DRUGHARNESS: AN EVOLVABLE AGENT HARNESS FOR COMPUTATIONAL DRUG DISCOVERY

Anonymous authors
Paper under double-blind review

ABSTRACT

Computational drug discovery requires agents to compose scientific operations and adapt their strategies to changing evidence and constraints. We introduce DrugHarness, an evolvable agent harness that couples composable scientific execution with conditional skill induction. Its execution environment integrates over 80 scientific operations, biomedical resources, and task-specific code while preserving associations among molecular candidates, results, and artifacts. From execution trajectories and outcome feedback, DrugHarness induces action hypotheses specifying observable applicability conditions, recommended actions, and supporting evidence. These hypotheses are translated into candidate skills and evaluated by an independent review agent through source verification and targeted execution trials when needed. Admitted skills guide subsequent tasks, whose feedback drives revisions to their content and scope. Evaluation on MolBench and SMDD-Bench demonstrates capabilities across molecular screening, optimization, and constrained design. Without evolution, DrugHarness exceeds reported MolClaw-CC results by 10.8 percentage points in affinity-comparison accuracy and 55% in mean top-three active hits on MolBench. In retrospective evaluation on skill-development tasks, evolution increases Qwen's affinity-comparison accuracy from 62.2% to 81.1% and optimization success from 79.5% to 87.2%. In a prospective HPK1 study, DrugHarness achieves a more favorable median docking score than Biomni among system-specific candidates. A synthesized compound achieves a cellular IC50 of 14.19 nM, a 2.64-fold potency improvement over the best input reference, with activity competitive with two selected compounds designed by experienced medicinal chemists. These results support conditional skill evolution for scientific agents and demonstrate a workflow connecting computational drug design with experimental validation.

1 INTRODUCTION

Drug discovery requires coordinating evidence and computation across molecular screening, optimization, and design. Experimental activity records, molecular representations, and property or binding predictors provide complementary information for candidate selection (Liu et al., 2025; RDKit contributors; Swanson et al., 2024; Passaro et al., 2025). Turning these capabilities into a solution requires a sequence of decisions: which evidence to retrieve, which candidates to modify, which calculations to perform, and when a set of molecules satisfies the objective. Recent agent systems have begun to automate these multi-step tasks, making the organization of scientific actions and the reuse of effective strategies central questions for autonomous drug discovery (Averly et al., 2025; Wang et al., 2026).

Progress spans three connected directions. *Tool and environment integration* makes scientific resources accessible through specialized interfaces, code-based composition, and reusable infrastructure (Bran et al., 2024; Huang et al., 2026; Gao et al., 2025b). *Workflow orchestration* organizes these operations through domain skills, specialist collaboration, and stateful processes that maintain intermediate results and guide subsequent actions (Zhang et al., 2026a; Gao et al., 2025a; Cao et al., 2026). *Experience and skill reuse* brings earlier solutions back into execution: prior designs inform new molecular proposals, historical tool trajectories guide lead optimization, and accumulated scientific experience supports later reasoning and tool selection (Gao et al., 2025a; Li et al., 2026;

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.

Tang et al., 2025; 2026b). Together, these directions establish a basis for agents that both execute scientific workflows and accumulate guidance from their use.

Despite this progress, turning heterogeneous scientific experience into dependable action guidance presents three coupled challenges. **(1) Grounding a recommendation in the right evidence.** An activity measurement, a predicted property, an agent's interpretation, and a final task score answer different questions; their objects, endpoints, and observation times determine which action claims they support. **(2) Preserving applicability across tasks.** A useful source-task action may become inappropriate when the candidate state or remaining constraints change. For example, a set with adequate molecular properties but insufficient diversity may benefit from broader structural exploration rather than further optimization of similar candidates. **(3) Establishing executable and revisable guidance.** A plausible recommendation may rest on an unchecked scientific premise or an operation that behaves differently from its description. Its critical assumptions and execution path need active examination, and subsequent task feedback must inform revisions as new evidence becomes available. These challenges call for a learning process that connects scientific evidence, observable conditions, and active scientific review to the continuing use of a skill.

We introduce **DrugHarness**, a unified system for composable scientific execution and conditional skill learning. Its scientific environment provides over 80 scientific operations across ten capability categories, connecting tools, biomedical resources, and code while retaining associations among candidates, results, and artifacts. To ground reusable guidance, the learner compares these experiences and forms action hypotheses with explicit sources and applicability conditions. It translates each hypothesis into a skill whose conditions the executor assesses against the new task's visible state. An independent review agent examines evidence, applicability, and executability, using the same scientific tools and code environment for targeted checks when needed. Admitted skills become available to subsequent tasks, whose trajectories and native outcome feedback support further revision. Review records also allow rejected or deferred proposals to inform later changes. This design connects scientific judgment with the formation, use, and revision of action strategies.

We evaluate DrugHarness on two benchmarks and a prospective molecular design study. On MolBench, DrugHarness without evolution achieves 91.9% affinity-comparison accuracy and 1.24 mean Hits@3, exceeding the reported MolClaw-CC results by 10.8 percentage points and 55%, respectively. On SMDD-Bench, it satisfies all evaluation criteria in 24 of 40 lead-optimization tasks and supports structurally constrained design (Tables 1 and 2). In retrospective evaluation on skill-development tasks, frozen learned skills increase Qwen's affinity-comparison accuracy from 62.2% to 81.1% and optimization success from 79.5% to 87.2% relative to historical runs without skills (Table 3). In prospective HPK1 design, DrugHarness achieves a more favorable median docking score than Biomni among system-specific candidates (Figure 2). A synthesized compound achieves a Jurkat p-SLP76 IC_{50} of 14.19 nM, a 2.64-fold potency improvement over the best input reference, with activity competitive with two selected compounds independently designed by experienced medicinal chemists (Section 4.4). These results demonstrate scientific execution capabilities, in-domain benefits from skill evolution, and experimentally validated molecular design.

Our main contributions are:

- **Composable execution for molecular discovery.** DrugHarness integrates over 80 scientific operations, biomedical resources, and task-specific code, preserving associations among candidates, results, and artifacts to support adaptive scientific workflows. We will release the implementation, tool interfaces, and environment configurations.
- **Evidence-reviewed evolution of conditional skills.** DrugHarness induces action hypotheses with explicit applicability conditions and supporting evidence. An independent agent reviews candidate skills through source checks and targeted scientific trials when needed, while subsequent execution feedback refines their content and scope.
- **Evaluation from benchmarks to prospective validation.** We assess execution on two benchmarks, skill evolution through retrospective in-domain evaluation, and prospective HPK1 design through computational comparison with Biomni and experimental evaluation alongside reference and selected expert-designed compounds.

2 Related Work

Scientific agents for drug discovery. Scientific agents organize capabilities through tool augmentation, code-based execution, and reusable environments, as illustrated by ChemCrow, Biomni, and ToolUniverse (Bran et al., 2024; Huang et al., 2026; Gao et al., 2025b). SciToolAgent further represents tool dependencies and compatibility in a knowledge graph for tool-chain planning (Ding et al., 2025). Drug-discovery workflows use hierarchical domain skills, specialist collaboration, or stateful skill graphs to coordinate research stages (Zhang et al., 2026a; Gao et al., 2025a; Cao et al., 2026); CAi and LIDDiA organize candidate evidence and feedback-driven molecular exploration (Wang et al., 2026; Averly et al., 2025). Experience already participates in these workflows: PharmAgents summarizes earlier designs, and TRACE retrieves optimization trajectories to guide tool sequencing (Gao et al., 2025a; Li et al., 2026).

Experience-driven skill learning. Experience can be distilled into rules and examples, context-aware guidelines, or consolidated skills (Zhao et al., 2024; Fu et al., 2024; Ni et al., 2026). ACE maintains guidance incrementally, while SkillOpt and WikiSkill combine skill revision with execution-based selection and persistent learning state (Zhang et al., 2026b; Yang et al., 2026; Tang et al., 2026a). Concurrent work on SkillCommit examines how replay-supported behavior governs skill consolidation and scope expansion (He & Yang, 2026). Scientific applications learn different objects: ChemAgent maintains chemical subproblem solutions, DrugSAGE reuses experience in property-prediction model development, and SciToolAgent-Evo acquires scientific tool chains and situation-dependent selection guidance (Tang et al., 2025; Zhang et al., 2026c; Tang et al., 2026b). These studies provide complementary mechanisms for experience reuse across reasoning, modeling, and tool-use tasks.

3 Method

3.1 System Overview and Scientific Execution Environment

DrugHarness connects scientific execution, experience, and reusable guidance in a single loop (Figure 1). An agent pursues a task's objective and constraints through scientific actions; the resulting experiences support the induction of conditional action hypotheses and candidate skills. An independent review agent examines each candidate's evidence and executability, using targeted scientific trials when needed. Admitted skills guide subsequent tasks, whose execution feedback informs further revision of the guidance and its applicability.

DrugHarness provides over 80 agent-accessible scientific operations across ten capability categories, integrating biomedical resources, specialized computational tools, and a scientific Python environment (Table 6). These capabilities span evidence retrieval, target and structure preparation, molecular generation, docking, property assessment, editing, optimization, and retrosynthetic analysis. Resources such as BindingDB, RDKit, ADMET-AI, and Boltz-2 support complementary scientific decisions (Liu et al., 2025; RDKit contributors; Swanson et al., 2024; Passaro et al., 2025). The agent combines tool calls with task-specific code to analyze results and continue from intermediate artifacts. DrugHarness retains associations among scientific objects, computational conditions, sources, and outputs across these actions. The runtime supplies the action loop; trajectories, final deliverables, and permitted outcome feedback provide the material for skill learning.

3.2 Inducing Conditional Scientific Strategies

Similar actions can serve different purposes across tasks, so learning preserves the conditions under which an action occurs. An experience is

$$e_i = (x_i, \tau_i, y_i, f_i), \qquad \tau_i = \big((a_{i,j}, o_{i,j})\big)_{j=1}^{T_i}. \tag{1}$$

Here x_i contains the objective, constraints, and public inputs; $a_{i,j}$ and $o_{i,j}$ are the action and visible observation at within-task step j; T_i is the number of recorded steps; y_i is the actual deliverable; and f_i is permitted outcome feedback. Contemporaneous observations, the executor's interpretations, and subsequent evaluations remain distinguishable, including missing or incomplete results.

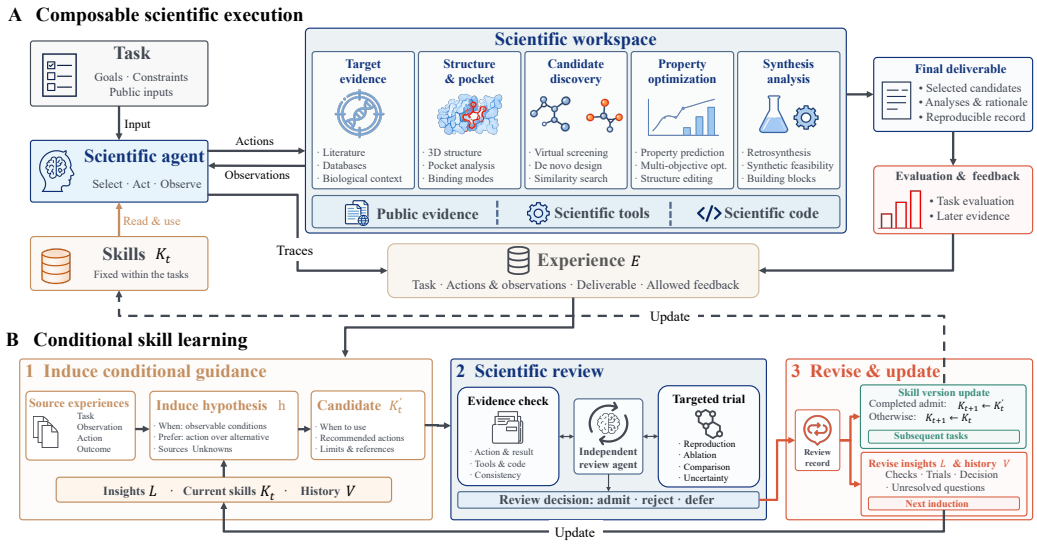


Figure 1: **Overview of DrugHarness.** DrugHarness integrates composable scientific execution with conditional skill learning. The agent combines public evidence, scientific tools, and code within a shared scientific workspace to produce task deliverables. Execution trajectories and permitted feedback provide experiences from which the system derives guidance conditioned on task context. An independent review agent evaluates candidate skills and, when needed, conducts targeted trials in the same workspace. Approved skills update the skill version used in subsequent tasks, while review records inform revisions to insights and history. Scientific capabilities can be flexibly combined rather than following a fixed sequence of stages.

The basic object induced from experience is an action hypothesis,

$$h = (g, c, a_0, a, R_h, U_h). \tag{2}$$

The goal g specifies what the action should advance; c describes conditions observable by the executor; a_0 and a are the current or alternative action and the recommended action. Sources R_h support or limit the hypothesis, and U_h records unresolved information. Conditions explain why action priorities may change. As an illustrative scenario, consider constructing a molecular set whose members satisfy individual property requirements but lack set diversity. A learner could hypothesize that expanding structural sources should take priority over further property optimization of similar candidates, and organize supporting and limiting evidence from source experiences.

Researchers specify the task scope, quality objectives, material permissions, execution environment, and native evaluation rules, denoted by Ω. Within this scope, the learner maintains experiences E, revisable insights L, deployed skills K, and revision history V. At learning round t, it reads a batch $B_t \subseteq E$ and combines it with these states:

$$(\widetilde{L}_t, h_t) = \mathrm{Induce}(B_t, L_t, K_t, V_t; \Omega, \mathcal{B}_t). \tag{3}$$

The reading and generation allowance $\mathcal{B}_t$ is bounded by the total development budget $\mathcal{B}$. For one candidate branch, the outputs are revised insights $\widetilde{L}_t$ and a selected hypothesis h_t, with $h_t = \varnothing$ when no reusable modification is identified. Induce is an LLM task that first identifies recurring decision problems, then compares the relevant scientific objects, endpoints, methods, and evidence states. It may reread sources within Ω to resolve a specific question, distinguishing observations, interpretations, and outcomes.

Compatible conditions and consistent recommendations allow support to be combined; different conditions retain separate scopes. Conflicting evidence motivates narrowing or splitting a recommendation when justified, or retaining the disagreement. If existing skills already cover the decision, induction can update insights alone. Algorithm 2 in Appendix C details this process. Its hypothesis remains an action choice to be tested, with its sources and scope carried into candidate review. Multiple directions may be constructed and reviewed in parallel as separate branches, each retaining its parent version and modification intent.

3.3 Candidate Skill Construction and Evidence Review

Candidate construction translates a hypothesis into guidance the executor can read. Each candidate branch applies one explicit behavioral modification intent to the current skills:

$$K'_t = \mathrm{Edit}(K_t, h_t). \tag{4}$$

Edit expresses the goal, conditions, recommended action, and limitations as skill guidance, retaining necessary evidence in accompanying resources. Names and descriptions indicate relevant situations; the body provides the information needed to assess conditions and adjust actions. In the molecular-set example, the skill preserves the premises that individual requirements are met and set diversity is insufficient. Additions, narrowing, replacement, merging, splitting, or deactivation all serve the same modification intent. The revised guidance together with the remaining skills forms the complete candidate version K'_t.

The candidate specifies the decision situations it is expected to affect and observable checks of the intended change. An independent review agent examines its source evidence, applicability conditions, and executable instructions. The reviewer can retrieve original sources, inspect the available interfaces, and use the same scientific tools and code environment as the task executor. When a material question requires computation, it performs a targeted local trial and retains the actual inputs, outputs, and limitations as review evidence. The completed review returns

$$\rho_t = \mathrm{Review}(K'_t, h_t, R_{h_t}; \Omega) \in \{\mathrm{admit}, \mathrm{reject}, \mathrm{defer}\}. \tag{5}$$

Review checks source fidelity, scientific premises, scope, condition observability, and executability. A clear conflict with critical facts or constraints supports rejection; insufficient evidence supports deferral. Admission makes the reviewed version eligible for subsequent task use. Local trials address specific scientific or operational questions; task-level benefit is assessed from subsequent execution and the effect studies described in Section 4.3. Counterexamples, unresolved information, and completed trial records remain associated with the candidate.

3.4 Task-time Use and Feedback-driven Revision

Each task binds a specified skill version. The executor selects relevant skills from their names and descriptions, reads their bodies and resources as needed, and relates the guidance to the currently visible state s_j. Using h and c for a hypothesis and its condition as expressed in the skill, applicability is assessed as

$$\widehat{\chi}_j(h) = \mathrm{Assess}(c; s_j) \in \{\mathrm{supported}, \mathrm{contradicted}, \mathrm{unknown}\}. \tag{6}$$

This is the executor's semantic judgment. Supported conditions allow a recommendation to inform action selection; contradicted conditions lead the executor to other available grounds for action. When a critical condition is unknown, the executor decides whether obtaining more information is useful to the current task. In the molecular-set example, current property and diversity evaluations inform whether to expand structural sources; an unevaluated critical property leaves the premise unknown. Progressive loading provides access to guidance, while the agent decides whether to use, adapt, or leave a recommendation unused.

For a candidate branch with a specified parent version, a completed admission decision controls the update:

$$K_{t+1} = \begin{cases} K'_t, & \text{review completed and } \rho_t = \mathrm{admit}, \\ K_t, & \text{otherwise.} \end{cases} \tag{7}$$

Rejection, deferral, an incomplete review, or a batch without a candidate preserves the current version. Each admitted version retains its complete content, parent identity, and review evidence. Separately admitted branches remain identifiable versions; their admission does not automatically combine their modifications. A task binds an explicit version that stays fixed throughout that execution. If deployed guidance contains a verifiable source misattribution, unit error, or explicit specification conflict under its claimed conditions, the system records the evidence and withdraws that content from future provision. Replacement guidance follows candidate construction and independent review.

Regardless of deployment changes, the round's proposals and feedback update insights and history:

$$(L_{t+1}, V_{t+1}) = \mathrm{Revise}(\widetilde{L}_t, V_t, \zeta_t). \tag{8}$$

The record ζ_t contains the proposal, completed source checks and local trials, admission decision or incomplete status, version history, and unresolved information. Rejections can supply counterexamples and deferrals can identify missing evidence, revising insights even when no new version is deployed. As subsequent tasks actually run, their trajectories, deliverables, and permitted native outcome feedback are added to E and become available to later batches. Review updates use the records already produced in that branch; later task feedback enters learning only after it exists. These materials support further changes until no actionable basis remains or the resource limit is reached.

Appendix C provides Algorithms 1 and 2, which summarize the learning loop and its induction subroutine. A program coordinates LLM tasks for condition comparison, candidate writing, and independent scientific review, while managing permissions, budgets, trial records, version history, and the provision of admitted skills. The resulting guidance remains revisable as scientific experience accumulates.

4 EXPERIMENTS

We evaluate three aspects of DrugHarness: scientific execution without learned skills, execution with experience-derived skills, and prospective evaluation of designed molecules. Model and runtime comparisons support the execution analysis, while task trajectories clarify how learned guidance enters scientific decisions.

4.1 EXPERIMENTAL SETUP

System configurations. *DrugHarness (w/o Evol.)* uses the full scientific environment without cross-task learned skills; its main configuration is Codex with GPT-5.6 Terra/xhigh. Model comparisons use Codex with Qwen3.7-Max, and runtime comparisons use Hermes with Terra/xhigh. Both comparisons cover all 190 MolBench tasks; runtime comparisons additionally cover 202 SMDD Types 1–5 tasks. *DrugHarness (Evol.)* uses admitted, frozen skills: Qwen3.8-Max/xhigh performs automatic refinement and independent review. DrugHarness uses a default execution budget of one hour per benchmark task and six hours for the HPK1 design task.

Evaluation tasks and metrics. We use the two benchmarks for complementary purposes. SMDD-Bench evaluates DrugHarness on diverse constrained molecular-design problems and serves as a broad test of scientific execution (Han et al., 2026). MolBench provides a fine-grained decomposition of molecular decision-making and operations, spanning property filtering, similarity filtering, affinity comparison, virtual screening, molecular editing, and optimization, together with task-specific quantitative feedback (Zhang et al., 2026a). We therefore use MolBench as the primary testbed for studying experience-driven evolution, where such fine-grained feedback enables systematic analysis of skill induction, use, and revision. SMDD-Bench uses task success rates; we evaluate a fixed 40-task lead-optimization (LO) subset spanning 25 targets and the complete remaining families. MolBench metrics include accuracy, mean active hits among the top three candidates (Hits@3), and positive-improvement success rate. Task breakdowns and metric definitions appear in Appendix A. The prospective HPK1 task evaluates analog design under collaborator-provided references and constraints (Section 4.4).

4.2 MAIN RESULTS

Overall Performance. We use Codex with GPT-5.6 Terra as the main configuration to evaluate scientific execution without evolution. On SMDD-Bench, DrugHarness achieves 28.0% success in interaction point discovery and 7.7% in scaffold hopping, while solving 24 of 40 LO tasks (Table 1). On MolBench, affinity accuracy reaches 91.9%, 10.8 percentage points above MolClaw-CC, and mean Hits@3 reaches 1.24 versus 0.80. Screening accuracy is 98.0%, all 39 editing tasks are solved, and 37 of 39 optimization tasks yield positive improvement (94.9%; Table 2). Together, these results establish support for candidate judgments, precise molecular operations, and designs with joint structural and property constraints.

Backbone and Runtime Comparisons. We vary the model within Codex to assess backbone dependence, and compare Hermes with Codex under Terra to assess runtime effects. With Qwen3.7-

Table 1: **Results on SMDD-Bench.** Success rates (%); original-harness baselines are reported by Han et al. (2026). DrugHarness is evaluated without evolution using two runtimes.

Method / Runtime	Model	2D Pharm.↑	IP↑	SH↑	LO↑	FA↑
SMDD harness	Claude Sonnet 4.6	**28.0**	0.0	3.8	53.5	0.0
	GPT-5.4	12.0	0.0	3.8	57.6	**1.7**
	Gemini 3.1 Pro	20.0	4.0	0.0	55.6	1.7
	Kimi K2.5 Thinking	12.0	0.0	1.9	43.5	0.0
	DeepSeek V3.2	8.0	0.0	3.8	34.7	0.0
	MiniMax M2.7	16.0	0.0	1.9	27.1	0.0
	Qwen3.5 397B A17B	4.0	0.0	1.9	40.0	0.0
DrugHarness / Hermes (w/o Evol.)	GPT5.6-Terra	12.0	20.0	3.8	37.5	0.0
DrugHarness / Codex (w/o Evol.)	GPT5.6-Terra	12.0	**28.0**	**7.7**	**60.0**	0.0

Table 2: **MolBench results across systems, models, and runtimes.** External results: Zhang et al. (2026a); Wang et al. (2026). Definitions and aggregation: Appendix A.1.

Method	MolClaw comparison group			CAi	**DrugHarness** (w/o Evol.)		
Model	Sonnet 4.6	Sonnet 4.6	Qwen3.7-Max	Qwen3.7-Max	Qwen3.7-Max	GPT5.6-Terra	GPT5.6-Terra
Runtime	Claude Code†	Claude Code	N/R	CAi	Codex	Hermes	Codex
Screening							
Accuracy (%)↑	98.0	98.0	18.0	96.0	96.0	98.0	**98.0**
Precision (%)↑	—	—	32.0	98.0	98.0	98.0	**100.0**
Recall (%)↑	—	—	44.5	97.0	97.0	98.0	**99.0**
F1 (%)↑	—	—	33.8	97.3	97.3	98.0	**99.3**
Specificity (%)↑	—	—	44.0	98.0	100.0	98.0	**100.0**
Validity (%)↑	—	—	78.0	100.0	98.0	100.0	**100.0**
Affinity comparison							
Accuracy (%)↑	51.4	81.1	54.1	54.1	62.2	83.8	**91.9**
Valid choice (%)↑	—	—	100.0	100.0	100.0	100.0	**100.0**
Virtual screening							
Hit@3 (%)↑	—	—	52.0	40.0	60.0	48.0	**60.0**
Mean Hits@3↑	0.56	0.80	0.76	0.56	0.72	0.92	**1.24**
Top-3 rate (%)↑	—	—	25.3	18.7	24.0	30.7	**41.3**
Mean GT rank↓	—	—	56.23	55.49	29.05	27.69	**17.27**
Hit count↑	—	—	13/25	10/25	15/25	12/25	**15/25**
Editing							
Add (%)↑	—	—	70.0	100.0	100.0	100.0	**100.0**
Delete (%)↑	—	—	88.9	100.0	100.0	100.0	**100.0**
Substitute (%)↑	—	—	95.0	95.0	95.0	100.0	**100.0**
Overall (%)↑	97.4	100.0	87.2	97.4	97.4	100.0	**100.0**
Validity (%)↑	—	—	100.0	100.0	100.0	100.0	**100.0**
Optimization							
Mean Δ↑	0.8660	1.7240	0.1352	0.2263	0.6054	0.4530	0.5988
Success rate (%)↑	—	**100.0**	48.7	79.5	79.5	84.6	94.9
Valid SMILES (%)↑	—	—	97.4	100.0	87.2	**100.0**	94.9
Scaffold hard (%)↑	—	—	**71.8**	46.2	38.5	59.0	35.9
Scaffold soft (%)↑	—	—	**88.9**	76.5	61.8	77.0	63.6

N/R: Runtime not reported. †: Claude Code without MolClaw skills.

Max, DrugHarness attains 62.2% affinity accuracy versus 54.1% for both CAi and MolClaw, and 60.0% Hit@3 versus 40.0% and 52.0%, respectively; screening (96.0%) and editing (97.4%) match

Table 3: **Experience-driven evolution on MolBench.** Codex/Qwen3.7-Max with historical no-skill runs and retrospective in-domain evaluation using frozen domain-specific skills. Proportions are percentages. Complete metrics: Appendix A.2.

Setting	Screening	Affinity	Virtual screening			Editing	Optimization	
	Acc.↑	Acc.↑	Hit@3↑	Mean Hits@3↑	Mean GT rank↓	Acc.↑	SR↑	Valid JSON↑
w/o Evol.	96.0	62.2	**60.0**	**0.72**	29.05	**97.4**	79.5	87.2
Evol.	**100.0**	**81.1**	48.0	0.64	**26.01**	**97.4**	**87.2**	**100.0**

CAi (Wang et al., 2026). With Terra, both runtimes achieve 98.0% screening and 100% editing accuracy, while Codex leads Hermes in affinity accuracy (91.9% versus 83.8%), mean Hits@3 (1.24 versus 0.92), and LO success (24/40 versus 15/40). Codex/Terra is the strongest overall configuration tested. Nevertheless, competitive results under a shared Qwen backbone and effective execution through both runtimes indicate that DrugHarness's performance is not explained solely by backbone choice.

4.3 Experience-driven Evolution

Beyond backbone and runtime choices, we examine whether execution experience can improve DrugHarness through reusable skills. Domain-specific guidance raises screening accuracy to 100%, affinity accuracy from 62.2% to 81.1%, and optimization success from 79.5% to 87.2%, while retaining 97.4% editing accuracy (Table 3). All 39 optimization tasks produce valid JSON submissions, compared with 34 in historical no-skill runs.

Learned guidance makes retrieved evidence actionable. In affinity task ms2-029, both skill versions led the executor to materialize the same collection of 933 records. The earlier guidance prompted text searches and structural reasoning; the revision prompted RDKit structure matching to associate candidates with activity records and distinguish K_i from IC50. The decision changed from incorrect to correct. Across the domain, the learned skill corrects nine historical errors and retains twenty-one previously correct answers.

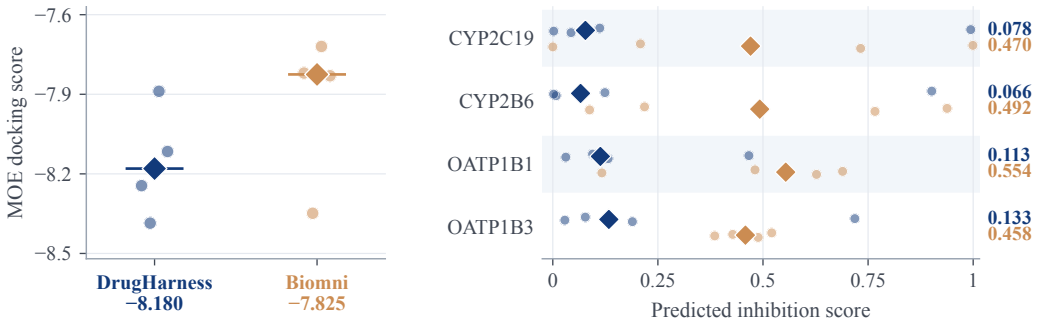
Optimization experience yields guidance for checking the actual deliverable. From formatting failures and their repairs, the learner forms instructions to read back and validate the saved answer before submission. The independent reviewer tests this procedure in a targeted execution trial before deployment; subsequent runs produce valid JSON on all 39 tasks, with positive property improvement on 34. The learned skills achieve 30/37 correct affinity comparisons and 34/39 successful optimizations, approaching the fixed researcher-written references of 33/37 and 35/39, respectively.

Virtual screening illustrates feedback-driven skill revision. The initial skill emphasized recovery of missing jobs, but runs exposed missing submissions and incomplete candidate coverage. DrugHarness used these trajectories and feedback to revise the skill. With the frozen revision, complete coverage of the original 60 candidates increased from 14/25 to 25/25 tasks. Revised runs achieved a top-three hit on 12 tasks and a mean reference rank of 26.01 versus the historical no-skill 29.05.

4.4 Prospective Validation of Designed Molecules

We design human HPK1 (MAP4K1) inhibitors from four experimentally characterized references, aiming to improve Jurkat p-SLP76 inhibition over the best reference (IC50: 37.47 nM). Designs preserve the shared core and specified double-bond geometry, modifying only terminal amide nitrogen substituents.[1] DrugHarness uses Codex/Qwen3.8-Max; Biomni uses its default configuration.

[1]Candidate molecules are reported using coded identifiers; their chemical structures are withheld for confidentiality.



(a) MOE docking

(b) Selected inhibition predictions

Figure 2: **Computational evaluation of HPK1 candidates.** Four system-specific structures per group remain after excluding six shared structures from ten evaluated entries each. Colors: ■ DrugHarness; ■ Biomni. Symbols: molecules (●), medians (◆). Lower is favorable. The four inhibition endpoints were selected post hoc for lower DrugHarness medians; its median hERG inhibition was higher (0.949 versus 0.663) and QED lower (0.3645 versus 0.4085).

In Silico Evaluation of Designed Molecules. Among the four system-specific candidates per evaluated set, DrugHarness achieves a more favorable median MOE docking score than Biomni (-8.180 versus -7.825; Figure 2(a)). It also has lower median inhibition predictions for CYP2C19 (0.078 versus 0.470), CYP2B6 (0.066 versus 0.492), OATP1B1 (0.113 versus 0.554), and OATP1B3 (0.133 versus 0.458; Figure 2(b)).

Experimental Validation of Designed Molecules. A DrugHarness-designed compound was synthesized and experimentally evaluated by our medicinal chemistry collaborators. It achieved a Jurkat p-SLP76 IC50 of 14.19 nM, lower than those of all four reference compounds (37.47–227.77 nM), together with 96.92% HPK1 inhibition at 1 nM. On the same design task, the DrugHarness-designed compound showed competitive biochemical and cellular activity compared with two selected compounds independently designed by medicinal chemists with over five years of experience (Figure 3).

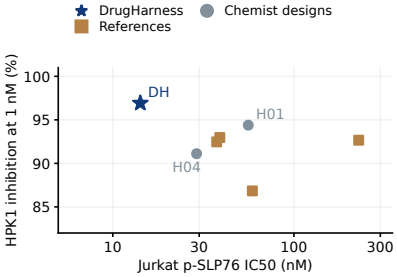


Figure 3: **Comparison with selected chemist designs.** Lower cellular IC50 and higher 1 nM HPK1 inhibition indicate stronger activity.

5 Limitations and Conclusion

Limitations. Systematic comparisons of runtime, computational cost, and resource requirements against other agent frameworks and human experts remain open. Skill evolution is evaluated on a single benchmark with a limited number of tasks under a retrospective, in-domain setting; its effectiveness at larger scales and generalization to unseen targets and task families remain to be established. Experimental validation is also limited in scale and coverage of developability properties. Future work will expand evaluation across benchmarks and targets, conduct held-out assessments, and broaden experimental validation under matched conditions.

Conclusion. DrugHarness provides a mechanism for evolving scientific agent strategies through conditional skill learning. It turns execution experience into reusable guidance with explicit applicability conditions, independent scientific review, and feedback-driven revision. Evaluation on two benchmarks demonstrates broad molecular task capabilities, while retrospective results on skill-development tasks show improvements in screening, affinity comparison, and optimization. A prospective design study further links computational predictions to experimental activity, yielding a synthesized compound with improved cellular potency over all input references and activity competitive with selected expert-designed compounds. These findings support evidence-reviewed skill evolution as a promising approach to improving scientific agents and motivate broader evaluation of its generalization and experimental utility.

AI USE STATEMENT

AI assistants supported manuscript drafting and language polishing, research discussion and feedback, and code development and verification. All AI-assisted content was reviewed by the authors, who take full responsibility for the accuracy, originality, and integrity of the manuscript.

ETHICS STATEMENT

DrugHarness is intended for drug-discovery research. Proprietary molecular structures are withheld to protect collaborator confidentiality. Computational predictions and in vitro activity measurements do not establish clinical safety or efficacy.

REPRODUCIBILITY STATEMENT

We will open-source the DrugHarness implementation, evaluation code, data-processing procedures, and environment and experiment configurations to support reproducibility. Proprietary molecular structures used in the HPK1 validation study are excluded from the release due to collaborator confidentiality.

REFERENCES

Reza Averly, Frazier N Baker, Ian A Watson, and Xia Ning. LIDDIA: Language-based intelligent drug discovery agent. In *Proceedings of the 2025 Conference on Empirical Methods in Natural Language Processing*, pp. 12004–12028, 2025.

Cédric Bouysset and Sébastien Fiorucci. ProLIF: a library to encode molecular interactions as fingerprints. *Journal of Cheminformatics*, 13(1):72, 2021. doi: 10.1186/s13321-021-00548-6. URL https://doi.org/10.1186/s13321-021-00548-6.

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. URL https://doi.org/10.1038/s42256-024-00832-8.

Annalisa Buniello, Daniel Suveges, Carlos Cruz-Castillo, Manuel Bernal Llinares, Helena Cornu, Irene Lopez, Kirill Tsukanov, Juan María Roldán-Romero, Chintan Mehta, Luca Fumis, et al. Open Targets Platform: facilitating therapeutic hypotheses building in drug discovery. *Nucleic Acids Research*, 53(D1):D1467–D1475, 2025. doi: 10.1093/nar/gkae1128. URL https://doi.org/10.1093/nar/gkae1128.

Stephen K. Burley, Charmi Bhikadiya, Chunxiao Bi, Sebastian Bittrich, Henry Chao, Li Chen, Paul A. Craig, Gregg V. Crichlow, Kenneth Dalenberg, Jose M. Duarte, et al. RCSB Protein Data bank: Tools for visualizing and understanding biological macromolecules in 3D. *Protein Science*, 31(12):e4482, 2022. doi: 10.1002/pro.4482. URL https://doi.org/10.1002/pro.4482.

Duanhua Cao, Geng Chen, Jiaxin Jiang, Jie Yu, Runze Zhang, Mingan Chen, Wei Zhang, Lifan Chen, Feisheng Zhong, Yingying Zhang, et al. Generic protein–ligand interaction scoring by integrating physical prior knowledge and data augmentation modelling. *Nature Machine Intelligence*, 6(6):688–700, 2024. doi: 10.1038/s42256-024-00849-z. URL https://doi.org/10.1038/s42256-024-00849-z.

He Cao, Siyu Liu, Fan Zhang, Zijing Liu, Hao Li, Bin Feng, Shengyuan Bai, Leqing Chen, Kai Xie, and Yu Li. Mozi: Governed Autonomy for Drug Discovery LLM Agents, 2026. URL https://arxiv.org/abs/2603.03655v1.

Ethan Cerami, Jianjiong Gao, Ugur Dogrusoz, Benjamin E. Gross, Selcuk Onur Sumer, Bülent Arman Aksoy, Anders Jacobsen, Caitlin J. Byrne, Michael L. Heuer, Erik Larsson, et al. The cBio Cancer Genomics Portal: An Open Platform for Exploring Multidimensional Cancer Genomics Data. *Cancer Discovery*, 2(5):401–404, 2012. doi: 10.1158/2159-8290.CD-12-0095. URL https://doi.org/10.1158/2159-8290.CD-12-0095.

John S. Delaney. ESOL: Estimating Aqueous Solubility Directly from Molecular Structure. *Journal of Chemical Information and Computer Sciences*, 44(3):1000–1005, 2004. doi: 10.1021/ci034243x. URL https://doi.org/10.1021/ci034243x.

Ji Ding, Shidi Tang, Zheming Mei, Lingyue Wang, Qinqin Huang, Haifeng Hu, Ming Ling, and Jiansheng Wu. Vina-GPU 2.0: Further Accelerating AutoDock Vina and Its Derivatives with Graphics Processing Units. *Journal of Chemical Information and Modeling*, 63(7):1982–1998, 2023. doi: 10.1021/acs.jcim.2c01504. URL https://doi.org/10.1021/acs.jcim.2c01504.

Keyan Ding, Jing Yu, Junjie Huang, Yuchen Yang, Qiang Zhang, and Huajun Chen. SciToolAgent: a knowledge-graph-driven scientific agent for multitool integration. *Nature Computational Science*, 5(10):962–972, 2025. doi: 10.1038/s43588-025-00849-y. URL https://doi.org/10.1038/s43588-025-00849-y.

Ian Dunn and David R. Koes. FlowMol3: flow matching for 3D de novo small-molecule generation. *Digital Discovery*, 5(5):2052–2066, 2026. doi: 10.1039/D5DD00363F. URL https://doi.org/10.1039/D5DD00363F.

Jerome Eberhardt, Diogo Santos-Martins, Andreas F. Tillack, and Stefano Forli. AutoDock Vina 1.2.0: New Docking Methods, Expanded Force Field, and Python Bindings. *Journal of Chemical Information and Modeling*, 61(8):3891–3898, 2021. doi: 10.1021/acs.jcim.1c00203. URL https://doi.org/10.1021/acs.jcim.1c00203.

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

Vendy Fialková, Jiaxi Zhao, Kostas Papadopoulos, Ola Engkvist, Esben Jannik Bjerrum, Thierry Kogej, and Atanas Patronov. LibINVENT: Reaction-based Generative Scaffold Decoration for in Silico Library Design. *Journal of Chemical Information and Modeling*, 62(9):2046–2063, 2022. doi: 10.1021/acs.jcim.1c00469. URL https://doi.org/10.1021/acs.jcim.1c00469.

Yao Fu, Dong-Ki Kim, Jaekyeom Kim, Sungryull Sohn, Lajanugen Logeswaran, Kyunghoon Bae, and Honglak Lee. AutoGuide: Automated generation and selection of context-aware guidelines for large language model agents. In A. Globerson, L. Mackey, D. Belgrave, A. Fan, U. Paquet, J. Tomczak, and C. Zhang (eds.), *Advances in Neural Information Processing Systems*, volume 37, pp. 119919–119948. Curran Associates, Inc., 2024. doi: 10.52202/079017-3811. URL https://proceedings.neurips.cc/paper_files/paper/2024/file/d8efbb5dd415974eb095c3f06bff1f48-Paper-Conference.pdf.

Bowen Gao, Yanwen Huang, Yiqiao Liu, Wenxuan Xie, Wei-Ying Ma, Ya-Qin Zhang, and Yanyan Lan. PharmAgents: Building a Virtual Pharma with Large Language Model Agents, 2025a. URL https://arxiv.org/abs/2503.22164v2.

Jianjiong Gao, Bülent Arman Aksoy, Ugur Dogrusoz, Gideon Dresdner, Benjamin Gross, S. Onur Sumer, Yichao Sun, Anders Jacobsen, Rileen Sinha, Erik Larsson, et al. Integrative Analysis of Complex Cancer Genomics and Clinical Profiles Using the cBioPortal. *Science Signaling*, 6(269), 2013. doi: 10.1126/scisignal.2004088. URL https://doi.org/10.1126/scisignal.2004088.

Shanghua Gao, Richard Zhu, Pengwei Sui, Zhenglun Kong, Sufian Aldogom, Yepeng Huang, Ayush Noori, Reza Shamji, Krishna Parvataneni, Theodoros Tsiligkaridis, and Marinka Zitnik. ToolUniverse: An open platform for democratizing AI scientists, 2025b. URL https://arxiv.org/abs/2509.23426v3.

Mahdi Ghorbani, Leo Gendelev, Paul Beroza, and Michael J. Keiser. Autoregressive fragment-based diffusion for pocket-aware ligand design. *arXiv*, 2024. doi: 10.48550/arXiv.2401.05370. URL https://doi.org/10.48550/arXiv.2401.05370.

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.

Jiazhen He, Eva Nittinger, Christian Tyrchan, Werngard Czechtizky, Atanas Patronov, Esben Jannik Bjerrum, and Ola Engkvist. Transformer-based molecular optimization beyond matched molecular pairs. *Journal of Cheminformatics*, 14(1):18, 2022. doi: 10.1186/s13321-022-00599-3. URL https://doi.org/10.1186/s13321-022-00599-3.

Yu He and Weikai Yang. SkillCommit: Evolving Agent Skills through Behaviorally Validated Scope Expansion, 2026. URL https://arxiv.org/abs/2608.15165v1.

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 *Proceedings of the Neural Information Processing Systems Track on Datasets and Benchmarks*, 2021. URL https://datasets-benchmarks-proceedings.neurips.cc/paper/2021/hash/4c56ff4ce4aaf9573aa5dff913df997a-Abstract-round1.html.

Kexin Huang, Serena Zhang, Hanchen Wang, Yuanhao Qu, Yingzhou Lu, Ryan Li, Yusuf Roohani, Lin Qiu, Shiyi Cao, Gavin Li, Junze Zhang, Di Yin, Rick Wierenga, Deniz Kavi, Sherry Liu, Tianwei She, Shruti Marwaha, Jennefer N. Carter, Xin Zhou, Matthew T. Wheeler, Jonathan A. Bernstein, Mengdi Wang, Peng He, Jingtian Zhou, Michael P. Snyder, Le Cong, Aviv Regev, and Jure Leskovec. Autonomous biomedical research with an artificial intelligence agent. *Science*, 393(6813):eadz4351, 2026. doi: 10.1126/science.adz4351. URL https://doi.org/10.1126/science.adz4351.

Sunghwan Kim, Jie Chen, Tiejun Cheng, Asta Gindulyte, Jia He, Siqian He, Qingliang Li, Benjamin A Shoemaker, Paul A Thiessen, Bo Yu, et al. PubChem 2025 update. *Nucleic Acids Research*, 53(D1):D1516–D1525, 2025. doi: 10.1093/nar/gkae1059. URL https://doi.org/10.1093/nar/gkae1059.

KinHub. KinHub: Human kinases. https://www.kinhub.org/kinases.html. Human kinase annotation resource.

Radoslav Krivák and David Hoksza. P2Rank: machine learning based tool for rapid and accurate prediction of ligand binding sites from protein structure. *Journal of Cheminformatics*, 10(1):39, 2018. doi: 10.1186/s13321-018-0285-8. URL https://doi.org/10.1186/s13321-018-0285-8.

Melissa J Landrum, Shanmuga Chitipiralla, Kuljeet Kaur, Garth Brown, Chao Chen, Jennifer Hart, Douglas Hoffman, Wonhee Jang, Chunlei Liu, Zenith Maddipatla, et al. ClinVar: updates to support classifications of both germline and somatic variants. *Nucleic Acids Research*, 53(D1): D1313–D1321, 2025. doi: 10.1093/nar/gkae1090. URL https://doi.org/10.1093/nar/gkae1090.

Catherine L Lawson, Helen M Berman, Li Chen, Brinda Vallat, and Craig L Zirbel. The Nucleic Acid Knowledgebase: a new portal for 3D structural information about nucleic acids. *Nucleic Acids Research*, 52(D1):D245–D254, 2024. doi: 10.1093/nar/gkad957. URL https://doi.org/10.1093/nar/gkad957.

Vincent Le Guilloux, Peter Schmidtke, and Pierre Tuffery. Fpocket: An open source platform for ligand pocket detection. *BMC Bioinformatics*, 10(1):168, 2009. doi: 10.1186/1471-2105-10-168. URL https://doi.org/10.1186/1471-2105-10-168.

Hao Li, He Cao, Bin Feng, Yanjun Shao, Xiangru Tang, Zhiyuan Yan, Yonghong Tian, Li Yuan, and Yu Li. Beyond chemical QA: Evaluating LLM's chemical reasoning with modular chemical operations. In D. Belgrave, C. Zhang, H. Lin, R. Pascanu, P. Koniusz, M. Ghassemi, and N. Chen (eds.), *Advances in Neural Information Processing Systems, Datasets and Benchmarks Track*, volume 38. Curran Associates, Inc., 2025. doi: 10.52202/085713-5122.

Lingxiao Li, Haobo Zhang, Ruohao Fan, Bin Chen, and Jiayu Zhou. Molecular Lead Optimization via Agentic Tool Planning. In *Proceedings of the 32nd ACM SIGKDD Conference on Knowledge Discovery and Data Mining V.2*, pp. 11328–11338, 2026. doi: 10.1145/3770855.3819017. URL https://doi.org/10.1145/3770855.3819017.

Tiqing Liu, Linda Hwang, Stephen K Burley, Carmen I Nitsche, Christopher Southan, W Patrick Walters, and Michael K Gilson. BindingDB in 2024: a FAIR knowledgebase of protein-small molecule binding data. *Nucleic Acids Research*, 53(D1):D1633–D1644, 2025. doi: 10.1093/nar/gkae1075. URL https://doi.org/10.1093/nar/gkae1075.

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

Milot Mirdita, Konstantin Schütze, Yoshitaka Moriwaki, Lim Heo, Sergey Ovchinnikov, and Martin Steinegger. ColabFold: making protein folding accessible to all. *Nature Methods*, 19(6): 679–682, 2022. doi: 10.1038/s41592-022-01488-1. URL https://doi.org/10.1038/s41592-022-01488-1.

Jingwei Ni, Yihao Liu, Xinpeng Liu, Yutao Sun, Mengyu Zhou, Pengyu Cheng, Dexin Wang, Erchao Zhao, Xiaoxi Jiang, and Guanjun Jiang. Trace2Skill: Distill Trajectory-Local Lessons into Transferable Agent Skills, 2026. URL https://arxiv.org/abs/2603.25158v5. Reading version: 2603.25158v5.

Saro Passaro, Gabriele Corso, Jeremy Wohlwend, Mateo Reveiz, Stephan Thaler, Vignesh Ram Somnath, Noah Getz, Tally Portnoi, Julien Roy, Hannes Stark, et al. Boltz-2: Towards Accurate and Efficient Binding Affinity Prediction. *bioRxiv*, 2025. doi: 10.1101/2025.06.14.659707. URL https://doi.org/10.1101/2025.06.14.659707.

PDBFixer contributors. PDBFixer. URL https://github.com/openmm/pdbfixer. Software resource.

RDKit contributors. RDKit: Open-source cheminformatics. URL https://www.rdkit.org. Software resource.

Summer Rosonovski, Maria Levchenko, Rajat Bhatnagar, Umamageswari Chandrasekaran, Lynne Faulk, Islam Hassan, Matt Jeffryes, Syed Irtaza Mubashar, Maaly Nassar, Madhumiethaa Jayaprabha Palanisamy, et al. Europe PMC in 2023. *Nucleic Acids Research*, 52(D1):D1668–D1676, 2024. doi: 10.1093/nar/gkad1085. URL https://doi.org/10.1093/nar/gkad1085.

Lakshidaa Saigiridharan, Alan Kai Hassen, Helen Lai, Paula Torren-Peraire, Ola Engkvist, and Samuel Genheden. AiZynthFinder 4.0: developments based on learnings from 3 years of industrial application. *Journal of Cheminformatics*, 16(1):57, 2024. doi: 10.1186/s13321-024-00860-x. URL https://doi.org/10.1186/s13321-024-00860-x.

Diogo Santos-Martins, Yiran He, Jerome Eberhardt, Parnika Sharma, Niccolò Bruciaferri, Matthew Holcomb, Manuel A. Llanos, Althea Hansel-Harris, Allison Pearl Barkdull, Andreas Frank Tillack, et al. Meeko: Molecule Parametrization and Software Interoperability for Docking and Beyond. *Journal of Chemical Information and Modeling*, 65(24):13045–13050, 2025. doi: 10.1021/acs.jcim.5c02271. URL https://doi.org/10.1021/acs.jcim.5c02271.

Philipp Schake, Sarah Naomi Bolz, Katja Linnemann, and Michael Schroeder. PLIP 2025: introducing protein–protein interactions to the protein–ligand interaction profiler. *Nucleic Acids Research*, 53(W1):W463–W465, 2025. doi: 10.1093/nar/gkaf361. URL https://doi.org/10.1093/nar/gkaf361.

Arne Schneuing, Charles Harris, Yuanqi Du, Kieran Didi, Arian Jamasb, Ilia Igashov, Weitao Du, Carla Gomes, Tom L. Blundell, Pietro Lio, et al. Structure-based drug design with equivariant diffusion models. *Nature Computational Science*, 4(12):899–909, 2024. doi: 10.1038/s43588-024-00737-x. URL https://doi.org/10.1038/s43588-024-00737-x.

Kyle Swanson, Parker Walther, Jeremy Leitz, Souhrid Mukherjee, Joseph C Wu, Rabindra V Shivnaraine, and James Zou. ADMET-AI: a machine learning ADMET platform for evaluation of large-scale chemical libraries. *Bioinformatics*, 40(7):btae416, 2024. doi: 10.1093/bioinformatics/btae416. URL https://doi.org/10.1093/bioinformatics/btae416.

Damian Szklarczyk, Rebecca Kirsch, Mikaela Koutrouli, Katerina Nastou, Farrokh Mehryary, Radja Hachilif, Annika L Gable, Tao Fang, Nadezhda T Doncheva, Sampo Pyysalo, et al. The STRING database in 2023: protein–protein association networks and functional enrichment analyses for any sequenced genome of interest. *Nucleic Acids Research*, 51(D1):D638–D646, 2023. doi: 10.1093/nar/gkac1000. URL https://doi.org/10.1093/nar/gkac1000.

Liyan Tang, Cyrus Rashtchian, Chun-Sung Ferng, Andrew Tomkins, Da-Cheng Juan, and Tu Vu. WikiSkill: Compiling Agent Experience into Persistent Knowledge for Skill Evolution, 2026a. URL https://arxiv.org/abs/2608.27454v1. Reading version: 2608.27454v1.

Shidi Tang, Ji Ding, Xiangyu Zhu, Zheng Wang, Haitao Zhao, and Jiansheng Wu. Vina-GPU 2.1: Towards Further Optimizing Docking Speed and Precision of AutoDock Vina and Its Derivatives. *IEEE/ACM Transactions on Computational Biology and Bioinformatics*, 21(6):2382–2393, 2024. doi: 10.1109/TCBB.2024.3467127. URL https://doi.org/10.1109/TCBB.2024.3467127.

Xiangru Tang, Tianyu Hu, Muyang Ye, Yanjun Shao, Xunjian Yin, Siru Ouyang, Wangchunshu Zhou, Pan Lu, Zhuosheng Zhang, Yilun Zhao, Arman Cohan, and Mark Gerstein. ChemAgent: Self-updating library in large language models improves chemical reasoning. In Y. Yue, A. Garg, N. Peng, F. Sha, and R. Yu (eds.), *International Conference on Learning Representations*, volume 2025, pp. 100922–100958, 2025.

Yuqi Tang, Chenyi Zhou, Libin Wang, Keyan Ding, Qiang Zhang, and Huajun Chen. SciToolAgent-Evo: An Ontology-Aware Self-Evolving Agent for Open-World Scientific Tool Acquisition, 2026b. URL https://arxiv.org/abs/2607.28692v1.

The ENCODE Project Consortium. Expanded encyclopaedias of DNA elements in the human and mouse genomes. *Nature*, 583(7818):699–710, 2020. doi: 10.1038/s41586-020-2493-4. URL https://doi.org/10.1038/s41586-020-2493-4.

The GTEx Consortium. The GTEx Consortium atlas of genetic regulatory effects across human tissues. *Science*, 369(6509):1318–1330, 2020. doi: 10.1126/science.aaz1776. URL https://doi.org/10.1126/science.aaz1776.

The UniProt Consortium. UniProt: the Universal Protein Knowledgebase in 2025. *Nucleic Acids Research*, 53(D1):D609–D617, 2025. doi: 10.1093/nar/gkae1010. URL https://doi.org/10.1093/nar/gkae1010.

Oleg Trott and Arthur J. Olson. AutoDock Vina: Improving the speed and accuracy of docking with a new scoring function, efficient optimization, and multithreading. *Journal of Computational Chemistry*, 31(2):455–461, 2010. doi: 10.1002/jcc.21334. URL https://doi.org/10.1002/jcc.21334.

Mihaly Varadi, Damian Bertoni, Paulyna Magana, Urmila Paramval, Ivanna Pidruchna, Malarvizhi Radhakrishnan, Maxim Tsenkov, Sreenath Nair, Milot Mirdita, Jingi Yeo, et al. AlphaFold Protein Structure Database in 2024: providing structure coverage for over 214 million protein sequences. *Nucleic Acids Research*, 52(D1):D368–D375, 2024. doi: 10.1093/nar/gkad1011. URL https://doi.org/10.1093/nar/gkad1011.

Sameer Velankar, José M. Dana, Julius Jacobsen, Glen van Ginkel, Paul J. Gane, Jie Luo, Thomas J. Oldfield, Claire O'Donovan, Maria-Jesus Martin, and Gerard J. Kleywegt. SIFTS: Structure Integration with Function, Taxonomy and Sequences resource. *Nucleic Acids Research*, 41(D1):D483–D489, 2013. doi: 10.1093/nar/gks1258. URL https://doi.org/10.1093/nar/gks1258.

Zhu Wang, Jiangyu Chen, Yingjun Shang, Yuhui Yao, Laiao Lu, Tianfan Fu, and Na Zou. Cai copilot: Reducing operational workload in molecular design through intent-driven agentic workflows. *arXiv preprint arXiv:2608.06961*, 2026.

John D. Westbrook, Chenghua Shao, Zukang Feng, Marina Zhuravleva, Sameer Velankar, and Jasmine Young. The chemical component dictionary: complete descriptions of constituent molecules in experimentally determined 3D macromolecules in the Protein Data Bank. *Bioinformatics*, 31 (8):1274–1278, 2015. doi: 10.1093/bioinformatics/btu789. URL https://doi.org/10.1093/bioinformatics/btu789.

Jeremy Wohlwend, Gabriele Corso, Saro Passaro, Noah Getz, Mateo Reveiz, Ken Leidal, Wojtek Swiderski, Liam Atkinson, Tally Portnoi, Itamar Chinn, et al. Boltz-1 Democratizing Biomolecular Interaction Modeling. *bioRxiv*, 2024. doi: 10.1101/2024.11.19.624167. URL https://doi.org/10.1101/2024.11.19.624167.

Yifan Yang, Ziyang Gong, Weiquan Huang, Qihao Yang, Ziwei Zhou, Zisu Huang, Yan Li, Xuemei Gao, Qi Dai, Bei Liu, Kai Qiu, Yuqing Yang, Dongdong Chen, Xue Yang, and Chong Luo. SkillOpt: Executive Strategy for Self-Evolving Agent Skills, 2026. URL https://arxiv.org/abs/2605.23904v2. Reading version: 2605.23904v2.

Barbara Zdrazil, Eloy Felix, Fiona Hunter, Emma J Manners, James Blackshaw, Sybilla Corbett, Marleen de Veij, Harris Ioannidis, David Mendez Lopez, Juan F Mosquera, et al. The ChEMBL Database in 2023: a drug discovery platform spanning multiple bioactivity data types and time periods. *Nucleic Acids Research*, 52(D1):D1180–D1192, 2024. doi: 10.1093/nar/gkad1004. URL https://doi.org/10.1093/nar/gkad1004.

Lisheng Zhang, Lilong Wang, Xiangyu Sun, Wei Tang, Haoyang Su, Yuehui Qian, Qikui Yang, Qingsong Li, Zhenyu Tang, Haoran Sun, et al. MolClaw: An autonomous agent with hierarchical skills for drug molecule evaluation, screening, and optimization. *bioRxiv*, 2026a. doi: 10.64898/2026.04.03.716272. URL https://www.biorxiv.org/content/10.64898/2026.04.03.716272v1.

Qizheng Zhang, Changran Hu, Shubhangi Upasani, Boyuan Ma, Fenglu Hong, Vamsidhar Kamanuru, Jay Rainton, Chen Wu, Mengmeng Ji, Hanchen Li, Urmish Thakker, James Y Zou, and Kunle Olukotun. Agentic context engineering: Evolving contexts for self-improving language models. In C. Vondrick, B. Hariharan, C. Raffel, L. Pinto, D. Yang, and A. Faust (eds.), *International Conference on Learning Representations*, volume 2026, pp. 86069–86100, 2026b. URL https://proceedings.iclr.cc/paper_files/paper/2026/file/8a94ff6f922d995d7d3f4ebf4143e442-Paper-Conference.pdf.

Yikun Zhang, Xiwei Cheng, Tianyu Liu, Yuanqi Du, and Wengong Jin. DrugSAGE: Self-evolving Agent Experience for Efficient State-of-the-Art Drug Discovery, 2026c. URL https://arxiv.org/abs/2605.15461v1. Reading version: 2605.15461v1.

Andrew Zhao, Daniel Huang, Quentin Xu, Matthieu Lin, Yong-Jin Liu, and Gao Huang. ExpeL: LLM Agents Are Experiential Learners. In *Proceedings of the AAAI Conference on Artificial Intelligence*, volume 38, pp. 19632–19642, 2024.

Ying Zhou, Yintao Zhang, Donghai Zhao, Xinyuan Yu, Xinyi Shen, Yuan Zhou, Shanshan Wang, Yunqing Qiu, Yuzong Chen, and Feng Zhu. TTD: Therapeutic Target Database describing target druggability information. *Nucleic Acids Research*, 52(D1):D1465–D1477, 2024. doi: 10.1093/nar/gkad751. URL https://doi.org/10.1093/nar/gkad751.

A BENCHMARK AND EVALUATION DETAILS

This section summarizes the task definitions and evaluation quantities of the source benchmarks. The baseline values in Tables 2 and 1 are literature-reported results under their respective original protocols.

A.1 MOLBENCH

Task coverage. The 190 MolBench tasks comprise 35 property-filtering, 15 similarity-filtering, 37 affinity-comparison, 25 ranking, 39 editing, and 39 optimization tasks.

Screening. MolBench-MS includes molecular property filtering, binding-affinity comparison, and docking-based screening (Zhang et al., 2026a). Property filtering selects molecules satisfying the supplied constraints from ten candidates. Binding-affinity comparison selects one of two molecules using a target-specific criterion, with experimental K_i defining the reference order. Docking screening ranks 60 candidates per task, including experimentally active and inactive molecules. The source study defines mean Hits@3 as

$$\text{MeanHits@3} = \frac{1}{n}\sum_{i=1}^{n} \left|\text{Top3}(i) \cap A_i\right|, \tag{9}$$

where A_i is the active set for task i.

Combined comparison. Table 2 combines the Sonnet baselines reported in MolClaw with the Qwen baselines reported in CAi and the three DrugHarness NoSkill configurations. The first MolClaw-group column is the reported Claude Code baseline without MolClaw skills.

Table notation. All proportion metrics in the MolBench tables are percentages. Acc., Prec., Spec., and Valid denote accuracy, precision, specificity, and validity; SR denotes success rate. Hit@3 is the percentage of ranking tasks with at least one active molecule in the top three, whereas Mean Hits@3 is the mean active-molecule count. Top-3 rate is Mean Hits@3 divided by three, expressed as a percentage; Count gives successful tasks over the total. Mean GT rank is the mean predicted rank of the reference top-three molecules. In affinity comparison, Valid measures valid candidate choices. In editing, Subst. denotes substitution and Overall aggregates editing operations. Optimization Valid is the valid-SMILES percentage; Hard and Soft denote the two scaffold-retention measures. Percentages are displayed to one decimal place; calculations retain the underlying values.

Editing and optimization. MolBench-MO distinguishes functional-group editing from physicochemical property optimization. Editing accuracy is the fraction of outputs satisfying all requested modifications. Optimization concerns QED, LogP, or LogS and reports both a property-change quantity and success against predefined criteria. The editing experiment contains 39 tasks. Optimization success uses positive native improvement over 39 tasks. Table 4 reports ordinary means of saved per-task gains separately by property.

Table 4: **Property optimization with Codex/Terra.** Valid pairs denote valid scoring pairs; scaffold retention uses the hard criterion.

Property	Tasks	Valid pairs	Positive gain	Mean gain	Scaffold retained
LogP	11	10	10	0.9260	4/11
QED	20	19	19	0.3748	3/20
Solubility	8	8	8	0.7089	7/8

Across the 39 tasks, the hard scaffold-retention rate is 35.90% and mean soft scaffold retention is 0.6360. These quantities describe structural preservation separately from the 94.9% property-improvement success rate.

A.2 COMPLETE EVOLUTION RESULTS

Table 5 reports all 23 MolBench metrics for the Qwen3.7-Max configurations compared in Table 3.

Table 5: **Complete MolBench evolution metrics.** Both configurations use Codex/Qwen3.7-Max. Proportions are percentages; the mean Δ is the descriptive task-wise mean of saved native improvements.

Metric	w/o Evol.	Evol.
Screening		
Accuracy (%)↑	96.0	100.0
Precision (%)↑	98.0	100.0
Recall (%)↑	97.0	100.0
F1 (%)↑	97.3	100.0
Specificity (%)↑	100.0	100.0
Validity (%)↑	98.0	100.0
Affinity comparison		
Accuracy (%)↑	62.2	81.1
Valid choice (%)↑	100.0	97.3
Virtual screening		
Hit@3 (%)↑	60.0	48.0
Mean Hits@3↑	0.72	0.64
Top-3 rate (%)↑	24.0	21.3
Mean GT rank↓	29.05	26.01
Hit count↑	15/25	12/25
Editing		
Add (%)↑	100.0	100.0
Delete (%)↑	100.0	100.0
Substitute (%)↑	95.0	95.0
Overall (%)↑	97.4	97.4
Validity (%)↑	100.0	100.0
Optimization		
Mean Δ↑	0.6054	0.7213
Success rate (%)↑	79.5	87.2
Valid SMILES (%)↑	87.2	100.0
Scaffold hard (%)↑	38.5	48.7
Scaffold soft (%)↑	61.8	69.0

A.3 SMDD-BENCH

Our evaluation includes the complete Types 1–3 and 5 and the 40-task LO subset described below. Table abbreviations are 2D Pharm. (2D pharmacophore identification), IP (interaction point discovery), SH (scaffold hopping), LO (lead optimization), and FA (fragment assembly). The five source families contain 25, 25, 52, 340, and 60 instances, respectively (Han et al., 2026). The overall success rate is the fraction of successful instances across these families, rather than an unweighted average of family-level rates.

2D pharmacophore identification. The input comprises a protein sequence and ten active and ten inactive molecules. The output is a Python predicate that classifies a supplied molecular SMILES string. Evaluation measures recall on hidden active molecules and specificity on hidden inactive molecules.

Interaction point discovery. Given a receptor and pocket specification, the agent submits three typed three-dimensional points. Type-compatible points are matched one-to-one to reference interaction sites using a distance criterion of less than 2.5 Å. Partial credit equals the number of matched points divided by three; success requires all three matches.

Scaffold hopping. The output is a molecule with a different scaffold but preserved binding interactions. The detailed evaluator requires molecular validity, Morgan-fingerprint Tanimoto similarity below 0.5, a scaffold maximum-common-substructure ratio below 0.65, interaction-fingerprint similarity above 0.75, and predicted binding probability above 0.7.

Lead optimization. The agent modifies a reference molecule while satisfying task-specific improvement targets, hold-constant intervals, and hard constraints. Evaluation jointly checks molecular validity, scaffold similarity, the property requirements, and predicted binding conditions. The reference-similarity threshold is at least 0.7. Property directions and tolerances belong to individual task specifications.

LO subset selection. DrugHarness uses a fixed subset of 40 LO tasks spanning 25 targets, with one or two tasks per target. Six previously selected tasks and a designated DHODH task were retained. Additional tasks were selected to cover optimization directions, objective counts, and structural input sizes using public task inputs, with natural task identifiers breaking ties. The subset contains 9/9/9/8/5 tasks with one/two/three/four/five objectives, respectively. Structure sizes comprise 10/18/12 tasks with at most 200, 201–500, and more than 500 ATOM-record residues.

Fragment assembly. One or two spatially placed fragments must be retained within a valid output molecule. Evaluation checks fragment containment, connectivity where required, hard constraints, predicted binding probability above 0.7, affinity within 0.1 log units of the witness reference, and fragment RMSD below 2.0 Å. Topological containment alone is insufficient.

B Scientific Tools and Computational Environment

B.1 Scientific Capabilities

DrugHarness provides a scientific environment spanning biomedical evidence retrieval, molecular design, and computational evaluation. Table 6 groups representative tools and resources by scientific function and summarizes their inputs and outputs. Operations can be invoked independently and combined according to the task, without imposing a fixed sequence of stages.

Table 6: **Scientific capabilities of DrugHarness.** Representative tools and resources supporting over 80 agent-accessible scientific operations are organized into ten capability categories. A tool may support multiple operations; entries summarize capabilities rather than enumerate individual interfaces.

Capability	Tools and resources	Inputs → outputs
Biomedical evidence retrieval	Open Targets (Buniello et al., 2025); cBioPortal (Cerami et al., 2012; Gao et al., 2013); TTD (Zhou et al., 2024); ClinVar (Landrum et al., 2025); GTEx (The GTEx Consortium, 2020); ENCODE (The ENCODE Project Consortium, 2020); STRING (Szklarczyk et al., 2023); Europe PMC (Rosonovski et al., 2024); ClinicalTrials.gov and FDA record interfaces.	Disease, target, variant, or experimental context → associations, variants, expression, functional and network evidence, publications, and study records with provenance.
Target identity and structural data	UniProt (The UniProt Consortium, 2025); RCSB PDB (Burley et al., 2022); SIFTS (Velankar et al., 2013); AlphaFold DB (Varadi et al., 2024); NAKB (nucleic-acid structures) (Lawson et al., 2024); KinHub (KinHub).	Names, sequences, or database identifiers → target identities, sequences, structure candidates and coordinates, chain/residue mappings, and family annotations.
Receptor, ligand, and pocket preparation	PDBFixer (PDBFixer contributors); Meeko (Santos-Martins et al., 2025); RDKit (RDKit contributors); CCD (Westbrook et al., 2015); fpocket (Le Guilloux et al., 2009); P2Rank (Krivák & Hoksza, 2018).	Protein/ligand structures, component selections, and preparation parameters → prepared receptors and ligand states, pocket candidates, docking grids, and input files.
Interaction and pharmacophore analysis	PLIP (Schake et al., 2025); ProLIF (Bouysset & Fiorucci, 2021); RDKit; atomic geometry analysis and co-crystal alignment/clustering.	Complexes, structural ensembles, or labeled molecules → contacts, water-mediated paths, interaction fingerprints, conserved three-dimensional interaction points, and molecular features.
Compound sources and activity evidence	ChEMBL (Zdrazil et al., 2024); BindingDB (Liu et al., 2025); PubChem (Kim et al., 2025); local-library similarity search; Google Patents retrieval by publication number.	Target/compound identifiers, reference molecules, or patent publication numbers → chemical identities, measured activities with assay context, candidate rankings, or patent documents.
Molecule generation and fragment linking	DiffSBDD (Schneuing et al., 2024); FlowMol3 (Dunn & Koes, 2026); AutoFragDiff (Ghorbani et al., 2024); explicit molecular-graph linking.	Generation settings, optional pocket context, or fragments with attachment sites → three-dimensional candidates, fragment mappings, and linked-molecule proposals.

Continued on next page

Table 6 (continued)

Capability	Tools and resources	Inputs → outputs
Docking, rescoring, and binding prediction	AutoDock Vina (Trott & Olson, 2010; Eberhardt et al., 2021); QuickVina2-GPU 2.1 (Tang et al., 2024; Ding et al., 2023); EquiScore (Cao et al., 2024); Boltz-2 (Passaro et al., 2025; Wohlwend et al., 2024); MSA preparation (Mirdita et al., 2022).	Receptors, ligands, grids/pockets, sequences, and required MSAs → poses, rescoring results, predicted complexes, binding-related values, and confidence estimates.
Molecular processing and property assessment	RDKit; ESOL (ChemCoTBench refit) (Delaney, 2004; Li et al., 2025); PyTDC-compatible penalized LogP (Huang et al., 2021; Ertl & Schuffenhauer, 2009); ADMET-AI (Swanson et al., 2024).	SMILES/SDF, candidate sets, and calculation parameters → standardized molecules, physicochemical properties, structural alerts, similarities, solubility, and ADMET predictions.
Molecular editing and optimization proposals	Explicit molecular-graph editing; REINVENT4 (Loeffler et al., 2024); Mol2Mol (He et al., 2022); LibINVENT (Fialková et al., 2022).	Starting molecules, scaffolds, edit requests, or generation settings → edited candidates, scaffold decorations, analogues, and operation-integrity diagnostics.
Retrosynthetic analysis	AiZynthFinder (Saigiridharan et al., 2024).	Target molecules, stock, and search settings → candidate retrosynthetic routes, stock matches, and search results.

C SKILL LEARNING ALGORITHMS

The main loop combines hypothesis induction, independent scientific review, version updates, and revision from subsequent-task feedback. The induction subroutine specifies how scientific conditions and evidence shape a proposed action change.

Algorithm 1 Skill learning and revision from scientific experience

Require: Experiences E, insights L, skills K, history V, scope Ω, budget $\mathcal{B}$
Ensure: Admitted skill versions, revised L, E, and V

for each development batch B_t within Ω and $\mathcal{B}$ **do**
  Incorporate trajectories and permitted native feedback already produced by task runs into E.
  $(\widetilde{L}, h) \leftarrow \mathrm{Induce}(B_t, L, K, V; \Omega, \mathcal{B}_t)$ ▷ One candidate branch
  $L \leftarrow \widetilde{L}$
  if $h = \varnothing$ **then**
    Save L and unresolved records; update V; **continue**.
  end if
  $K' \leftarrow \mathrm{Edit}(K, h)$; retain parent K, full content, sources, and intended change.
  An independent review agent checks sources, conditions, and executable instructions.
  Use the shared scientific tools or code for targeted local trials when needed; retain trial evidence.
  Complete $\mathrm{Review}(K', h, R_h; \Omega)$ and record ρ, or record incomplete status.
  if review completed and $\rho = \mathrm{admit}$ **then**
    Record the admitted version K' for subsequent tasks bound explicitly to that version.
    $K \leftarrow K'$ for this branch.
  else
    Keep K; record rejection, deferral, or incomplete status.
  end if
  Revise L, V using the proposal and review records actually available.
  if the stage stopping condition is met **then**
    break
  end if
end for
Add subsequent task trajectories and permitted native feedback to E when produced.

Multiple proposal directions may be constructed and reviewed in parallel as separate branches. The update above describes one branch; its admission applies to the complete version reviewed, with no automatic merge of separately admitted candidates.

Algorithm 2 Inducing a conditional action hypothesis

Require: Batch B_t, insights L, skills K, history V, scope Ω, allowance $\mathcal{B}_t$
Ensure: Revised insights $\widetilde{L}$ and one hypothesis h or $\varnothing$

1. Identify recurring decision problems within Ω from actions and observations.
2. Separate contemporaneous observations, agent interpretations, and later feedback.
3. Check the scientific objects, endpoints, and result states.
4. Compare decision-relevant conditions; reread permitted sources for specific questions within $\mathcal{B}_t$.
5. Combine support for compatible conditions and consistent recommendations; retain separate scopes for different conditions.
6. For conflicts under comparable conditions, narrow or split guidance when justified; otherwise retain disagreement and uncertainty.
7. Compare with K and V to identify existing coverage and prior revision feedback.
8. Revise L to obtain $\widetilde{L}$; propose (g, c, a_0, a, R, U) for a testable gap, specifying the expected action difference.
9. Return one direction for this candidate branch and retain other directions for separate branches; return $\varnothing$ if no reusable modification is identified.