ 0.7	21/42 (50%)	19/47 (40%)
Sources with best-edit recall = 1.0	11/42 (26%)	2/47 (4%)
Mean Δ Vina (best edit – source, kcal/mol)	−0.34	+0.14
Sources with best edit Vina $\leq$ source	31/42 (74%)	11/30 (37%)
Sources with best edit Vina within 0.5 kcal/mol	37/42 (88%)	18/30 (60%)

Response Table 13: Systematic PLIP-based binding-mode retention on two structurally and mechanistically unrelated targets (RIPK1 serine/threonine kinase and SARS-CoV-2 M^{pro} cysteine protease). PLIP-retention rows use the interaction-aware (PLIP-constrained) skill mode on both targets. M^{pro} Vina rows correspond to the default-skill (union) ranking; the per-source pairs are reduced to $n = 30$ because two of the 32 union-rescued M^{pro} sources did not yield a paired Vina docking score (ligand-preparation failure on the edit side).



Response Figure 5: Per-source distribution of maximum PLIP recall (left columns) and maximum Jaccard (right columns) over SynCraft-edited products on RIPK1 (top row) and M^{pro} (bottom row). Vertical lines mark the 0.5 and 0.7 recall thresholds whose source-level pass rates appear in Response Table 13.

On both targets the majority of source candidates have at least one SynCraft-edited analog that retains $\geq 50\%$ of the source PLIP interaction profile. On RIPK1 the best edit also improves the Vina docking score on average ($\Delta V_{\text{ina}} -0.34$ kcal/mol). On M^{pro}, the per-source best retro-validated edit under the default skill ranking reproduces the source Vina docking score to within ± 0.5 kcal/mol for 60% of the rescued sources, with a mean ΔV_{ina} of $+0.14$ kcal/mol (Response Figure 1). The two targets are from entirely unrelated protein families (kinase vs. cysteine protease), so the consistency of the retention results across them supports the view that the binding-mode preservation generalises rather than being target-specific.

We retain the qualitative case studies (main-text Figures 2–4) as mechanistic illustrations of individual edits, and the RIPK1 ($n = 42$) and M^{pro} ($n = 47$) systematic analyses above are the corresponding larger- N answers to the cherry-picking concern. The caption of Figure 4 has been updated to cross-reference the systematic RIPK1 cohort directly, so the reader can locate the larger- N context from the case-study figure itself.

Minor Issues

(#7) The LLM is restricted to outputting a JSON-formatted sequence of commands. Is this format always respected? Are the generated sequences always valid and, if not, how often are they invalid and why?

The JSON format is essentially always respected. Across the four-LLM comparison on the full P2M and ResGen source pools, parseable JSON output was produced in $\geq 98\%$ of source-pass attempts on every backend (Response Table 14). The rare failures are typically empty responses or premature mid-token truncation, and are simply discarded from the pass-k pool with no format-recovery scaffolding required.

LLM	pass-k	JSON parse rate
Gemini-2.5-Pro	5	100.0%
GPT-5.2	1	99.6%
DeepSeek-V4-Pro	5	98.3%
DeepSeek-V4-Flash	5	100.0%

Response Table 14: Per-LLM JSON-parseability rate on the P2M source pool (ResGen rates are within 1–2 percentage points). The downstream per-source partition into Success / Sim-drift / Retro-fail / Recon-fail is given separately in Response Table 11 (R2 #4); note that the per-source Recon-fail rate there is dominated by valence/disconnection failures rather than JSON-parse failures.

(#8) The methods section describes how synthesizability was established for the ground truth reference data set, but not for the molecules generated by SynCraft in the validation set of the quantitative evaluation. Was it established the same way (SimpRetro for 30 minutes?).

Yes. The SimpRetro protocol used to validate the SynCraft-generated edits in the quantitative evaluation is identical to the one used to construct the reference dataset: USPTO templates with frequency ≥ 2 , the Enamine commercial building-block library, and a 30-minute wall-clock budget per molecule on a single CPU core. We have added a dedicated SI section (*SimpRetro Validation Protocol*, Section 1.8) that states this once, explicitly, and the corresponding Methods paragraph in the revised manuscript points to it. The same explicit description also addresses R1 W3 (above), where the same observation was made about the original Methods text.

(#9) What is the meaning of the quotation marks around “synthetic nightmare”? Are these the LLM words? An expert’s?

Apologies for the ambiguity. The phrase was our own informal characterisation and the quotation marks were used only for emphasis. We have replaced it with a more direct phrasing (“is difficult to be synthesized”) in the main text.

Code and Data

The revised release adds (i) a `notebooks/` directory containing analysis notebooks for every figure and table; (ii) the packaged agent skill in `skill/`; (iii) per-experiment reproduction scripts under `scripts/` (one `reproduce_*.sh` per headline experiment). The repository now also includes the multi-LLM inference outputs, the systematic PLIP-retention CSVs, the eMolecules sanity-check inputs, and the M^{pro} skill demo trace.

We thank both reviewers and the editor again for the detail and care of their reading. The comments and suggested experiments have substantially sharpened the work, particularly the random-RAG ablation, the systematic binding-mode retention analysis, and the failure-mode decomposition; each of these has changed our own understanding of where the framework’s leverage actually comes from. We see SynCraft as a concrete interface through which the general-purpose reasoning of modern LLMs can be brought to bear on the practical chemistry of hit optimisation, and we are grateful for the opportunity to refine the manuscript along these lines. We hope these revisions meet the standard required for publication in *Nature Machine Intelligence*.

Junren Li and Luhua Lai

References

- [1] Benoit Baillif, Jason Cole, Patrick McCabe, and Andreas Bender. Benchmarking structure-based three-dimensional molecular generative models using genbench3d: ligand conformation quality matters. *arXiv preprint arXiv:2407.04424*, 2024.
- [2] Kunyang Sun, Dorian Bagni, Joseph M Cavanagh, Yingze Wang, Jacob M Sawyer, Bo Zhou, Andrew Gritsevskiy, Oufan Zhang, and Teresa Head-Gordon. Synllama: generating synthesizable molecules and their analogs with large language models. *ACS Central Science*, 2025.
- [3] Yuxin Wang et al. Retrieval-augmented in-context reasoning for synthesis-aware molecular generation. *arXiv preprint arXiv:2505.07027*, 2025.

POINT-BY-POINT RESPONSE TO THE REFEREES

Guiding Large Language Models to Predict Edit Sequences for Molecular Synthesizability Optimization

Manuscript reference: NATMACHINE-25126069A

We thank the editor and the reviewers for their careful and constructive assessment of our revised manuscript. We have addressed all remaining comments below. Reviewer comments are reproduced in italic, followed by our responses and the corresponding revisions.

Reviewer #1

We sincerely thank Reviewer #1 for the positive assessment of the revised manuscript and for confirming that the previous concerns have been fully addressed.

Reviewer #2

Comment 1

The new comparison of LLMs shows large discrepancies between LLMs. In fact, the difference between LLMs is larger than the difference between the original results with Gemini and traditional methods. The authors should probably comment on this fact.

Response

We agree with the reviewer. At Sim > 0.5, the four LLM backends span 39.0-64.9% on Pocket2Mol (a 25.9-percentage-point range) and 44.5-65.7% on ResGen (a 21.2-percentage-point range). These ranges are larger than the corresponding margins of Gemini-2.5-Pro over the strongest projection baseline: 12.0 percentage points on Pocket2Mol (42.7% versus 30.7%) and 7.2 percentage points on ResGen (44.7% versus 37.5%). Thus, the underlying LLM is indeed a first-order determinant of SynCraft's absolute performance.

We have revised Supporting Information Section 1.7 to state this result explicitly and to clarify that SynCraft should be interpreted as a backend-agnostic editing framework rather than a model-independent level of performance.

Comment 2

In their rebuttal, the authors claim that "The remaining three baselines (LiGAN, AR / 3D-SBDD, DiffSBDD) were not used in training either." However, the paper still claims that "From the pool of molecules designated as unsynthesizable, we constructed our 'Synthesis Cliff' dataset, which provides the ground-truth examples for our study. To do this, we first randomly sampled 4,000 unsynthesizable molecules from each of the FIVE generative models."

Response

We thank the reviewer for identifying this inconsistency. The description in the Methods is correct; the quoted sentence in our previous rebuttal was incorrect. Molecules from all five source generators (LiGAN, AR/3D-SBDD, Pocket2Mol, DiffSBDD and ResGen) contributed to the Synthesis Cliff retrieval corpus.

In our previous response, we used the term 'training' narrowly to refer to task-specific parametric training or fine-tuning, which SynCraft does not perform. In the context of the reviewer's question, however, this wording was misleading because the five-model Synthesis Cliff corpus is used for retrieval-augmented in-context inference. We apologize for the miscommunication and correct our previous statement here.

We further confirm that the retrieval corpus and evaluation molecules are strictly disjoint: the evaluation sets were constructed from the remaining unsampled molecules, and no evaluation molecule is included in the retrieval corpus.

Comment 3

Response Figure 5 refers to some 'rows', but the figure only has one row. I did not check if this figure appears in the SI, but, if it does, figure and/or caption probably need to be fixed.

Response

Thank you for catching this error. The reviewer is correct: the figure appears as Supplementary Figure 14 and contains a single row with two panels. Both panels show per-source maximum interaction recall, for RIPK1 on the left and SARS-CoV-2 Mpro on the right. No Jaccard distribution is shown.

We have corrected the Supplementary Figure 14 caption so that it describes only the two recall distributions. This was a caption error only; the plotted data, the numerical results and the conclusions are unchanged. We also corrected other typos in the supporting information accordingly.