BRIDGE-BENCH: BENCHMARKING REAL-WORLD ISSUES IN DRUG GENERATION AND EVALUATION

Anonymous authors
Paper under double-blind review

ABSTRACT

Large language models (LLMs) and tool-using agents are increasingly proposed as assistants for drug discovery, but their evaluation remains fragmented. Existing benchmarks often emphasize isolated capabilities, such as molecular property prediction and chemical question answering, while practical therapeutic research requires coordinated decisions across biology, chemistry, pharmacology, modeling, manufacturability, and translation. We introduce BRIDGE-Bench (Benchmarking Real-world Issues in Drug Generation and Evaluation), a pipeline-level benchmark built from real-world drug discovery and development questions. BRIDGE-Bench covers target biology, hit and lead discovery, binding pocket analysis, absorption, distribution, metabolism, and excretion (ADME), quantitative systems pharmacology (QSP), retrosynthesis, antibody–drug conjugate (ADC) design, drug delivery, chemistry, manufacturing, and controls (CMC), and wet-lab planning. Each instance is organized around a concrete decision scenario with reference answers and scoring criteria. We evaluated 13 harness–model configurations within the Claude Code and Codex agent frameworks. Final average scores ranged from 19.3 to 36.0, with the strongest aggregate result obtained by Claude Code with `claude-opus-5`. The benchmark supports a modular and trajectory-level evaluation of answer correctness, evidence grounding, tool use, uncertainty awareness, cross-stage trade-off reasoning, and error modes that matter for practical therapeutic decisions.

1 INTRODUCTION

Drug discovery is a long, uncertain and decision-intensive process with high development costs. A therapeutic program may fail because the selected target is weakly causal, the binding pocket is not druggable, the generated compounds are potent but not developable, the synthetic route is unrealistic, the modality is mismatched to the disease biology, the formulation or manufacturing plan is fragile, or the proposed experiment cannot test the intended hypothesis. Failures often arise not from isolated prediction error, but from missed constraints across stages.

Large language models (LLMs) and tool-using agents are increasingly being explored for drug discovery. These systems can search for literature, reason over biomedical evidence, call chemistry tools, analyze molecular or structural information, and assist with experimental planning (Bran et al., 2024; Boiko et al., 2023). This makes them attractive as research assistants, but it also makes evaluation harder. A useful drug-discovery agent should not only answer factual questions or optimize a molecule against a narrow objective. Rather, it should plan, use evidence, call tools appropriately, reason under uncertainty, and make development-aware recommendations.

Existing benchmarks have made important progress in molecular machine learning, chemical reasoning, biomedical question answering, and agentic tool using (Wu et al., 2018; Huang et al., 2021; Mirza et al., 2025). However, many evaluations remain organized around isolated tasks, data modalities, or early discovery problems. They less directly test whether an agent can reason across the broader therapeutic pipeline, especially stages such as predicting absorption, distribution, metabolism, and excretion (ADME), constructing quantitative systems pharmacology (QSP) models, deducing retrosynthesis pathways, designing antibody–drug conjugates (ADCs), and performing chemistry, manufacturing, and controls (CMC) analyses including drug formulation. BRIDGE-

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

Bench addresses this gap by evaluating agents through realistic decision scenarios centered on real development pain points.

Our main contributions are as follows.

- We introduce BRIDGE-Bench, a pipeline-level benchmark for evaluating LLM agents on real-world issues in drug discovery and development.
- We design a pain-point-oriented taxonomy that covers target biology, hit and lead discovery, binding pocket analysis, ADME, QSP, retrosynthesis, ADC design, drug delivery, CMC/manufacturing, and wet-lab planning.
- We evaluate agents along both final-answer quality and agent-process quality, including planning, tool use, evidence grounding, uncertainty awareness, and simulation reasoning.
- We benchmark 13 harness–model configurations, covering six Codex-based systems and seven Claude Code-based systems.

2 Related Work

Benchmarks for molecular machine learning. Molecular machine learning has long relied on standardized benchmarks for property prediction, virtual screening, molecular generation, and molecular representation learning. MoleculeNet (Wu et al., 2018) introduced a benchmark suite covering biophysical, physiological, physical chemistry, and quantum mechanical tasks with standardized metrics and dataset splits. Therapeutics Data Commons (TDC) (Huang et al., 2021) expanded this idea into a broader platform for therapeutics machine learning, integrating datasets, task formulations, data functions, generation oracles, and leaderboards across drug discovery and development. Molecular generation benchmarks such as GuacaMol (Brown et al., 2019) and Molecular Sets (MOSES) (Polykovskiy et al., 2020) further standardized evaluation of de novo design systems. These resources remain foundational for model comparison, but they primarily evaluate predictive or generative models under predefined task settings. BRIDGE-Bench instead evaluates agents in decision scenarios that combine evidence interpretation, multi-objective trade-offs, and downstream consequences.

LLM benchmarks for chemistry and biomolecular reasoning. Recent benchmarks have begun to evaluate LLMs in chemistry and molecular science. ChemBench (Mirza et al., 2025) assesses chemical knowledge and reasoning abilities of LLMs against the expertise of practicing chemists. Mol-Instructions (Fang et al., 2024) provides a large-scale biomolecular instruction dataset covering molecule-oriented, protein-oriented, and biomolecular text tasks. MoleculeQA (Lu et al., 2024) focuses on factual accuracy in molecular comprehension, with question-answer pairs grounded in curated molecular descriptions. These benchmarks are important for assessing whether LLMs understand chemical and biomolecular concepts, but their primary unit of evaluation is still the question, instruction, or molecular fact. BRIDGE-Bench differs by grounding each instance in a drug discovery pain point and requiring agents to produce actionable, risk-aware decisions.

Drug discovery agent benchmarks. Several recent benchmarks move closer to real drug discovery workflows. DMPKBench (Li et al., 2025) evaluates LLMs and agents on drug metabolism and pharmacokinetics tasks, including multimodal and quantitative reasoning. Small-Molecule Drug Design Benchmark (SMDD-Bench) (Han et al., 2026) evaluates LLM agents on small-molecule drug design tasks that require chemical reasoning, three-dimensional intuition, and tool use. DrugDiscoveryBench (Scale AI & Phylo, 2026) evaluates coding agents on multi-step early-discovery workflows spanning target identification, hit identification, patent mining, cheminformatics, structural reasoning, structure–activity relationships, and affinity analysis. DrugBench (GHDDI AI Lab, 2025) constructs PubMed-derived chain-of-thought multiple-choice questions across the drug research and development life cycle. These works demonstrate the value of domain-specific and agentic evaluation. BRIDGE-Bench builds on this direction but targets broader cross-stage evaluation, explicitly covering early discovery, QSP, retrosynthesis, ADC design, CMC, delivery, and wet-lab planning within one pain-point-oriented framework.

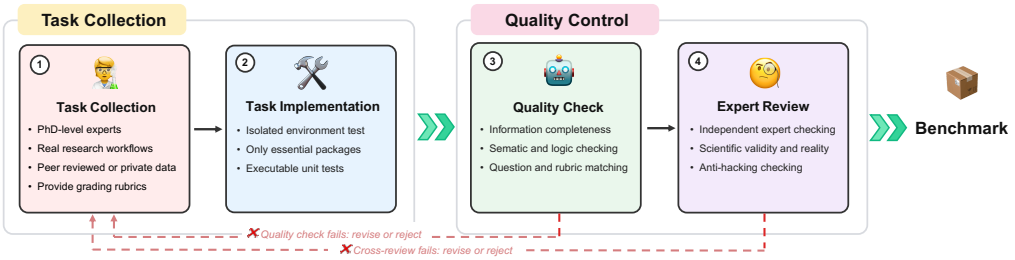


Figure 1: Expert-guided task collection and quality-control workflow for BRIDGE-Bench. PhD-level experts contribute real-life drug discovery and development questions, which are later checked for scientific validity, plausibility, answerability, and rubric consistency before inclusion.

3 Data Collection

3.1 Question Collection and Quality Control

BRIDGE-Bench is built from questions contributed by human experts. We recruit PhD-level domain experts with front-line experience in research and development to construct benchmark questions. Experts were asked to contribute realistic scientific or development pain points, rather than textbook facts, isolated prediction tasks, or prompts answerable by surface-level retrieval.

Every accepted question is paired with a reference answer and grading rubric. The reference answer specifies the expected decision, the rationale, and key evidence, while the rubric scores both final-answer and agent-process quality, including scientific correctness, evidence grounding, uncertainty calibration, planning, tool use, and cross-stage consistency when applicable. Each task is scored out of 100. Every submitted question was subject to quality control before being accepted. Both human and AI reviewers checked scientific validity, plausibility, answerability, and scoring consistency. Questions that did not meet the requirements were sent back to human experts for revision until qualified.

Because BRIDGE-Bench focuses on exploratory scientific tasks, most of the questions cannot be evaluated reliably by exact-match rules. A scientifically valid response may combine literature evidence, mechanistic reasoning, and development-aware trade-offs in ways that are difficult to reduce to a single predetermined string or numeric answer. We therefore use rubric-guided LLM scoring to evaluate both the final answer and the reasoning process. The scoring prompt is conditioned on the reference answer and the expert-written rubric, and asks the evaluator to assess whether the model reaches a scientifically justified conclusion, uses relevant evidence, acknowledges uncertainty, and follows an appropriate problem-solving trajectory.

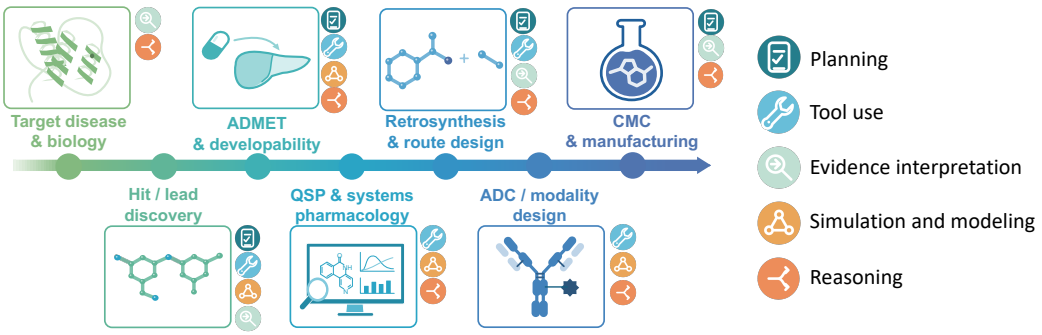


Figure 2: Illustration of preclinical workflow coverage and agent abilities in BRIDGE-Bench.

3.2 Dataset Scale and Pipeline Coverage

BRIDGE-Bench aims to provide broad coverage of the preclinical drug discovery pipeline, spanning as many key stages as possible (illustrated in Figure 2). It contains 38 questions spanning 10 drug discovery and development stages (illustrated in Figure 3). It includes both literature-grounded questions and experiment-inspired questions. Literature-grounded questions were adapted from public papers and databases. Experiment-grounded questions were designed based on experiments done by experts.

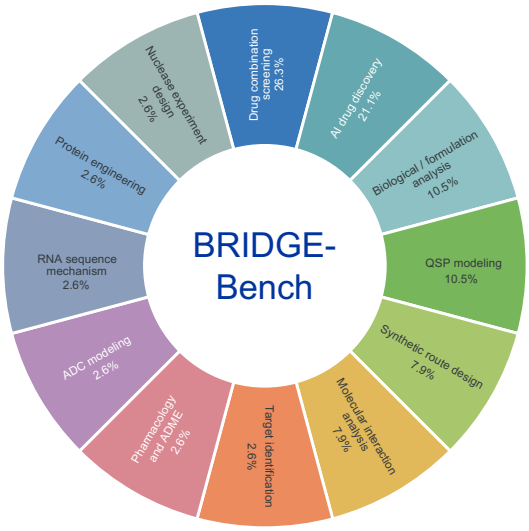


Figure 3: Question-type taxonomy in BRIDGE-Bench. The benchmark includes questions spanning drug combination screening, AI drug discovery, biological and formulation analysis, QSP modeling, synthetic route design, molecular interaction analysis, target identification, pharmacology and ADME, ADC modeling, RNA sequence mechanism, protein engineering, and nuclease experiment design.

Figure 2 illustrates the high-level preclinical drug discovery and development workflow used to organize the benchmark, together with the core agent abilities targeted by the question design.

We designed the taxonomy by comparing stage coverage in existing drug-domain agent benchmarks (Table 1).

Table 1: Coverage of drug discovery and development stages in representative drug-domain agent benchmarks.

Benchmark	**Early discovery**			**Translational modeling**		**Design and development**				
	Target biology	**Hit/Lead discovery**	**Binding pocket**	**ADME/DMPK**	**QSP modeling**	**Retro-synthesis**	**ADC design**	**Drug delivery**	**CMC**	**Wet-lab planning**
DMPKBench (Li et al., 2025)	×	×	×	✓	×	×	×	×	×	×
SMDD-Bench (Han et al., 2026)	×	✓	✓	△	×	△	×	×	×	△
DrugDiscoveryBench (Scale AI & Phylo, 2026)	✓	✓	✓	△	×	×	×	×	×	×
BRIDGE-Bench (Ours)	✓	✓	✓	✓	✓	✓	✓	✓	✓	✓

Notation. ✓, △, and × denote explicit, partial, and little or no coverage, respectively.

3.3 LEAKAGE CONTROL

We implemented leakage controls to reduce the chance that benchmark performance reflects memorization rather than task-solving ability. The benchmark contains different types of question, and our strategies vary.

For questions derived from the literature, we mitigated potential leakage by substantially rephrasing the questions and reconstructing or modifying the associated images, therefore reducing overlap with publicly available source materials. For questions inspired by existing datasets, we removed or abstracted compound names, target names, project identifiers, institution names, batch identifiers, and other confidential details. When necessary, the numerical values were normalized or discretized.

4 EXPERIMENTS

4.1 EXPERIMENTAL SETUP

We evaluate BRIDGE-Bench using a unified protocol across 13 harness–model configurations. Each evaluated system consists of a base model and an agent harness that controls task execution. The evaluated configurations include six Codex-based systems with `gpt-5.6-sol`, `gpt-5.5`, `glm-5.3-flash`, `kimi-k3`, `gemini-3.5-flash`, and `doubao-seed-2-1-pro-260628`, and seven Claude Code-based systems with `claude-opus-4-8`, `claude-opus-5`, `doubao-seed-2-1-pro-260628`, `gemini-3.5-flash`, `glm-5.3-flash`, `kimi-k3`, and `qwen3.8-max`. This design includes both harness-specific systems and models shared across harnesses, allowing us to separate backbone-model effects from execution-scaffold effects. Agents were allowed to use Internet access and the tools available within their harnesses. Each task was run once for each harness–model configuration. If a run failed because of an execution or internet error, it was restarted.

We use grading rubrics provided by human experts together with `gpt-5.6-sol` as the LLM judge for evaluation. We do not rely on a script-based judge because our assessment is not limited to the correctness of the final output. It also considers the workflow, reasoning trajectory, tool use, and execution process that lead to the result. This allows us to evaluate whether the agent followed appropriate steps, handled intermediate decisions effectively, and produced the final answer through a sound and reliable process.

To reduce potential bias, the judging process is anonymized. The trajectory files do not contain model-identifying information; therefore the judge does not know which model is being judged. The testing and judging environments are also kept completely separate, so the judge evaluates only the submitted workflow and final outputs, without access to the original execution environment or model metadata.

4.2 MAIN RESULTS

Figure 4 summarizes the final average score for each of the 13 evaluated harness–model pairs. The highest aggregate average score is achieved by Claude Code with `claude-opus-5` (36.0).

The top four systems have similar mean scores in this single-run evaluation, suggesting that the strongest configurations should not be over-interpreted as a strict ordering. The scores ranging from 19.3 to 36.0 also show that many current agent configurations remain far from reliable on these drug-discovery tasks. The ranking should therefore be read together with the task-profile and error analyses below. High average scores reflect stronger task-level performance, but they do not imply uniform competence across different stages of the workflow.

Performance also varied by domain category. Chemistry-oriented reasoning tasks achieved the highest average score, approximately 46.0, whereas cancer drug-combination tasks averaged 28.0 and biological mechanism or prediction tasks averaged only 17.7. This pattern suggests that current agentic models are relatively stronger on tasks with structured reasoning paths, such as retrosynthesis or chemistry workflows, but struggle more with biological interpretation, mechanism reconstruction, and prediction tasks that require integrating incomplete evidence.

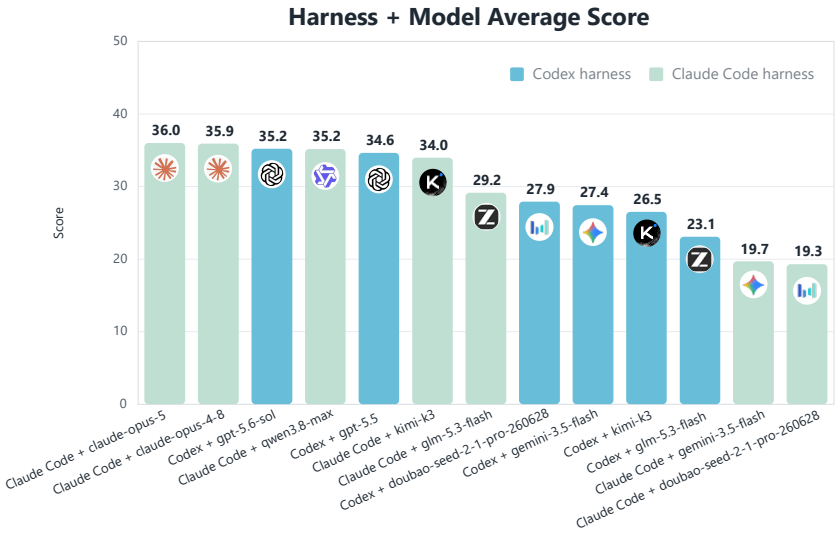


Figure 4: Main benchmark scoring results for the 13 evaluated harness–model configurations.

4.3 Model Strengths and Weaknesses

We further analyzed which task types produced relative strengths or unstable behavior for each harness–model configuration. Table 2 summarizes these qualitative profiles. Overall, stronger aggregate systems tended to be more reliable on mechanistic reasoning and evidence-chain tasks, whereas weaker or more unstable systems often failed because they skipped required workflows, substituted simulated data, or generalized beyond the available evidence.

Several patterns emerge from these profiles. The strongest models are not always better at every task type. Instead, they are less likely to collapse when a task combines domain reasoning with strict artifact requirements. On the other hand, weaker models often have isolated strengths, such as curve fitting, docking, or retrosynthesis, but these strengths do not transfer reliably to multi-step tasks that require candidate coverage, data provenance, and final-output compliance.

4.4 Harness Effects

At the aggregate harness level, Claude Code and Codex produced similar mean scores, 29.9 and 29.1 respectively. To further separate errors attributable to the base model from those introduced by the surrounding execution scaffold, we evaluated four base models that were evaluated under both the Codex and Claude Code harnesses: `doubao-seed-2-1-pro-260628`, `gemini-3.5-flash`, `glm-5.3-flash`, and `kimi-k3`. This paired setting allows us to compare shared backbones in a controlled manner while accounting for differences in prompt construction, tool mediation, file handling, and final-artifact submission, rather than treating them only as additional points in the aggregate ranking.

The paired results show that harness choice can alter both final scores and failure profiles. `doubao-seed-2-1-pro-260628` and `gemini-3.5-flash` performed better under Codex than under Claude Code (27.9 vs. 19.3 and 27.4 vs. 19.7, respectively). For both models, Codex improved stability on structured execution and deliverable-heavy tasks, including candidate screening, retrosynthesis, ADME, and ADC workflows, whereas Claude Code showed more diffuse failures, despite some advantage for `gemini-3.5-flash` on molecular docking and interaction analysis. The opposite pattern appeared for `glm-5.3-flash` and `kimi-k3`, which performed better under Claude Code than under Codex (29.2 vs. 23.1 and 34.0 vs. 26.5, respectively). Claude Code improved their performance on molecular docking, image quantification, mechanism-oriented reasoning, network pharmacology, retrosynthesis, and ADC tasks. Under Codex, their strengths were narrower, mainly in ADME, retrosynthesis, and, for `kimi-k3`, drug-combination tasks, while complex multi-step evidence organization remained weaker.

Table 2: Qualitative task-type profiles of evaluated harness–model configurations. Strengths and weaknesses are reported relative to the other evaluated systems, rather than as absolute capability claims.

Harness + Model	Clear relative strengths	Unstable areas / weaknesses
`Claude Code + claude-opus-5`	Most consistent across task types; strongest on **mechanistic/experimental reasoning**	Surprisingly weak on **retrosynthesis** in this run set; image quantification only average
`Claude Code + claude-opus-4-8`	Strong on **mechanistic/biochemical reasoning** and **retrosynthesis**; good evidence-chain construction; consistent across task types	Less strong on structured modeling tasks
`Codex + gpt-5.6-sol`	Most balanced Codex-side combination; strong on **mechanistic/biochemical reasoning**	Often loses points from incomplete candidate coverage, ranking logic, or missing full database workflow
`Claude Code + qwen3.8-max`	Strongest signal in **target docking/interaction** tasks	Weaker on image quantification; mechanism explanations less strong than Claude Opus models
`Codex + gpt-5.5`	Stable on **mechanistic reasoning**	Weaker on **image quantification** and docking; tends to miss exact numeric/statistical details
`Claude Code + kimi-k3`	Strong Claude-side pairing relative to Codex kimi	Still misses detailed verification or complete candidate lists
`Claude Code + glm-5.3-flash`	Good harness fit for **retrosynthesis**, docking, image quantification, and network-pharmacology-style tasks relative to Codex glm	Still weak on candidate coverage in drug-combination tasks
`Codex + doubao-seed-2-1-pro-260628`	Comparatively less weak on **image quantification**, ADC, and some cancer-combination tasks	No consistent standout; often loses points for incomplete workflows or insufficient provenance
`Codex + gemini-3.5-flash`	Strong on **curve fitting**; comparatively good on ADC DAR and retrosynthesis among weaker models	Very unstable
`Codex + kimi-k3`	Best Codex fit is **retrosynthesis** plus moderate cancer drug-combination screening	Tends to provide plausible but under-verified workflows
`Codex + glm-5.3-flash`	Clear relative strength in **structure inference**; does well on the ADME	Weak on **target docking**, image quantification, and mechanistic experimental reasoning; often incomplete evidence chains
`Claude Code + gemini-3.5-flash`	Best relative signal is **molecule interaction analysis**; some usable mechanism reasoning	Highly unstable; more format and deliverable failures
`Claude Code + doubao-seed-2-1-pro-260628`	Occasional acceptable performance on ADME and simple retrosynthesis	Overall weak; loses heavily on numeric/statistical rigor, image quantification, and required output compliance

Overall, Codex behaved more like a structured execution and deliverable-control layer, whereas Claude Code more often amplified research planning and long-chain reasoning. However, this effect was not uniform across all models. These results argue for reporting per-task error analyses rather than relying only on mean score. We report the main qualitative failure modes in Section 4.6.

4.5 Efficiency Profile

We also compare each harness–model configuration in terms of runtime, token usage, and final score (illustrated in Figure 5). This view is intended to separate task-solving abilities from raw computational effort.

The available trajectory statistics suggest that efficiency and accuracy were only weakly coupled. The correlation between execution duration and score was approximately -0.03. Thus, longer execution did not reliably translate into better benchmark performance.

In this analysis, more time and token usage should not be interpreted as characteristics of better models. A high-scoring but expensive configuration may be preferable for difficult scientific end-to-end tasks, whereas a lower-cost configuration may be useful for triage only if its failure modes are predictable and detectable. Reporting runtime, token cost, and score together provides a practical view of agent deployment trade-offs that is not captured by the overall scores alone.

4.6 Error Analysis

We further inspected low-scoring and failed runs to identify why current agents remain unreliable on realistic drug-discovery tasks. The dominant failures were not only factual mistakes. Many responses failed because they did not produce the required artifact, analyzed the wrong scientific object, substituted convenient proxy data for the requested input, or replaced the required computational workflow with plausible domain knowledge. Table 3 summarizes the main failure modes.

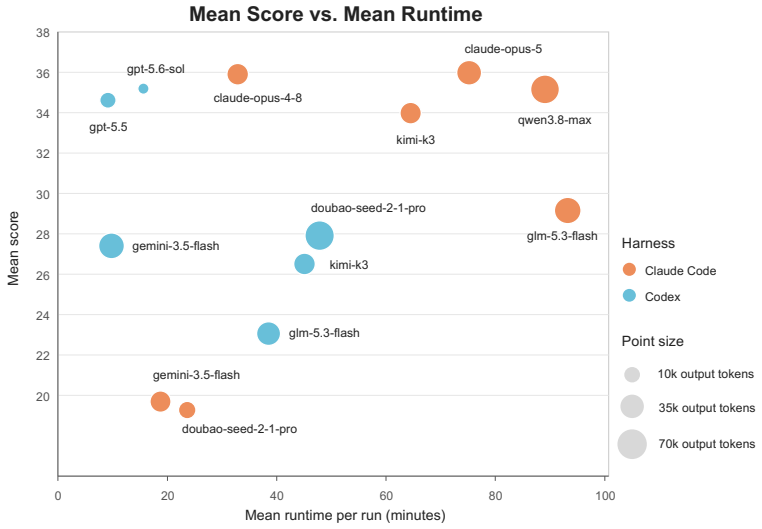


Figure 5: Runtime, token usage, and score profiles across the evaluated harness–model configurations.

This analysis was intended to make the score gaps in Sections 4.3 and 4.4 interpretable. We therefore focused on failure modes that changed the final decision, invalidated the submitted artifact, or made the workflow non-reproducible. Minor presentation errors were not treated as primary failure causes unless they prevented the grader from verifying the result.

Table 3: Common failure modes in low-scoring BRIDGE-Bench runs. The table summarizes what each task required and how the agent's response failed.

Failure mode	Example task	What went wrong
Missing final deliverable	Identify and rank potential drug combinations for leukaemia using public drug databases.	The agent stopped at planning or exploratory analysis and did not submit a clear ranked list of small-molecule pairs.
Wrong task object	Analyze an RNA sequence from a FASTA file and explain its possible effects on ribosome structure and translation.	The agent did not identify the sequence as LD/LM32Cs1C1 snoRNA and missed key mechanistic evidence, causing the explanation to drift away from the intended biological object.
Proxy or substitute data	Analyze the interaction between Gymnemic acid I and AKT1, including binding pocket, binding energy, and key residues.	Some runs replaced the required ligand or protein with simplified or unrelated proxy structures, making the reported binding energies and residue interactions invalid.
Skipped workflow	Predict synergistic drug combinations using drug-combination screening data together with molecular and target features.	Instead of using the required evidence sources and matching workflow, many responses recommended familiar clinical combinations that sounded plausible but did not match the scoring criteria.
Non-reproducible image quantification	Count ATRIP–ATR PLA foci per cell in microscopy images and submit CSV, JSON, method, and script outputs.	Some runs counted the wrong image region, averaged across multiple nuclei, or hard-coded results, making the analysis unreliable and non-reproducible.
Invalid logic	Perform retrosynthetic analysis by identifying the target SMILES, key reactants, bond disconnections, and reaction types.	The agent often recognized broad scaffolds or reaction classes, but used an incorrect target structure, wrong reactants, or a route that did not match the expected bond-disconnection logic.

These patterns suggest that the main problem is not one specific missing skill. Rather, BRIDGE-Bench reveals that agents often fail when they must connect scientific understanding with correct workflow execution and disciplined reporting. For example, in mechanism tasks, if the agent identifies the wrong molecule, target, or pathway at the start, all later reasoning becomes unreliable. In docking, screening, image-analysis, and retrosynthesis tasks, the agent may still produce scien-

tifically plausible text, but this can hide serious errors, such as using the wrong input, skipping a required scoring step, or generating outputs that cannot be reproduced. Such errors are particularly important in drug discovery, where downstream decisions depend on precise information about the molecule, target, assay setting, and required output format.

The six failure categories point to different priorities for improving future agents. No-deliverable and hard-coding failures suggest the need for stronger submission validation and file-state checks. Screening and retrosynthesis failures require workflow-level constraints that ensure agents follow the expected evidence path instead of replacing it with a plausible-sounding answer.

This also shows why aggregate scores are not enough. Two systems can achieve similar final scores while failing in very different ways. Reporting failure modes alongside scores therefore gives a more useful picture of agent limitations and highlights concrete targets for improvement, including better task-object grounding, stricter artifact validation, checks against proxy data, and reproducible traces for image and chemistry workflows.

5 Conclusion

We introduced BRIDGE-Bench, a benchmark designed to evaluate LLM agents on real bottlenecks in drug discovery and development. Rather than testing isolated molecular tasks or single pipeline steps, each BRIDGE-Bench task is framed as a concrete development problem with supporting evidence, decision requirements, downstream consequences, and explicit scoring criteria. The benchmark spans the drug-discovery pipeline, including target biology, hit and lead discovery, ADME, QSP, retrosynthesis, ADC design, drug delivery, CMC/manufacturing, and wet-lab planning.

Using a shared protocol, we evaluated 13 harness–model configurations on 38 tasks. Average scores ranged from 19.3 to 36.0. The best-performing configuration was Claude Code with `claude-opus-5`. These small gaps among the top configurations suggest that several current agents can reach similar overall performance. However, the long-tailed score distribution and many low-scoring runs show that no system is yet consistently reliable across the full benchmark.

BRIDGE-Bench also has limitations. The current dataset contains 38 expert-curated tasks and should be viewed as a focused stress test rather than an exhaustive map of drug discovery. Because of evaluation cost, each task was run once for each harness–model configuration, so the current results do not capture run-to-run variation within the same system. The benchmark also cannot fully represent proprietary industrial workflows, prospective experimental validation, or the full diversity of therapeutic modalities.

The results also show that a single average score does not fully describe agent capability. Performance varied strongly by task type. Chemistry reasoning and structured calculation tasks were generally easier, while reconstruction of biological mechanism, image quantification, and drug-combination screening with complete database workflows were much harder. Stronger configurations were better at handling tasks that combined scientific reasoning with strict artifact requirements, whereas weaker configurations often showed narrow strengths that did not transfer to multi-step evidence integration.

Harness choice also affected both scores and failure modes. Claude Code and Codex had similar overall averages, but the same backbone model could perform quite differently depending on the harness. Codex tended to help with structured execution, tabular outputs, and constrained calculations, while Claude Code more often supported research planning and long-chain evidence organization. Runtime and artifact counts were only weakly related to final score, suggesting that more tokens or more files do not necessarily mean better performance.

Together, these findings suggest that drug-discovery agent evaluation should move beyond single-number leaderboards. BRIDGE-Bench allows us to see not only how well agents score, but also where they fail, including missing deliverables, proxy data use, incorrect object identification, skipped screening workflows, non-reproducible image quantification, and invalid logic. BRIDGE-Bench provides a practical testbed for evaluating whether LLM agents can support realistic, evidence-grounded decisions across the drug-discovery and development pipeline.

Ethics Statement

The main contribution of this work lies in constructing BRIDGE-Bench, a benchmark for systematically evaluating large language model (LLM) agents on realistic drug discovery and development tasks. The benchmark questions are derived from publicly available scientific sources, legally accessible databases, and expert-designed or experiment-inspired scenarios. When tasks are inspired by real research settings, project identifiers, institution names, compound names, target names, batch identifiers, and other potentially sensitive details are removed, abstracted, or anonymized. No proprietary industrial dataset is released as part of the benchmark.

BRIDGE-Bench and the associated experiments are intended solely for academic research and evaluation. The benchmark is not designed for direct deployment in clinical, regulatory, or high-stakes industrial decision-making. By focusing on transparent benchmark construction, expert-guided rubrics, and responsible assessment of agent capabilities and failure modes, we aim to support the development of safer and more reliable AI systems for drug discovery and development.

Artificial intelligence use statement

No artificial intelligence (AI) systems were used to design the benchmark questions, define the benchmark tasks, and construct the reference answers and the scoring rubrics. AI only assisted quality control checks. Final acceptance and rubric decisions were made by human experts. AI systems were used as evaluated models in the benchmark experiments. Generative AI tools were also used to assist with language polishing and clarity improvements during manuscript preparation. All scientific content, benchmark design decisions, experimental interpretations, and final manuscript text were reviewed and approved by the authors, who remain fully responsible for the submitted work.

References

Daniil A. Boiko, Robert MacKnight, Ben Kline, and Gabe Gomes. Autonomous chemical research with large language models. *Nature*, 624(7992):570–578, 2023. doi: 10.1038/s41586-023-06792-0.

Andres M. Bran, Sam Cox, Oliver Schilter, Carlo Baldassari, Andrew D. White, and Philippe Schwaller. Augmenting large language models with chemistry tools. *Nature Machine Intelligence*, 6(5):525–535, 2024. doi: 10.1038/s42256-024-00832-8.

Nathan Brown, Marco Fiscato, Marwin H. S. Segler, and Alain C. Vaucher. GuacaMol: Benchmarking models for de novo molecular design. *Journal of Chemical Information and Modeling*, 59(3): 1096–1108, 2019. doi: 10.1021/acs.jcim.8b00839.

Yin Fang, Xiaozhuan Liang, Ningyu Zhang, Kangwei Liu, Rui Huang, Zhuo Chen, Xiaohui Fan, and Huajun Chen. Mol-Instructions: A large-scale biomolecular instruction dataset for large language models. In *International Conference on Learning Representations (ICLR)*, 2024.

GHDDI AI Lab. DrugBench. `https://github.com/GHDDI-AILab/DrugBench`, 2025. Chain-of-thought multiple-choice benchmark derived from PubMed abstracts; manuscript under review.

Kevin Han, Renfei Zhang, Kathy Wei, Hamed Mahdavi, Niloofar Mireshghallah, and Amir Barati Farimani. SMDD-Bench: Can LLMs solve real-world small molecule drug design tasks? *arXiv preprint arXiv:2605.21740*, 2026.

Kexin Huang, Tianfan Fu, Wenhao Gao, Yue Zhao, Yusuf Roohani, Jure Leskovec, Connor W. Coley, Cao Xiao, Jimeng Sun, and Marinka Zitnik. Therapeutics data commons: Machine learning datasets and tasks for drug discovery and development. In *Advances in Neural Information Processing Systems (NeurIPS) Datasets and Benchmarks Track*, 2021.

Jie Li, Baiming Chen, Zhiyang Zou, Rumin Zhang, Sheng Ding, and Jinjiang Guo. DMPKBench: A multi-modal benchmark for evaluating LLMs and agents in drug discovery DMPK tasks. In *NeurIPS 2025 Workshop: AI for Science*, 2025.

Xingyu Lu, He Cao, Zijing Liu, Shengyuan Bai, Leqing Chen, Yuan Yao, Hai-Tao Zheng, and Yu Li. MoleculeQA: A dataset to evaluate factual accuracy in molecular comprehension. In *Findings of the Association for Computational Linguistics: EMNLP 2024*, pp. 3769–3789, 2024. doi: 10.18653/v1/2024.findings-emnlp.216.

Adrian Mirza, Nawaf Alampara, Sreekanth Kunchapu, Martiño Ríos-García, Benedict Emoekabu, Aswanth Krishnan, Tanya Gupta, Mara Schilling-Wilhelmi, Macjonathan Okereke, Anagha Aneesh, et al. A framework for evaluating the chemical knowledge and reasoning abilities of large language models against the expertise of chemists. *Nature Chemistry*, 17(7):1027–1034, 2025. doi: 10.1038/s41557-025-01815-x.

Daniil Polykovskiy, Alexander Zhebrak, Benjamin Sanchez-Lengeling, Sergey Golovanov, Oktai Tatanov, Stanislav Belyaev, Rauf Kurbanov, Aleksey Artamonov, Vladimir Aladinskiy, Mark Veselov, et al. Molecular sets (MOSES): A benchmarking platform for molecular generation models. *Frontiers in Pharmacology*, 11:565644, 2020. doi: 10.3389/fphar.2020.565644.

Scale AI and Phylo. DrugDiscoveryBench. `https://github.com/scaleapi/DrugDiscoveryBench`, 2026. Open-source benchmark repository; no accompanying publication.

Zhenqin Wu, Bharath Ramsundar, Evan N. Feinberg, Joseph Gomes, Caleb Geniesse, Aneesh S. Pappu, Karl Leswing, and Vijay Pande. MoleculeNet: A benchmark for molecular machine learning. *Chemical Science*, 9(2):513–530, 2018. doi: 10.1039/C7SC02664A.

A Task List

Table 4 lists the benchmark questions included in BRIDGE-Bench. Each row gives a one-sentence description of the scientific or analytical task.

Table 4: One-sentence summary of BRIDGE-Bench questions.

Question	One-sentence summary
Question 1	Identify a target molecule from a molecular structure figure and design its retrosynthetic route, reactants, and key reaction mechanism.
Question 2	Analyze an RNA FASTA sequence and explain how the RNA may affect ribosome structure, assembly, or translation function.
Question 3	Use public drug-combination screening data to identify and rank potential drug combinations for leukaemia.
Question 4	Predict and rank candidate drug combinations for prostate cancer using drug-combination and disease-relevant data.
Question 5	Quantify ATRIP–ATR interaction foci per cell from PLA microscopy images and assess the effects of olaparib and elesclomol–copper.
Question 6	Analyze tetramer compositions after mixed refolding of wild-type and mutant streptavidin and design a separation and purification strategy.
Question 7	Screen candidate therapeutic targets of Gymnema sylvestre for diabetes using compound, disease-target, and protein databases.
Question 8	Use public drug-combination screening data to identify and rank potentially synergistic drug combinations for ovarian cancer.
Question 9	Predict candidate drug-combination regimens for glioma using drug-combination screening data and relevant databases.
Question 10	Screen and rank breast-cancer-relevant candidate drug combinations from public drug-combination screening data.
Question 11	Count drug-loaded particles within a specified size range from SEM images and explain the image-interpretation criteria.
Question 12	Convert a target molecular structure image into SMILES and provide a feasible retrosynthetic disconnection, synthetic route, and mechanism analysis.
Question 13	Use remdesivir in vitro potency, microsomal stability, and protein-binding data to estimate EC50/EC90, CLint, fu, and PK advancement criteria.
Question 14	Quantify RAD51 signal under different siRNA conditions from RAD51/DAPI fluorescence images and interpret implications for DNA repair.

Question	One-sentence summary
Question 15	Analyze the potential binding pocket, interacting residues, docking score, and pharmacological meaning of Gymnemaside VI binding to AKT1.
Question 16	Use public drug-combination screening data to identify and rank candidate drug combinations for lung cancer.
Question 17	Propose retrosynthetic disconnections, key intermediates, reaction conditions, and side-reaction risks for a given target chemical structure.
Question 18	Explain gel-image differences after T7 exonuclease treatment of single DNA and DNA-library PCR products and design downstream experiments.
Question 19	Count the number and average diameter of PEGylated metal-oxide nanoparticles in three TEM image batches.
Question 20	Use public drug-combination screening data to identify and rank potentially synergistic drug combinations for pancreatic cancer.
Question 21	Build a DAR prediction model from ADC bioconjugation data and predict DAR values for two specified linker–payload systems.
Question 22	Use drug-combination screening data to identify and rank candidate drug combinations for renal-cell carcinoma.
Question 23	Analyze the structural source, binding mode, key residues, and interaction strength of Gymnemic acid I binding to AKT1.
Question 24	Predict and rank candidate drug combinations for melanoma from public drug-combination screening data.
Question 25	Analyze the potential binding mode, key interaction sites, and validation strategy for Stigmasterol binding to AKT1.
Question 26	Use public drug-combination screening data to identify and rank potentially synergistic drug combinations for colorectal cancer.
Question 27	Calibrate three parameters in a blinded virus–host dynamics SBML model using 16 observations and predict viral-load decay in the held-out set.
Question 28	Calibrate 30 parameters in a hepatic-lobule PBPK–metabolism model and extrapolate drug distribution and metabolism time courses under 19 held-out conditions.
Question 29	Build a receptor-dimerization and nuclear-translocation SBML model from a textual mechanism description and observations, then complete observation mapping and parameter calibration.
Question 30	Build a ventricular action-potential model from multi-ion-channel blockade data and free therapeutic concentrations for 12 anonymized compounds, then predict TdP risk categories.
Question 31	Starting from fixed target CSF1R, screen a molecular pool for oral, selective, developable small-molecule inhibitors for tenosynovial giant cell tumor and provide candidate-level evidence chains.
Question 32	Starting from fixed target KLKB1, screen reversible and selective small-molecule inhibitors suitable for acute on-demand oral treatment of hereditary angioedema.
Question 33	For EGFR C797S triple-mutant resistant NSCLC, screen reversible or allosteric small molecules that do not depend on covalent Cys797 binding, using multiple evidence sources rather than a single docking result.
Question 34	Screen candidate molecules targeting Menin for KMT2A-rearranged leukaemia and evaluate retention of activity against wild-type and M327I/G331R/T349M resistance mutants.
Question 35	Starting from target DPP1/CTSC, screen small-molecule candidates that inhibit neutrophil serine protease activation for non-cystic-fibrosis bronchiectasis.
Question 36	Screen candidate drugs for methotrexate-inadequate-response rheumatoid arthritis according to a predefined JAK-subtype pharmacology profile and verify activity on the JAK–STAT cytokine pathway.
Question 37	Starting from target DPP4, screen highly selective oral inhibitors for type 2 diabetes while considering a therapeutic position with low hypoglycaemia risk.
Question 38	Screen brain-penetrant 5-HT2A antagonists or inverse agonists for persistent negative symptoms in schizophrenia, evaluating functional mode, binding predictions, 5-HT2B agonism risk, and D2 affinity risk.