SCICOMPILER: COMPILING OPEN SCIENTIFIC ARTIFACTS INTO VERIFIED AGENT CAPABILITIES

Anonymous authors
Paper under double-blind review

ABSTRACT

Scientific advances are often released as papers, repositories, and checkpoints rather than as capabilities that scientific agents can directly use and retain, while existing agents still rely largely on fixed tool repertoires. We introduce SCICOMPILER, which discovers artifacts for a given task and compiles them from locked source into callable operations, independent execution validation, and capability registration. Each retained capability binds a scientific operation to its interface, provenance, source, weights, runtime, and execution evidence, allowing subsequent tasks to invoke the same validated computation without repeating acquisition. Across protein design and genomics, SCICOMPILER acquires reusable scientific operations that remain callable on new inputs and across downstream workflows. In genomics, a retained Evo 2 likelihood capability reaches 96.25% regulatory accuracy on reserved chromosomes, compared with 78.59% for the coding baseline, and the same operation is reused across regulatory, motif, and variant tasks. Controlled studies further show that structured MCP invocation preserves scientific outputs while reducing agent requests by 23.4% and worker time by 22.3% in protein tasks. Augmenting capability records with artifact identity and validated execution facts increases reuse recommendations by 20 percentage points. Together, these results show how scientific agents can grow beyond fixed tool inventories by turning research artifacts into validated, reusable capabilities with explicit provenance and execution evidence.

1 INTRODUCTION

Scientific agents coordinate tools, models, and computational workflows to address scientific tasks (Ren et al., 2025; Gao et al., 2025; Ding et al., 2025). Their capabilities typically depend on tool repertoires assembled before deployment (Zhang et al., 2026). Meanwhile, open research continually releases new methods as papers, repositories, and checkpoints (Miao et al., 2026), creating an opportunity for agents to discover, validate, and retain capabilities as new scientific needs arise.

Existing work addresses important parts of this problem. ToolMaker constructs tools from supplied repositories (Wölflein et al., 2025), while Paper2Agent turns supplied papers and code into callable servers (Miao et al., 2026). SciToolAgent-Evo studies ontology-guided acquisition and consolidation of unfamiliar tools (Tang et al., 2026). Related work on tool construction and reuse has further highlighted the importance of separating capability construction from subsequent invocation (Cai et al., 2024). However, these settings do not fully address what is required to turn a newly encountered scientific need into a reliable agent capability. Discovering a relevant artifact or compiling it into a callable interface is not sufficient: the interface may omit task-relevant inputs, expose a computation that does not match the scientific objective, or produce outputs without establishing which source, weights, runtime, and execution path generated them. More generally, a callable tool does not by itself establish that the intended scientific computation has been faithfully preserved and actually executed (Gryk & Ludäscher, 2017).

To address this gap, we introduce SCICOMPILER, a task-driven framework for compiling research artifacts into verified and reusable scientific capabilities. Given a scientific task and the current capability inventory, SCICOMPILER first determines whether the task can be served by an existing capability, requires a new capability, or should be declined. When a new capability is required, the system discovers and assesses candidate research artifacts, verifies and locks their official sources,

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

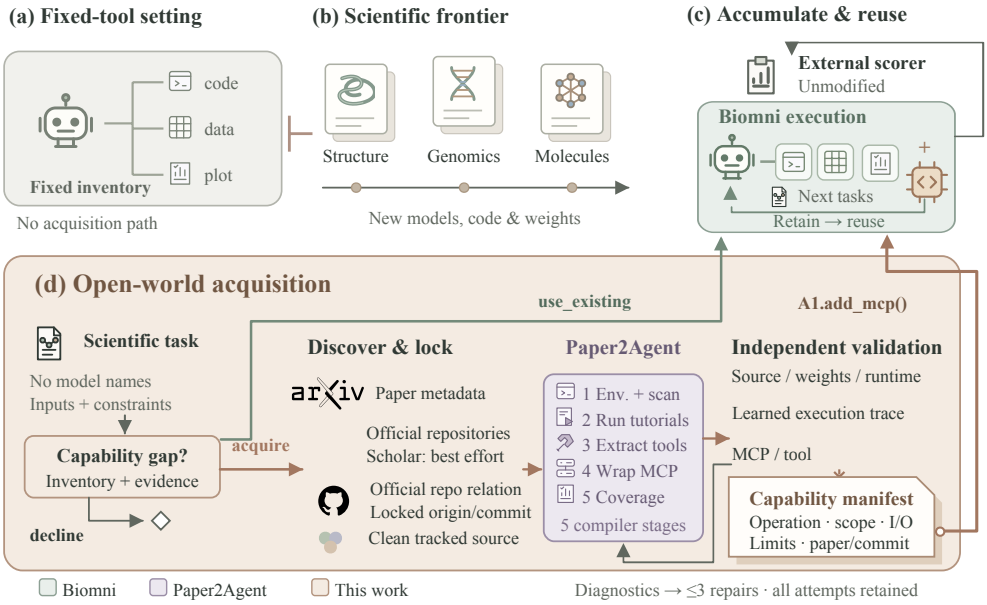

Figure 1: **From open research artifacts to an accumulating scientific repertoire.** Papers, code, and weights supply candidate operations; task requirements and the current inventory motivate acquisition or use. Our controller connects artifact discovery and locking to Paper2Agent's five compilation stages and independent admission. The accepted operation enters Biomni's repertoire with its source, weights, runtime, schema, and evidence, making it available to later tasks. The evaluation combines DNA acquisition-and-use sequences with controlled studies of invocation interfaces and capability records.

and compiles the selected artifact into a callable operation. A separate scientific admission stage then verifies the actual source and weights used at execution time, the runtime and task-specific settings, the intended scientific computation, and the provenance of the resulting output. Capabilities that pass admission are registered together with their interface, runtime, provenance, and execution evidence, allowing subsequent tasks to reuse the same validated computation without repeating discovery and compilation.

We evaluate SCICOMPILER along three complementary dimensions: whether compiled scientific capabilities can support subsequent tasks, how their invocation interfaces affect agent execution, and how validated execution evidence influences later capability selection. Across protein design and genomics, compiled operations remain reusable on new inputs and downstream workflows. In genomics, a retained Evo 2 likelihood capability reaches 96.25% regulatory accuracy on reserved chromosomes, compared with 78.59% for the coding baseline, and the same operation is subsequently reused across regulatory, motif, and variant tasks. In a controlled protein study, MCP-based invocation preserves identical scientific outputs while reducing agent requests by 23.4% and worker time by 22.3%. Finally, augmenting capability records with validated execution evidence increases recommendations to reuse existing capabilities by up to 20 percentage points. Together, these results show that reusable scientific capabilities should retain not only executable computation, but also the interfaces and execution evidence needed for effective future use.

Our contributions can be summarized as follows:

- We introduce SCICOMPILER, a task-driven framework that identifies capability gaps and compiles research artifacts into verified, reusable scientific capabilities through artifact discovery, source locking, executable compilation, independent scientific admission, and capability registration.
- We develop a capability representation and validation protocol that distinguishes callable software from valid scientific execution. Each retained capability binds the scientific

operation to its applicability, interface, source and weights, runtime, provenance, and execution evidence, enabling reliable reuse across subsequent tasks.

- We evaluate retained capabilities across protein design and genomics, showing that compiled operations can support new inputs and downstream workflows. Controlled studies further demonstrate that invocation interfaces affect agent execution cost and that validated execution evidence improves subsequent capability-selection decisions.

2 Related work

Tool use, interfaces, and construction. ReAct interleaves reasoning and actions (Yao et al., 2023), while Toolformer learns API use (Schick et al., 2023). CodeAct makes executable code the action space (Wang et al., 2024b); SWE-agent studies the effect of the agent–computer interface (Yang et al., 2024). Reusable construction is central to LLMs as Tool Makers (Cai et al., 2024), Voyager (Wang et al., 2024a), and ToolMaker (Wölflein et al., 2025). We extend this setting through unnamed artifact discovery and evaluate both scientific execution and subsequent invocation cost.

Scientific agents and capability libraries. ChemCrow organizes chemistry tools (Bran et al., 2024), Coscientist combines tools for chemical research (Boiko et al., 2023), and Biomni supplies a biomedical action space (Huang et al., 2025). The AI Scientist addresses research workflows (Lu et al., 2024), and AI co-scientist emphasizes hypothesis generation (Gottweis et al., 2025). ToolUniverse standardizes scientific invocation and composition (Gao et al., 2025). SciToolAgent-Evo studies ontology-aware capability gaps (Tang et al., 2026); SkillFoundry constructs operational contracts and reusable scientific packages (Shen et al., 2026). Our focus is the measured transition from a task requirement to an installed learned operation and its retained use.

Compilation and executable evaluation. Paper2Agent compiles supplied papers and code into tested MCP servers, evaluates them against repository-access coding agents, and studies validation components (Miao et al., 2026). We use its compiler and Biomni's harness, adding task-driven discovery, artifact locking, and independent execution admission. Contract2Tool studies preconditions and effects (Babu & Iyer, 2026), SkillSmith co-evolves skills and tools (Wei et al., 2026), and Agent Contracts formalizes resource constraints (Ye & Tan, 2026). Executable benchmarks include SWE-bench (Jimenez et al., 2024), ScienceAgentBench (Chen et al., 2024), and DeployBench (Wang et al., 2026). Skill Following distinguishes retrieval from actual use (Cho & Park, 2026). Our protocol connects acquisition history, actual scientific computation, unchanged external scores, and phase costs in one episode record.

3 Method

3.1 Problem formulation

Let $\mathcal{I} = \{c_1, \ldots, c_n\}$ denote the agent's current capability inventory, where each c_i is a previously validated scientific capability. Given a new scientific task x and a resource budget b, the agent decides whether to reuse an existing capability in $\mathcal{I}$, acquire a new capability, or decline the task. When acquisition is required, the goal is to turn an external research artifact into a reusable capability that can be invoked and verified by the agent.

We define a capability c as an executable scientific operation together with the information required to use and validate it:

$$c = (\text{operation}, \text{scope}, \text{I/O schema}, \text{source and weights}, \text{runtime and interface}, \text{execution evidence}).$$

The operation specifies the scientific computation being performed; the scope describes when that computation is applicable; the I/O schema defines how the capability is invoked; the source, weights, and runtime identify the concrete implementation being executed; and the execution evidence records whether the intended scientific computation was actually carried out. Once a newly compiled capability passes admission, it is added to the inventory,

$$\mathcal{I}' = \mathcal{I} \cup \{c\},$$

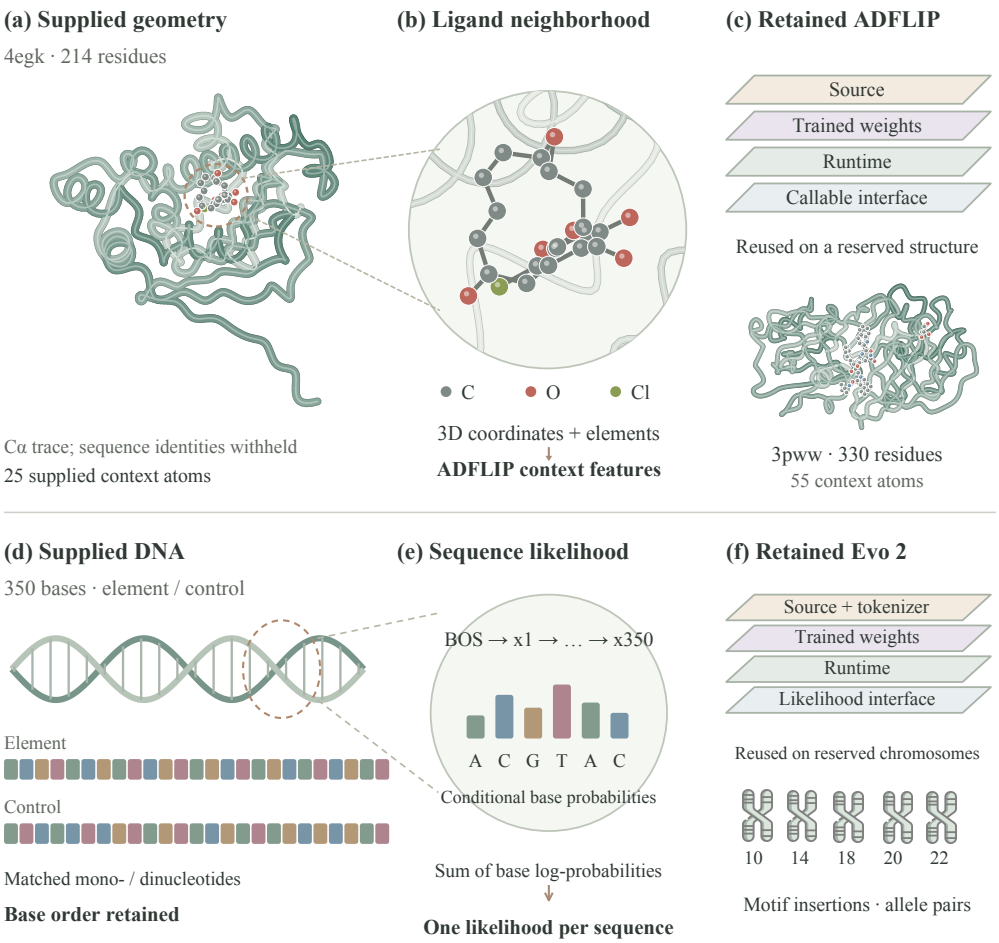

Figure 2: **Preserving the scientific information that motivates acquisition.** Top: the actual 4egk input supplies a 214-residue backbone and 25 context atoms; ADFLIP consumes their coordinates and element identities, and its retained package is invoked on 3pww. Bottom: regulatory elements and dinucleotide-matched controls enter a sequence-likelihood operation with base order intact. The retained Evo 2 operation serves five reserved chromosomes, motif insertions, and allele comparisons. Protein views use supplied coordinates and original PDB connectivity at independent scales; DNA base strings and probability bars illustrate the scoring contract schematically.

so that subsequent tasks can reuse the same validated computation without repeating discovery and compilation.

Figure 2 illustrates why a capability must preserve the task-relevant scientific computation rather than only a callable interface. In protein design, ligand coordinates and element identities must reach the context-conditioned model; in DNA discrimination, base order must be preserved and the capability must expose sequence likelihood rather than a generic representation. These examples motivate retaining the scientific operation together with its inputs, implementation, and execution evidence.

3.2 SciCompiler workflow

Given a scientific task x, the current capability inventory $\mathcal{I}$, and the available resource budget, SciCompiler first determines whether the task can be served by an existing capability, requires acquiring a new capability, or should be declined. Acquisition connects discovery and artifact locking to compilation, package validation, registered execution, and scientific admission. Admitted capabilities enter the retained inventory for subsequent reuse.

Capability decision. SCICOMPILER first evaluates the task against the current inventory and selects one of three actions:

$$d(x, \mathcal{I}) \in \{\texttt{use_existing}, \texttt{acquire}, \texttt{decline}\}.$$

The `use_existing` action invokes a previously validated capability in $\mathcal{I}$. The `acquire` action initiates the construction of a new capability from external research artifacts. The `decline` action terminates the episode when no suitable capability can be identified or constructed within the available constraints.

Discovery and assessment. When `acquire` is selected, SCICOMPILER derives the scientific requirements of the task and generates literature queries to identify candidate research artifacts. For each candidate, the system records the evidence supporting its selection or rejection and verifies the relationship between the scientific publication and the corresponding official repository. Assessment further examines available entry points, execution environments, weights when applicable, license information, and expected outputs. Search may be refined within a bounded budget when the available evidence is insufficient to select an artifact.

Artifact locking and compilation. Once an artifact is selected, SCICOMPILER fixes its repository origin, commit, tracked source, and task-specific contract before compilation. The locked artifact is then passed to Paper2Agent (Miao et al., 2026), whose compiler prepares the execution environment, runs available workflows, extracts candidate tools, generates an MCP server, and evaluates tool coverage. MCP provides named operations with explicit input schemas (MCP Contributors, 2025). Compatibility changes are restricted to generated wrappers, while the locked scientific source and private evaluation data remain outside the repair process.

Scientific admission and retention. Compilation establishes that an operation can be executed through the generated interface, but does not by itself establish that the intended scientific computation was actually performed. Package-level validation first checks the execution environment, generated tools, manifest, passing tests, coverage, and successful MCP initialization and invocation. Diagnostics may trigger compatibility repairs within the declared ceiling, followed by revalidation of the package while its locked scientific source remains unchanged. Biomni then registers the accepted server and invokes the operation on the task. A separate scientific admission stage combines programmatic checks with author-supervised inspection of process and file traces before external scoring. This inspection verifies the source and weights actually read, task-specific settings, the learned computation performed, and the provenance of the submitted output. Admission is evaluated separately from compiler-generated tests and self-reports.

Capabilities that pass scientific admission are retained in the inventory for subsequent Biomni use (Huang et al., 2025). The retained capability preserves the accepted scientific operation together with its source, weights, runtime, callable interface, and execution evidence. Subsequent tasks can therefore invoke the same validated computation without repeating discovery and compilation.

3.3 AN EPISODE PROTOCOL ACROSS CONSTRUCTION AND USE

Before dispatch, the protocol fixes the task inputs and current inventory, reserving later-use inputs in advance. It also binds the source and execution settings to the declared resource budgets. The episode record follows the agent's action and supporting evidence through acquisition and execution to the formal submission. Each attempt retains its terminal status and costs by phase; external scores are added only after all scheduled attempts and independent admission close. Engineering failures leave scores unavailable, while failed requests and repairs remain charged, keeping completion distinct from scientific quality and cost. The resulting record connects what an agent built to what it actually executed and what subsequent tasks can reuse.

4 EXPERIMENTAL DESIGN

We study acquired computation in protein design and genomics, then isolate invocation interfaces and inventory records. Tasks use released data and unchanged external scorers; inputs, controls, budgets, and study units are specified in Appendix A and Table 3.

Scientific tasks. Protein sequence design adapts LigandMPNN small-molecule inputs (Dauparas et al., 2025): an anonymized backbone and nonprotein atomic context define each task. The original `get_seq_rec` measures deposited-sequence recovery over the whole chain and interface. Context-present, context-removed, and exact-repeat conditions share seed 37, one CPU thread, and no GPU. Retention studies use eight further structures at three seeds and twenty original structures at one seed.

Three adapted DART-Eval tasks (Patel et al., 2024) use unlabelled 350-base DNA sequences with opaque identifiers. Regulatory discrimination contains 128 element/dinucleotide-shuffled-control pairs on each of six test chromosomes: chromosome 5 is initial; 10, 14, 18, 20, and 22 are reserved. The same retained likelihood operations then serve 512 motif/shuffled-motif pairs and 512 caQTL/control variants. Original scorers produce regulatory and motif accuracy and variant AUROC/AP (Appendix H).

Acquisition against coding. Both arms receive the same tasks and inventory. They have equal permission to search and install artifacts and operate under the same GLM-5.3 request/time ceilings. Protein initial pairs allow 640 requests/9,900 seconds; DNA allows 720/10,800 and one A40 per arm. Reserved DNA batches allow 30/1,800; motif and variant probes allow 30/3,600. Coding may retain its own artifact and helper. External protein comparisons add native Biomni and supplied-repository Paper2Agent; a separate paired compiler study tests task scoping (Appendix C).

Controlled interventions. The interface studies invoke the same learned function and weights in a fixed runtime, with identical inputs and settings. They vary direct Python versus registered MCP together with the corresponding prompt/tool context. Twenty protein pairs use alternating order and a 20-request/900-second ceiling; six DNA batch pairs use 15/600 and one scientific call per attempt. The inventory studies first vary whether a capability is present, then whether its record includes validated execution facts. A further study separates those facts into two constituent text blocks. Each condition receives three fresh `deepseek-v4-flash` decisions; the respective studies' 270, 180, and 240 observations measure recommendations.

Analysis. Every scheduled attempt remains in its denominator; costs include failed requests and repairs. Paired intervals resample the unit defined by each study, keeping observations from the same structure or decision task together. The DNA interface study uses batches as its resampling unit. For regulatory accuracy, paired observations are resampled within each fixed chromosome. Acquisition-plus-use prefixes include the initial investment; setup overhead and missing usage are reported separately. All adaptations and interventions are bound to their owning protocols.

5 Results

5.1 Acquisition constructs reusable scientific operations

The two complete protein development pairs acquire ADFLIP for 4egk (Yi et al., 2025) and LASErMPNN, the sequence-design component of NISE, for 5qqp (Fry et al., 2026). Coding selects LigandMPNN in both. Independent admission verifies the source, weights, context pathway, and submitted outputs before original scoring.

Table 1: **First-use acquisition versus coding with matched access.** Recovery is percent; total arm minutes include failures and repairs. Each case contains one structure and one sample per condition.

Case	Arm	Artifact	Whole (%)	Interface (%)	Requests	Time (min)
ADFLIP	Coding	LigandMPNN	57.01	73.33	40	11.40
ADFLIP	Open-world	ADFLIP	52.80	46.67	225	111.04
LASErMPNN	Coding	LigandMPNN	58.82	54.35	39	7.85
LASErMPNN	Open-world	LASErMPNN	57.98	45.65	306	94.78

Compilation is the principal first-use investment. It consumes 210/225 ADFLIP and 287/306 LASErMPNN requests, accounting for 90.1% and 87.3% of arm time. First-use acquisition incurs

a substantial compilation cost, while leaving an independently admitted operation available for subsequent reuse(Table 1).

Task scoping connects compilation to execution. In a fresh supplied-repository comparison, task-scoped compilation produces an admitted LASErMPNN operation at 232 requests, with 69.09% whole-chain and 74.14% interface recovery. Upstream Paper2Agent uses 368 requests and reaches its compiler deadline before final acceptance or task inference. With the repository and Biomni consumer fixed, this case demonstrates a task-scoped route from compilation to verified scientific execution. The task contract, tutorial scope, and generated interface form the intervention; the compiler comparison and a separate external-system cohort are detailed in Appendix C.

5.2 Retained computation remains usable across inputs

All 72 assigned invocations on eight further protein structures score, and every exact repeat matches. This includes both acquired operations and retained LigandMPNN at three fixed seeds. Whole-chain differences from coding are $+1.17$ points ($[-0.63, 3.45]$) for ADFLIP and $+0.87$ ($[-1.11, 3.39]$) for LASErMPNN. On twenty original inputs, all forty new LASErMPNN/coding invocations also score; the primary whole-chain difference is -0.18 points ($[-2.00, 1.60]$).

Table 2: **Retained configurations on two fixed input groups.** Recovery is percent; Δ is present minus absent, in percentage points. The twenty-input follow-up newly executes LASErMPNN/LigandMPNN and references closed ADFLIP scores. The eight-input study averages three seeds within each structure. Groups remain separate. Scored denominators count structures.

Configuration	Scored	Rec. (%)	Int. (%)	ΔRec.	ΔInt.
Original 20 inputs, seed 37; ADFLIP is historical					
LigandMPNN	20/20	52.43	60.53	+2.01	+12.25
ADFLIP (historical)	20/20	51.04	54.93	+1.79	+11.72
LASErMPNN	20/20	52.26	57.16	+1.83	+14.64
Further 8 inputs, three seeds per configuration					
LigandMPNN	8/8	51.39	65.71	+1.27	+17.79
ADFLIP	8/8	52.56	60.94	+2.76	+19.48
LASErMPNN	8/8	52.26	57.46	+2.85	+20.88

Atomic context contributes consistently on the further inputs. Mean interface recovery gains from retaining context are $+19.48$, $+20.88$, and $+17.79$ points for ADFLIP, LASErMPNN, and LigandMPNN, respectively. Whole-chain recovery is similar across configurations; interface means favor coding. Seed variation and all contrasts appear in Appendix D; input relationships appear in Appendix E.

5.3 One acquired DNA operation serves three scientific tasks

Acquisition selects Evo 2 7B (Brixi et al., 2026) and retains its sequence-likelihood operation for regulatory discrimination, motif footprinting, and allele comparison; coding selects Nucleotide Transformer 500M (Dalla-Torre et al., 2025). The reported Evo 2 configuration includes the compatibility-only continuation and submission-envelope intervention specified in Appendix H.2; the locked scientific source, checkpoint, prediction bytes, and external scorer remain unchanged. Figure 3(a,b) identifies this configuration as assisted.

Regulatory discrimination improves across every reserved chromosome. The acquired configuration reaches 96.25% versus coding's 78.59% on 640 reserved pairs: $+17.66$ points ($[14.38, 21.09]$). Across all 768 pairs, accuracy is 96.09% versus 79.95%. Inputs preserve exact mono- and dinucleotide matching, and no positive-sequence pair reaches the declared 80% edit-similarity threshold (Appendix H). A post-hoc error breakdown finds 126 reserved pairs correct only for the acquired configuration, thirteen only for coding, and eleven missed by both, showing where aggregate gains arise.

Reuse connects the acquired operation to subsequent tasks. All seven subsequent full-route decisions select and invoke the existing capability. The five regulatory reserves use 45 versus 60 requests;

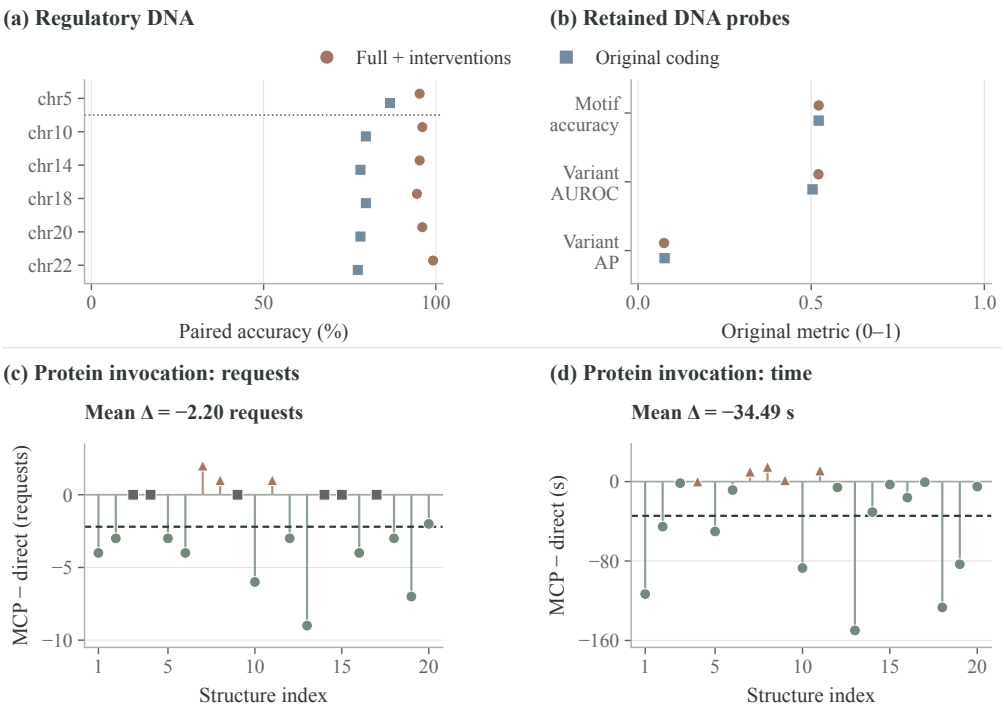

Figure 3: **Scientific utility and invocation cost of retained operations.** (a,b) DNA outcomes compare assisted Evo 2 with original Nucleotide Transformer configurations, including their different likelihood definitions. The dotted boundary separates the initial chromosome from five reserves. Motif accuracy, variant AUROC, and average precision retain their original scales. (c,d) A separate twenty-structure protein study shows MCP-minus-direct generation requests and native worker seconds. All pairs produce identical sequences in context-present, context-removed, and repeat conditions. Dashed lines mark paired means; negative values favor MCP. Interface and prompt/tool context vary together; invocation costs exclude acquisition and external integrity checks.

total acquisition-plus-use requests remain 346 versus 89. The same likelihood operations complete motif and variant probes (Figure 3(b)): motif accuracy ties at 52.15%, and variant AUROC is 0.521 versus 0.504 (AP 0.0751 versus 0.0764). A reusable likelihood interface therefore supports multiple workflows while scientific utility depends on the task. Complete results and original/intervention accounting appear in Appendix H.5.

5.4 Invocation interfaces preserve computation and shape agent work

All twenty protein pairs produce identical sequences through direct Python and MCP, including context-present, context-removed, and repeat conditions. Figure 3(c,d) shows the per-structure invocation costs; Appendix Figure 7 details the intervention. MCP reduces mean requests from 9.4 to 7.2 (23.4%) and native time from 154.52 to 120.03 seconds (22.3%). Paired differences are -2.20 requests ($[-3.50, -1.00]$) and -34.49 seconds ($[-57.49, -14.38]$). Savings persist in both execution-order groups and every leave-one-structure-out mean; prompt tokens fall 70.5%. The median time difference is -7.20 seconds, indicating larger savings on harder direct invocations (Appendix G).

The DNA replication preserves outputs with a different cost profile. Five admitted pairs have exactly equal likelihoods across 1,280 sequences. All six scheduled pairs contribute to the cost analysis, with MCP-minus-direct differences of -0.83 requests ($[-2.83, 0.83]$) and $+35.71$ seconds ($[-19.93, 96.65]$). The DNA study thus reproduces exact output agreement, while both cost intervals span zero. Appendix H.6 reports the complete submissions, admission outcomes, and batch costs.

5.5 Execution evidence makes retained capabilities easier to select

Capability presence changes recommended actions. With ADFLIP available, sixty inverse-folding decisions yield 48 use recommendations; removing it yields one. All ten acquire responses in the available condition cite a backbone-only caveat inherited from an earlier ProteinMPNN configuration. The accepted ADFLIP operation consumes context, motivating a capability-record update with its validated execution facts.

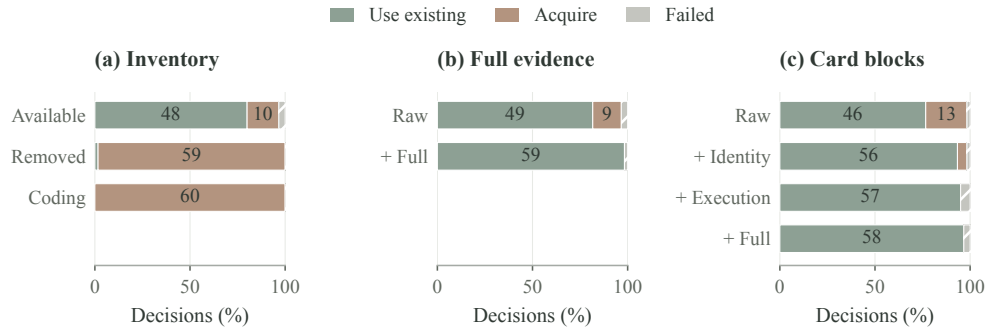

Figure 4: **Capability presence and execution records affect recommended actions.** (a) Original inventory intervention; (b) fresh evidence-card comparison; (c) fresh four-condition component follow-up. Each bar contains sixty decisions on the same twenty inverse-folding instances, with three observations each. Numbers are counts; hatching marks failures. Studies have separate observations and execute no chosen actions.

Execution records increase use recommendations in two fresh studies. Adding an evidence card raises use from 49/60 to 59/60, a task-level increase of 16.7 points ($[8.3, 26.7]$). A separate four-condition study on the same twenty tasks yields 46/56/57/58 use recommendations for raw, identity/history, execution-record, and full cards. Full-minus-raw is $+20.0$ points ($[8.3, 31.7]$). The prespecified identity/history and execution-record block effects are $+9.17$ ($[3.33, 15.83]$) and $+10.83$ ($[2.50, 19.17]$) points (Appendix F.1). Across two fresh studies, augmenting capability records with validated artifact and execution facts increases reuse recommendations, with positive effects from both constituent blocks in the component study.

6 Discussion

Scientific capabilities combine computation, interfaces, and evidence. The acquired package preserves the inputs and learned computation needed by subsequent tasks. Its interface determines the work required to invoke it, and its execution record helps the agent recognize when it is usable. The protein interface result and repeated inventory interventions support evaluating these three components together. The DNA tasks further show why execution receipts and task utility belong in the same capability record: one callable likelihood operation has strong regulatory discrimination but limited motif and variant utility.

Implications and scope. Acquisition policies should balance construction cost against expected future use. Broader artifact coverage, automatic reconciliation of capability records, and longer-term evaluation of cost-aware routing are natural next steps.

7 Conclusion

SciCompiler connects open research artifacts to executable scientific capabilities that an agent can retain and use. Across protein design and genomics, the evaluation follows this transition through discovery, compilation, scientific execution, and subsequent invocation. Controlled interface and inventory studies show that packaging and validated execution evidence are consequential parts of the retained capability. This provides a concrete basis for building and evaluating scientific agents whose repertoires grow through use.

AI USE STATEMENT

Generative AI assistants supported research planning, literature synthesis, implementation, debugging, execution supervision, evidence analysis, manuscript revision, and figure/LaTeX preparation. The evaluated systems use language models as specified in Section 4. Scientific outcomes come from unchanged external scorers after study closure. Numerical plots read closed results; mechanism diagrams use programmatic vector graphics. Authors are responsible for the content and AI-assisted artifacts.

ETHICS STATEMENT

The studies use released software and public structural and genomic data, with no new human-subject, animal, or wet-laboratory experiments. Outputs are computational predictions. Third-party software remains subject to its licenses and execution risks; provenance and integrity checks are not a security sandbox.

REPRODUCIBILITY STATEMENT

Section 3.3 defines the episode protocol; Appendices A–H provide task adaptations, settings, controls, costs, and interventions. Frozen declarations bind inputs, source, settings, and budgets before dispatch. Anonymous numerical, typesetting, and current-code bundles provide exported values and verifiers, manuscript sources and figures, and harness/regression code with upstream licenses. Historical raw traces and generated environments remain in the local archive. Numerical and implementation checks are distinguished from scientific reruns.

REFERENCES

Rahul Suresh Babu and Laxmipriya Ganesh Iyer. Contract2Tool: Learning Preconditions and Effects for Reliable Tool-Augmented LLM Agents. arXiv:2606.07904v1, 2026. Version 1, 2026-06-05.

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.

Garyk Brixi, Matthew G. Durrant, Jerome Ku, et al. Genome modelling and design across all domains of life with Evo 2. *Nature*, 652:1349–1361, 2026. doi: 10.1038/s41586-026-10176-5.

Tianle Cai, Xuezhi Wang, Tengyu Ma, Xinyun Chen, and Denny Zhou. Large Language Models as Tool Makers. In *International Conference on Learning Representations*, 2024.

Ziru Chen, Shijie Chen, Yuting Ning, Qianheng Zhang, Boshi Wang, Botao Yu, Yifei Li, Zeyi Liao, Chen Wei, Zitong Lu, Vishal Dey, Mingyi Xue, Frazier N. Baker, Benjamin Burns, Daniel Adu-Ampratwum, Xuhui Huang, Xia Ning, Song Gao, Yu Su, and Huan Sun. ScienceAgentBench: Toward Rigorous Assessment of Language Agents for Data-Driven Scientific Discovery. arXiv:2410.05080v3, 2024. Version 3, 2025-03-31.

Seonghyeon Cho and Chanjun Park. Skill Following: Evaluating Actual Skill Use in Retrieval-Enabled LLM Agents. arXiv:2609.00549v1, 2026. Version 1, 2026-09-01.

Hugo Dalla-Torre, Liam Gonzalez, Javier Mendoza-Revilla, Nicolas Lopez Carranza, Adam Henryk Grzywaczewski, Francesco Oteri, Christian Dallago, Evan Trop, Bernardo P. de Almeida, Hassan Sirelkhatim, Guillaume Richard, Marcin Skwark, Karim Beguir, Marie Lopez, and Thomas Pierrot. Nucleotide transformer: building and evaluating robust foundation models for human genomics. *Nature Methods*, 22:287–297, 2025. doi: 10.1038/s41592-024-02523-z.

Justas Dauparas, Gyu Rie Lee, Robert Pecoraro, Linna An, Ivan Anishchenko, Cameron Glasscock, and David Baker. Atomic context-conditioned protein sequence design using LigandMPNN. *Nature Methods*, 22(4):717–723, 2025. doi: 10.1038/s41592-025-02626-1.

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.

Benjamin Fry, Kaia Slaw, and Nicholas F. Polizzi. Zero-shot design of drug-binding proteins via neural iterative selection-expansion. *Nature*, 656(8126):237–249, 2026. doi: 10.1038/s41586-026-10670-w.

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. arXiv:2509.23426v3, 2025. Version 3, 2026-08-07.

Juraj Gottweis, Wei-Hung Weng, Alexander Daryin, Tao Tu, Anil Palepu, Petar Sirkovic, Artiom Myaskovsky, Felix Weissenberger, Keran Rong, Ryutaro Tanno, Khaled Saab, Dan Popovici, Jacob Blum, Fan Zhang, Katherine Chou, Avinatan Hassidim, Burak Gokturk, Amin Vahdat, Pushmeet Kohli, Yossi Matias, Andrew Carroll, Kavita Kulkarni, Nenad Tomasev, Yuan Guan, Vikram Dhillon, Eeshit Dhaval Vaishnav, Byron Lee, Tiago R D Costa, José R Penadés, Gary Peltz, Yunhan Xu, Annalisa Pawlosky, Alan Karthikesalingam, and Vivek Natarajan. Towards an AI co-scientist. arXiv:2502.18864v1, 2025.

Michael R. Gryk and Bertram Ludäscher. Workflows and provenance: Toward information science solutions for the natural sciences. *Library Trends*, 65(4):555–562, 2017. doi: 10.1353/lib.2017.0018.

Kexin Huang, Serena Zhang, Hanchen Wang, Yuanhao Qu, Yingzhou Lu, Yusuf Roohani, Ryan Li, Lin Qiu, Gavin Li, Junze Zhang, Di Yin, Shruti Marwaha, Jennefer N. Carter, Xin Zhou, Matthew Wheeler, Jonathan A. Bernstein, Mengdi Wang, Peng He, Jingtian Zhou, Michael Snyder, Le Cong, Aviv Regev, and Jure Leskovec. Biomni: A General-Purpose Biomedical AI Agent. *bioRxiv*, 2025. doi: 10.1101/2025.05.30.656746. Version 1, posted 2025-06-02.

Carlos E. Jimenez, John Yang, Alexander Wettig, Shunyu Yao, Kexin Pei, Ofir Press, and Karthik Narasimhan. SWE-bench: Can Language Models Resolve Real-World GitHub Issues? In *International Conference on Learning Representations*, 2024.

Jeremy Li, Suyash Shringarpure, Edmund Wong, and Andrew Ho. GeneBench-Pro: Evaluating Multistage Statistical Reasoning in Genomics, Quantitative Biology, and Translational Biomedicine. *bioRxiv*, 2026. doi: 10.64898/2026.06.29.735386.

Chris Lu, Cong Lu, Robert Tjarko Lange, Jakob Foerster, Jeff Clune, and David Ha. The AI Scientist: Towards Fully Automated Open-Ended Scientific Discovery. arXiv:2408.06292, 2024.

MCP Contributors. Model Context Protocol: Tools Specification, 2025. Online specification, revision 2025-06-18.

Jiacheng Miao, Joe R. Davis, Yaohui Zhang, Jonathan K. Pritchard, and James Zou. Reimagining research papers as interactive and reliable AI agents. *Nature*, 2026. doi: 10.1038/s41586-026-11044-y. Published online 16 September 2026.

Aman Patel, Arpita Singhal, Austin Wang, Anusri Pampari, Maya Kasowski, and Anshul Kundaje. DART-Eval: A comprehensive DNA language model evaluation benchmark on regulatory DNA. In *Advances in Neural Information Processing Systems*, volume 37, 2024. doi: 10.52202/079017-1981.

Shuo Ren, Can Xie, Pu Jian, Zhenjiang Ren, Chunlin Leng, and Jiajun Zhang. Towards scientific intelligence: A survey of llm-based scientific agents, 2025.

Timo Schick, Jane Dwivedi-Yu, Roberto Dessi, Roberta Raileanu, Maria Lomeli, Eric Hambro, Luke Zettlemoyer, Nicola Cancedda, and Thomas Scialom. Toolformer: Language Models Can Teach Themselves to Use Tools. In *Advances in Neural Information Processing Systems*, volume 36, 2023.

Shuaike Shen, Wenduo Cheng, Mingqian Ma, Alistair Turcan, Martin Jinye Zhang, and Jian Ma. SKILLFOUNDRY: Building Self-Evolving Agent Skill Libraries from Heterogeneous Scientific Resources. arXiv:2604.03964v1, 2026. Version 1, 2026-04-05.

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. arXiv:2607.28692v1, 2026. Version 1, 2026-07-30.

Guanzhi Wang, Yuqi Xie, Yunfan Jiang, Ajay Mandlekar, Chaowei Xiao, Yuke Zhu, Linxi Fan, and Anima Anandkumar. Voyager: An Open-Ended Embodied Agent with Large Language Models. *Transactions on Machine Learning Research*, 2024a.

Xingyao Wang, Yangyi Chen, Lifan Yuan, Yizhe Zhang, Yunzhu Li, Hao Peng, and Heng Ji. Executable Code Actions Elicit Better LLM Agents. In *Proceedings of the 41st International Conference on Machine Learning*, volume 235 of *Proceedings of Machine Learning Research*, pp. 50208–50232, 2024b.

Yuanli Wang, Yaoyao Qian, Yue Zhang, Hanhan Zhou, Jindan Huang, Tianfu Fu, Qiuyang Mang, Huanzhi Mao, Wenhao Chai, Wendong Fan, and Liqiang Jing. DeployBench: Benchmarking LLM Agents for Research Artifact Deployment. arXiv:2606.05238v1, 2026. Version 1, 2026-06-03.

Yangbo Wei, Zhen Huang, Shaoqiang Lu, Junhong Qian, Qifan Wang, Chen Wu, and Lei He. SkillSmith: Co-Evolving Skills and Tools for Self-Improving Agent Systems. arXiv:2606.01314v1, 2026. Version 1, 2026-05-31.

Georg Wölflein, Dyke Ferber, Daniel Truhn, Ognjen Arandjelović, and Jakob Nikolas Kather. LLM Agents Making Agent Tools. In *Proceedings of the 63rd Annual Meeting of the Association for Computational Linguistics (Volume 1: Long Papers)*, pp. 26092–26130, 2025. doi: 10.18653/v1/2025.acl-long.1266.

John Yang, Carlos E. Jimenez, Alexander Wettig, Kilian Lieret, Shunyu Yao, Karthik Narasimhan, and Ofir Press. SWE-agent: Agent-Computer Interfaces Enable Automated Software Engineering. In *Advances in Neural Information Processing Systems*, volume 37, 2024.

Shunyu Yao, Jeffrey Zhao, Dian Yu, Nan Du, Izhak Shafran, Karthik Narasimhan, and Yuan Cao. ReAct: Synergizing Reasoning and Acting in Language Models. In *International Conference on Learning Representations*, 2023.

Qing Ye and Jing Tan. Agent Contracts: A Formal Framework for Resource-Bounded Autonomous AI Systems. arXiv:2601.08815v3, 2026. Version 3, 2026-03-25.

Kai Yi, Kiarash Jamali, and Sjors H. W. Scheres. All-atom inverse protein folding through discrete flow matching. In *Proceedings of the 42nd International Conference on Machine Learning*, volume 267 of *Proceedings of Machine Learning Research*, pp. 72244–72257. PMLR, 2025.

Zijian Zhang, Aiwei Yin, Amaan Baweja, Jiaru Bai, Ignacio Gustin, Varinia Bernales, and Alán Aspuru-Guzik. El agente forjador: Task-driven agent generation for quantum simulation, 2026.

A SCIENTIFIC TASKS AND REPRODUCIBLE SETTINGS

Table 3: **Study units and interventions.** Multiple sequences, controls, seeds, or repeated decisions do not constitute additional acquisitions. Protein development history spans system revisions. DNA acquisition and transfer cohorts are declared before outcome scoring; the later interface study reuses scored inputs to test invocation equivalence and cost.

Study	Scientific inputs	Agent-level intervention
Protein acquisition	13 development attempts; external 3-input cohort	Matched-access coding; external systems and supplied-repository compiler
Protein retention	8 further structures × 3 seeds; 20 original structures	Assigned execution of three retained scientific configurations
DNA acquisition/use	768 regulatory element/control pairs, six chromosomes	One initial pair; five conditional retained-use pairs
DNA motif probe	16 motifs × 16 backgrounds × 2 orientations	One conditional pair using the same retained operations
DNA variant probe	512 caQTL/control variants; 1,024 alleles	One conditional pair using unchanged likelihood operations
Quantitative accessibility	144 K562 regions; two further batches of 36	Fresh sequence-to-assay acquisition/coding pair; conditional retention
Invocation interface	20 protein structures; six existing DNA batches	Direct Python versus registered MCP; separate 20-pair and six-pair studies
Inventory records	20 inverse-folding tasks; 10 GeneBench cases	Inventory presence, execution evidence, and literal card blocks; recommendations only

A.1 INPUTS, CONTROLS, AND OUTCOME METRIC

Execution tasks use the released LigandMPNN small-molecule evaluation list and interface information (Dauparas et al., 2025). Preparation extracts the declared chain's backbone, retains selected nonprotein heavy atoms, anonymizes identifiers, and replaces residue identities with glycine. The reference is the deposited observed sequence. Removing context leaves the backbone unchanged; the repeat duplicates the present input. Missing assembly partners are not reconstructed. Extraction and interface mapping are checked against the deposited coordinates before dispatch.

The unchanged `get_seq_rec` in LigandMPNN's `data_utils.py` scores canonical sequences using the frozen all-position or interface mask. Whole-chain context-present recovery is primary; interface recovery and context effects are descriptive. This measures deposited-sequence agreement. Models are not retrained, and local exposure exclusions do not rule out training overlap.

The ADFLIP acquisition uses 4egk (214 residues, 25 context atoms, 15 interface positions); its reserved task is 3pww (330 residues, 55 context atoms, 31 interface positions). Both are fixed before first prediction, with earlier preparation history disclosed. LASErMPNN acquisition uses 5qqp (238 residues, 76 context atoms, 46 interface positions), selected at source rank 185 after structural exclusions at ranks 183–184. It has no reserved reuse task.

A.2 LEARNED ARTIFACTS AND INFERENCE CONFIGURATION

ADFLIP uses adaptive sampling with eight steps, threshold 0.9, final argmax, and released exponential-moving-average weights (Yi et al., 2025). Its accepted function reads nonprotein coordinates and element identities through the original context pathway. All interface studies retain this function, source, checkpoint, runtime, and configuration. Except for the declared three-seed transfer study, each diagnostic row independently resets seed 37. Every row generates one sample on one CPU thread and uses no GPU.

The original 5qqp LASErMPNN acquisition uses one design per input and batch, one input at a time, upstream temperature defaults, and default disabled residues `X,C`. Ligand context is enabled; water and noncanonical-amino-acid ligand options are off. The source recommends protonated

ligands, whereas these frozen inputs contain heavy atoms. This executes the sequence-design component described in NISE (Fry et al., 2026), not its iterative joint design procedure. Coding selects LigandMPNN in both development-history pairs. Source and checkpoint reads, context counts, sampling, and formal outputs are checked from process and file evidence. The later external cohort has separately acquired settings (Appendix C).

A.3 Agent configuration and study schedules

Both complete acquisition comparisons request GLM-5.3, with 640 generation requests, 9,900 seconds, and a 16,384-output-token limit per request. Acquisition reserves up to 30 requests/1,500 seconds for discovery and 20 requests/900 seconds for inference; compilation receives the remaining allocation. Arms run in parallel under a four-hour experiment ceiling. Native compiler, coding, and discovery transports differ; requested aliases are recorded separately from returned provider identifiers. The LASErMPNN case permits generic invocation repair only after demonstrably pre-sampling failure, within the same three-repair ceiling. No extra formal scientific sample is taken.

Scientific workers receive declared inputs and source/runtime assets, with private references and earlier outputs outside their mounts. Package acceptance checks tests, coverage, and MCP calls; scientific admission checks provenance, learned execution, sampling, and formal output. Timing uses a shared host.

The interface pilot uses twenty retained-use structures. The follow-up selects source ranks 190–224; the final study uses twenty eligible inputs at ranks 227–283 under the frozen source-order rule. Arms alternate order with twenty requests/900 seconds per attempt and 800 requests/twelve hours per cohort. All attempts terminate before scores open. Exact input membership is retained in the released per-instance records.

The inventory intervention uses the twenty retained-use task descriptions and all ten GeneBench-Pro public cases in release order (package version 2026-06-26). These cases span statistical genetics, genomics, and translational reasoning (Li et al., 2026); they contribute only decision prompts, with no reference answers or grader loaded. They had prior development exposure. The capability-available condition displays the complete original ADFLIP manifest, including task-specific prose and inherited caveats. Other shared fields stay fixed.

The decision model is requested as `deepseek-v4-flash`; the frozen declaration records the provider's V4.1-Flash alias. Temperature is 0.1, output ceiling 16,384 tokens, and timeout 300 seconds; no seed is sent. Each of 270 slots makes one request with no retry or replacement. Inventory order rotates within each task and observation. Observation indices are bookkeeping pairs rather than shared random draws. Three decisions fail; none of the chosen actions is executed.

B First-use costs and scientific controls

First-use acquisition is dominated by compilation. ADFLIP acquisition used 225 generation requests—6 for discovery, 210 for compilation, and 9 for inference—and 111.04 minutes in total. LASErMPNN used 306 requests—5, 287, and 14, respectively—and 94.78 minutes. The corresponding matched-access coding runs used 40 and 39 requests and 11.40 and 7.85 minutes. Failed requests and permitted repairs remain charged. Common provisioning, input preparation, external scoring, and human effort are excluded from these arm times. Request-level failures, incomplete token-usage records, and full phase traces are provided in the released numerical records.

Table 4: **Context-present minus context-removed recovery.** Values are percentage points. Both arms' exact-repeat outputs match the corresponding present output.

Case	Arm	Whole-chain difference	Interface difference
ADFLIP	Coding (LigandMPNN)	$+4.21$	$+33.33$
ADFLIP	Open-world (ADFLIP)	-0.93	-6.67
LASErMPNN	Coding (LigandMPNN)	$+2.10$	$+6.52$
LASErMPNN	Open-world (LASErMPNN)	$+2.52$	-2.17

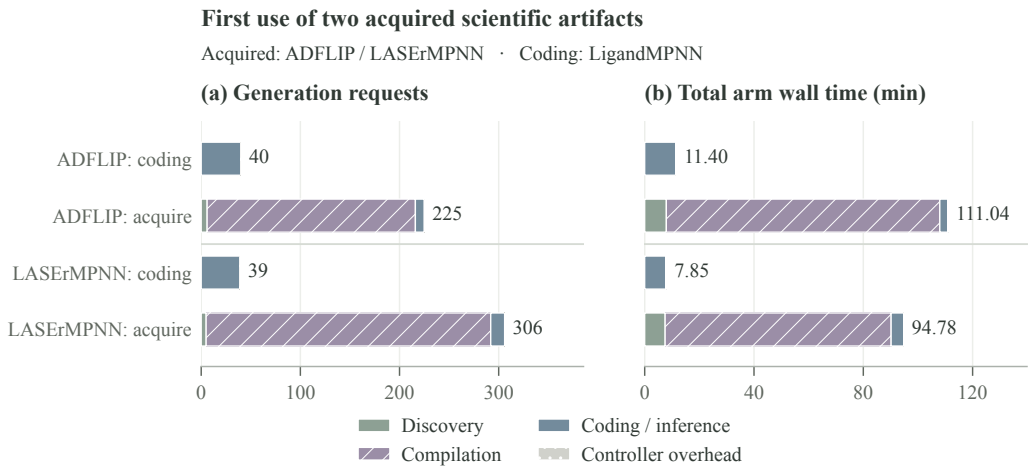

Figure 5: **Most first-use acquisition work constructs the retained operation.** Compilation includes environment work, execution, packaging, tests, failed phases, and repairs. Coding has the same acquisition permissions. The two cases use different structures and artifacts.

C PROSPECTIVE EXTERNAL-SYSTEM COMPARISON

The external-system cohort fixes 5lny, 3b5r, and 1gx8 at source ranks 285, 288, and 292. Four methods share initial inventory, acquisition permissions, GLM-5.3, and 640 requests/9,900 seconds: coding, full acquisition, native Biomni, and upstream Paper2Agent followed by Biomni. Biomni retains native retrieval and code execution, with transport and isolated-data adaptations. Paper2Agent receives the full arm's verified repository and 620 requests/9,000 seconds for compilation plus 20/900 for use. Its native interface has no task-specific overlay.

All twelve slots terminate before independent admission and original scoring. Seven methods submit and four pass scientific admission. Coding on 5lny and Biomni on 3b5r/1gx8 take four, four, and seven samples, exceeding the three-row contract; their scores remain unavailable. Scored Biomni uses LigandMPNN checkpoint 020, while admitted coding uses 010.

All four admitted methods match exact repeats. Present-minus-absent whole/interface differences are $-1.45/-11.76$ points for Biomni on 5lny, $+2.43/+9.09$ for coding on 3b5r, $+1.24/+13.33$ for coding on 1gx8, and $+6.21/+33.33$ for full on 1gx8. Total cost is 1,139 generation requests and fourteen token-count calls; four requests lack usage. Paper2Agent has two dispatched supplied-repository attempts, with no scored output.

C.1 SUPPLIED-REPOSITORY COMPILER COMPARISON

A separate declaration, fixed before the preceding cohort's private scores are opened, assigns fresh rank 294 / 3wz6 (330 residues, 103 context atoms, 58 interface positions) to two new compilers. Both receive the same locked LASErMPNN repository and native Biomni consumer, with common prospective pip-bootstrap and unified-entry-point fixes, zero external repairs, and unchanged upstream internal iterations. Each has 620 requests/9,000 seconds for compilation plus 20/900 for use. Task scoping changes the task contract, tutorial scope and prescribed interface together; this is a component case, not a discovery experiment or an isolation of those mechanisms. Earlier failed attempts remain closed.

Task-scoped compilation passes six independent package tests with no external repair. Its LASErMPNN configuration uses temperature 0.1, disabled residue X, the released nothing-held-out checkpoint, and upstream beta-fixing/repacking options. Three CPU processes each reset Python, NumPy, and Torch seed 37. Only tutorial B-factor markings change; coordinates and elements are preserved. Original parsers verify 103/0/103 context atoms and 101/0/101 first-shell atoms. Without

Table 5: **Fresh compilers given the same repository.** Recovery is percent; both scheduled methods remain in the denominator. Times include failed phases and exclude outer supervision. Upstream Paper2Agent reaches its compiler ceiling without final independent acceptance or task inference; NA is unavailable recovery. There is no complete scientific quality pair.

Structure	Compiler	Whole (%)	Interface (%)	Requests	Min.
3wz6	Task-scoped compiler	69.09	74.14	232	83.23
3wz6	Upstream Paper2Agent	NA	NA	368	150.00

context, whole/interface recovery is 66.36%/55.17%; exact repeats match. A failed dictionary lookup precedes the sole registered invocation.

Upstream Paper2Agent completes five stage invocations and reports 26 passing tests, but reaches its deadline before final independent acceptance or inference. The remaining validation activity is unknown. Both slots total 600 generation requests and five token-count calls, with complete usage. A supervision failure before any request and subsequent failed phases remain recorded.

D TRANSFER OF RETAINED SCIENTIFIC OPERATIONS

The retained-operation study invokes the unchanged accepted ADFLIP and LASErMPNN functions on further structures. Source, checkpoints, runtime, and inference defaults remain fixed; seeds 37, 73, and 101 apply separately to context-present, context-removed, and repeat rows. Execution is assigned by the controller.

Preparation starts after rank 294. The declared target of twenty exhausts the released source-order suffix, yielding eight eligible structures at ranks 299, 300, 301, 302, 303, 306, 307, and 315. All eight are fixed before prediction. Independent reconstruction verifies backbone, context, reference, and interface mask.

Each invocation uses a fresh isolated workspace, one CPU thread, no GPU or network, a 180-second limit, and no retry. Artifact order alternates within structure and seeds follow their fixed order. Forty-eight acquired-operation invocations schedule 144 rows. Function and process traces verify sampling, seed resets, parser counts, source, and checkpoint access. Only explicitly returned files enter scoring after all attempts and integrity checks close.

Primary summaries average the three seeds within each complete structure and then weight structures equally. Pairwise contrasts require complete seeds for both artifacts. Partial outcomes retain their denominators. Context effects, repeat agreement, seed variation, and paired differences use the same structure-level bootstrap as Appendix G. Runtime is local invocation time.

Retained coding comparator. The retained coding comparator freezes the original accepted LigandMPNN source, checkpoint, runtime, and settings before transfer scores open. New generic arguments and serialization wrap its unchanged official `run.py`, since the original coding shell hard-codes its first task. It uses temperature 0.1, one sample, atom context enabled, side-chain context and packing disabled, zero-occupancy parsing disabled, one CPU thread, and no GPU. Twenty-four invocations cover the same eight structures, three seeds, and controls.

Admission checks original CLI entry, featurization, sampling, seed reset, parser counts, and checkpoint reads. Each FASTA contains exactly the native/input record and one canonical design of the declared length. Scoring uses private deposited sequences after closure; CLI diagnostics against glycine placeholders are excluded. Each artifact keeps its accepted decoding settings.

All 48 acquired and 24 retained-coding invocations pass admission and scoring: 216 rows, 432 original metric calls, and 72 matching repeats. Native process totals are 1,976.09 seconds for ADFLIP, 963.55 for LASErMPNN, and 267.82 for LigandMPNN. The preparation shortfall and artificial tracepoint correction remain separate records.

ADFLIP has higher/lower whole-chain structure means than LigandMPNN on 6/2 inputs; LASErMPNN has higher/equal/lower means on 3/2/3. Interface comparisons are 3/0/5 and 0/1/7, respectively. The ADFLIP interface difference is -4.78 points ($[-12.13, 2.01]$); LASErMPNN's is

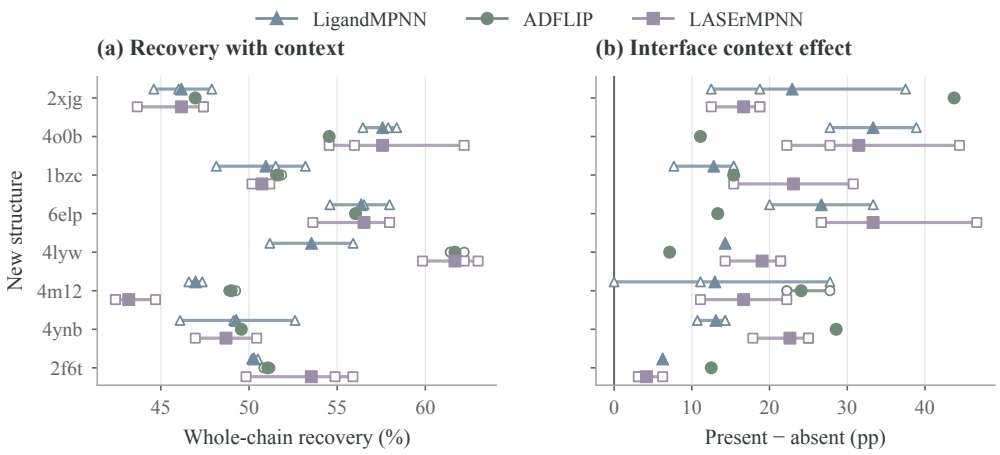

Figure 6: **Retained configurations on eight further structures.** Open markers show three prespecified seeds, filled markers their mean, and lines their range. (a) Whole-chain recovery with context; (b) interface recovery with minus without context. Seeds and controls share the same eight structure units. This cohort is designed to evaluate retained-operation reuse across additional structures rather than family-level distribution shift.

-8.25 ($[-11.29, -5.01]$). These are fixed-input contrasts between retained settings, including each operation's original decoding defaults.

Mean within-structure seed standard deviations for whole-chain recovery are 0.13, 2.15, and 1.64 points for ADFLIP, LASErMPNN, and LigandMPNN. ADFLIP uses final argmax and has identical interface recovery across these seeds; low variability is specific to that retained configuration. Mean interface context effects are $+19.48$ ($[12.48, 27.89]$), $+20.88$ ($[14.73, 26.67]$), and $+17.79$ ($[12.31, 23.74]$) points. All three seeds have positive interface differences on eight, eight, and seven structures, respectively. No best seed or best artifact is selected to define a new policy outcome.

D.1 Retained configurations on historical inputs

The twenty-input follow-up fixes the original interface inputs, seed 37, and the three controls after observing the eight-input comparison and relatedness audit. Forty new LASErMPNN/coding invocations use unchanged retained assets and drivers in alternating order, with 180 seconds per slot and no retry. Historical ADFLIP direct-function scores are bound before dispatch and referenced without prediction or rescoring.

The primary contrast is new LASErMPNN-minus-coding whole-chain recovery. Interface recovery, context effects, and historical ADFLIP comparisons are secondary. The historical arm contributes one output source per input; its two interfaces are not extra observations.

Contrasts retain complete-pair counts among twenty structures. The paired bootstrap uses 20,000 draws and seed 20260925. This single-seed follow-up stays separate from the eight-input three-seed study; local process times are not compared with historical agent time.

All forty new invocations pass admission and scoring, yielding 120 new control rows and 240 calls to the unchanged original metric. All forty exact repeats match; no slot is replaced or retried. Sixty historical H score rows are referenced with zero additional metric calls. LASErMPNN and coding scientific process totals are 751.01 and 203.82 seconds, respectively. These are local computation times, not reasoning-agent costs.

The primary whole-chain difference is -0.18 points, with a median of $+0.38$ and ten higher/ten lower LASErMPNN outcomes. The interface contrast is lower/equal/higher on 12/4/4 structures. Historical H minus coding is -1.39 points ($[-2.95, 0.15]$) for whole-chain recovery and -5.60 ($[-10.74, -0.60]$) for interface recovery. Table 6 retains all secondary contrasts; The numerical sup-

plement gives every input. Historical provenance and the fixed single seed qualify these comparisons independently of their intervals.

Table 6: **All configuration contrasts on the twenty original inputs.** K is newly invoked LASErMPNN, coding is newly invoked LigandMPNN, and H is historical ADFLIP. Mean differences are percentage points; intervals resample the twenty paired structures. Signs give negative/zero/positive structure differences. K-minus-coding whole-chain recovery is primary.

Contrast	Metric	Pairs	Mean (pp)	95% interval	Signs
K minus coding	Whole	20/20	-0.18	$[-2.00, 1.60]$	10/0/10
K minus coding	Interface	20/20	-3.37	$[-7.06, 0.36]$	12/4/4
Historical H minus coding	Whole	20/20	-1.39	$[-2.95, 0.15]$	12/1/7
Historical H minus coding	Interface	20/20	-5.60	$[-10.74, -0.60]$	12/5/3
Historical H minus K	Whole	20/20	-1.21	$[-2.84, 0.43]$	12/1/7
Historical H minus K	Interface	20/20	-2.22	$[-7.12, 2.47]$	9/3/8

E Input relatedness and scope of transfer

The input audit uses Biopython 1.85 global alignment, BLOSUM62, gap-open -10, gap-extension -0.5, identical terminal penalties, and tolerance 10^{-6}. Identity divides exact matches in the first optimal alignment by the longer sequence length. The 378 comparisons cover twenty interface and eight further inputs without altering their selection or scores.

At 80% identity, there are eight within-interface, four within-retention, and thirteen cross-group pairs, forming thirteen, five, and fourteen transitive components within the respective groups and their union. Seven further inputs have nearest earlier-input identity of 97.65–100%; 4m12 has 21.97%. These are descriptive sequence groups. Structure-bootstrap intervals retain the declared sampling unit.

F Validated execution evidence in the retained inventory

The twenty inverse-folding prompts share one scientific request, with distinct target identifiers, lengths, context counts, and file metadata. The model sees serialized descriptions, not coordinates or private sequences. Task-level resampling uses these instances; GeneBench contributes ten distinct released case texts.

The original inventory experiment shows that artifact presence can change routing, but its full manifest also carries earlier assessment language. A post-hoc content check selects all ten capability-available acquire responses, before considering their rationale content. Every response literally includes *backbone-only*, *HETATM*, and *discard*. These are textual observations, not ten correctness errors or proof of why the model chose acquisition. Original episode hashes bind the analysis.

Source inspection distinguishes two evidence scopes. The backbone-only parser finding applies to the previously retained ProteinMPNN configuration. The later accepted ADFLIP operation has independent evidence of nonprotein coordinates and element identities in its model features, actual registered execution, per-target seed reset, and a formal output. The initial manifest retained legacy caveat text from the earlier ProteinMPNN assessment. We therefore evaluate whether adding post-compilation execution evidence improves subsequent capability selection.

The evidence study fixes the same thirty tasks and three fresh decisions per task/condition, alternating condition order. Its 180 requests retain the earlier model, temperature, output ceiling, timeout, and no-retry rule. Only the manifest's `post_compilation_validation` field changes, adding validated facts without scores, sequences, answers, or a preferred action. Each control is newly observed.

The primary contrast averages the change in scheduled use frequency over twenty inverse-folding tasks, retaining failures in the denominator. Task unanimity and GeneBench are secondary. The manually assembled card changes content, length, and emphasis together; recommendations are not executed.

GeneBench exhibited a near-complete ceiling effect and was therefore excluded from the prospective component follow-up.

All 180 requests are charged with complete usage (numerical supplement). The provider returns `deepseek-flash` for requested `deepseek-v4-flash`. Raw has one output-budget exhaustion and one invalid JSON; supplemented has one invalid JSON. The card adds 780 prompt tokens per request.

The task-level increase is $+16.67$ points ($[8.33, 26.67]$): eight tasks increase and twelve tie. Unanimous tasks rise from eleven to nineteen. GeneBench gives unanimous use in both conditions on all ten cases.

F.1 Prospectively declared component follow-up

The component follow-up schedules 240 fresh decisions: twenty inverse-folding tasks, three observations, and four cards—raw (R), identity/history (S), execution record (E), and full (F). Prior observations motivate the study but do not enter its comparisons. GeneBench is excluded prospectively because of its earlier ceiling.

The two blocks are literal disjoint field subsets whose union reconstructs the original card. S carries artifact identity and the scope of the historical parser finding; E carries validated execution observations, updated claims, remaining limits, and evidence bindings. Raw/full prompts match the earlier study, with new requests. The blocks differ in length and contain overlapping scientific information.

Orders R, S, F, E; S, E, R, F; E, F, S, R; and F, R, E, S balance ordinal position and pairwise precedence. Model settings and the one-request/no-retry rule remain fixed; all slots close within the three-hour ceiling before analysis.

For task i, let u_{ic} be the percentage of its three scheduled observations recommending use under condition c, retaining failed or unstarted observations in that denominator. The two declared primary summaries average task-level block effects:

$$e_{S,i} = \tfrac{1}{2}[(u_{iS} - u_{iR}) + (u_{iF} - u_{iE})], \qquad e_{E,i} = \tfrac{1}{2}[(u_{iE} - u_{iR}) + (u_{iF} - u_{iS})]. \tag{1}$$

The interaction $u_{iF} - u_{iS} - u_{iE} + u_{iR}$ and simple contrasts are descriptive. Twenty thousand task-bootstrap draws retain all conditions and observations, using seed 20260925. Probability ceilings and overlapping content qualify interpretation of the block effects.

All 240 requests complete: 233 valid decisions and seven invalid-JSON responses with finish reason `stop`. In R, S, E, F order, use counts are 46/56/57/58, acquire counts 13/3/0/0, failures 1/1/3/2, and unanimous tasks 9/16/17/18.

Both average block effects are positive (Table 7). Full-minus-raw has eleven increasing tasks, eight ties, and one decrease. The interaction is -15.00 points; neither full-minus-single-block interval excludes zero. Every valid execution-only and full-card decision recommends use, so their net count difference comes from failures.

Table 7: **All prespecified component summaries and descriptive contrasts.** Differences are percentage points in scheduled use frequency. Intervals use paired task resampling. The first two rows are the declared average block effects; remaining rows are descriptive. Signs count tasks with negative/zero/positive differences.

Text-block contrast	Mean (pp)	95% interval	Task signs
Identity/history main effect	+9.17	$[3.33, 15.83]$	1/10/9
Execution-record main effect	+10.83	$[2.50, 19.17]$	3/7/10
Interaction	-15.00	$[-28.33, -1.67]$	8/10/2
Full minus raw	+20.00	$[8.33, 31.67]$	1/8/11
Identity/history minus raw	+16.67	$[6.67, 28.33]$	0/12/8
Execution-record minus raw	+18.33	$[6.67, 31.67]$	2/7/11
Full minus identity/history	+3.33	$[-5.00, 11.67]$	2/14/4
Full minus execution-record	+1.67	$[-5.00, 8.33]$	2/15/3

G PAIRED ROBUSTNESS AND COMPLETE INTERFACE RESULTS

The robustness analysis reads only closed interface and inventory records. Percentile intervals use 20,000 paired bootstrap draws, seed 20260925, resampling one structure or one decision task with its conditions and observations together. Intervals describe sensitivity within the fixed cohorts.

(a) Controlled interface comparison

	Direct Python	Registered MCP
Held fixed	20 structures × {context present, removed, repeat} ADFLIP function · weights · runtime · seed 37 One sample per condition; common sampling settings	
Changed together Interface + prompt/tool context	Python driver / arguments Direct-execution instructions	Named tool / input schema Registered tool descriptions

Figure 7: **Interface intervention and its fixed computation.** The generated operation, source, weights, runtime, inputs, and inference settings stay fixed. The invocation interface and prompt/tool context change together. Both conditions already have the generated packaging, so this experiment does not measure compiling a tool from scratch.

Table 8: **Sensitivity of final-interface paired differences.** Values are MCP minus direct. The leave-one-out column spans the twenty means obtained by dropping one structure. Tokens are thousands of provider-reported tokens; these observations have complete usage records.

Metric	Mean [95% interval]	Median	Leave-one-out
Requests	-2.20 [-3.50, -1.00]	-2.50	[-2.42, -1.84]
Native time (s)	-34.49 [-57.49, -14.38]	-7.20	[-37.09, -28.42]
With setup (s)	-34.57 [-57.52, -14.47]	-8.09	[-37.18, -28.50]
Prompt tokens (k)	-99.15 [-129.67, -72.78]	-77.78	[-102.95, -89.10]
Output tokens (k)	-1.14 [-1.60, -0.70]	-1.10	[-1.22, -1.00]

Mean request savings remain positive in both order strata: MCP minus direct is -2.00 requests on the ten direct-first structures and -2.40 on the ten MCP-first structures. Native-time differences are -37.93 and -31.05 seconds, respectively. Their descriptive intervals are $[-4.30, 0.00]$ and $[-3.60, -1.10]$ requests, and $[-74.38, -6.65]$ and $[-59.50, -7.60]$ seconds. Alternating order reduces one timing concern but does not randomize host activity or provider latency.

Adding external integrity/setup time gives mean totals of 181.47 seconds direct and 146.89 seconds MCP, a 19.1% reduction. Setup itself totals 538.89 and 537.34 seconds. The native-time median difference is -7.20 seconds, much smaller than the mean -34.49: a subset of harder direct invocations contributes disproportionately to average savings. Reported prompt tokens average 140,688.35 versus 41,541.45 (70.5% lower); output tokens average 4,470.45 versus 3,334.35 (25.4% lower). These costs include unsuccessful tool work and depend on the combined schema-and-context intervention.

Context increases, ties, or decreases whole-chain recovery on 17/2/1 structures and interface recovery on 15/3/2. Mean differences are $+1.79$ points (interval $[1.18, 2.41]$) and $+11.72$ points ($[6.62, 17.26]$). These diagnostics describe this retained configuration; they do not make context benefit an admission criterion. The two interfaces produce identical sequences, so their scientific outcomes are not independent replicates.

H DNA ACQUISITION AND CONDITIONAL RETAINED USE

H.1 SCIENTIFIC OPERATION AND ORIGINAL DATA

The DNA studies adapt three tasks from DART-Eval (Patel et al., 2024) around one learned operation: sequence likelihood. The three likelihood probes request a value for every unlabelled 350-base

DNA sequence. They name no model, repository, benchmark answer, or sequence-pair role. Each sequence has a random opaque identifier, and input order is shuffled. A released pretrained model must perform inference without training or fine-tuning. Masked models sum the original-token log probabilities obtained by masking one nonspecial DNA token at a time; causal models sum teacher-forced next-token log probabilities using their documented beginning-token convention. Both use natural logarithms, evaluation mode, and seed 37. Padding and special tokens are excluded. Averaging by token count, truncation, averaging orientations, and heuristic replacement are outside the requested operation.

Regulatory element discrimination (AC). The fixed subset contains 128 element/control pairs on each of DART-Eval's six test chromosomes: 5, 10, 14, 18, 20, and 22. Chromosome 5 supplies the initial task; the other five batches are reserved before dispatch. Within each chromosome, a fixed hash ordering selects canonical-ACGT inputs with nonoverlapping genomic windows. All 768 pairs are independently reconstructed using the original dataset class, original ENCODE coordinate file, GRCh38 reference, and original random transformations. Every reconstructed element and dinucleotide-shuffled control agrees with the released data transport. The 1,536 sequences are unique, including reverse complements; all pairs have matching mono- and dinucleotide counts. Thus elementary composition cannot distinguish the paired labels.

An input-only audit calculates global unit-edit distance between all 294,528 positive-sequence pairs, taking the smaller distance over forward and reverse-complement orientations. The descriptive similarity is $1 - d/350$, distinct from alignment-column identity. No pair reaches the prespecified 80% threshold: the smallest within-batch distance is 92, the smallest between-batch distance is 100, and the smallest reserved-to-initial distance is 104.

Motif footprinting (AD). A second, prospectively reserved task uses the same retained likelihood operation on original DART-Eval motif insertions. Preparation reproduces all 577,400 released rows byte for byte with the unchanged motif generator and HOCOMOCO v12 input, after independently reconstructing the 100 neutral backgrounds through the original genomic dataset class. A fixed hash ordering selects sixteen motifs and sixteen backgrounds, with both original orientations. This gives 512 motif/shuffled-motif pairs and 32 neutral sequences, or 1,056 sequences. None overlaps an AC input exactly or by reverse complement. The original shuffle matches mononucleotide counts, while dinucleotide counts differ in 506/512 pairs; Backgrounds and orientations are shared observations, rather than independent biological replicates.

Regulatory variant discrimination (AE). A third probe asks whether allele-likelihood changes distinguish the original African caQTL labels from controls. It fixes 512 variants by a score-independent hash order within the released original-use and peak-eligible population, without class balancing: 35 positives and 477 controls. The unchanged original variant dataset class reconstructs all 2,114-base reference/alternate windows exactly. We retain the central 350 bases, placing the differing nucleotide at zero-based position 175, to reuse the accepted operation unchanged. Windows are canonical, autosomal, nonoverlapping with AC and one another, and contain no exact or reverse-complement AC/AD sequence. The resulting 1,024 sequences have opaque, shuffled identifiers.

Labels are copied unchanged from a version-pinned public DART-Eval redistribution. Original Synapse label-file access returned HTTP 403, so genomic reconstruction verifies sequence transport but does not independently verify the raw label TSV. The shorter window and whole-sequence likelihood difference are declared adaptations; some published Task-5 model runs instead score the mutated token.

H.2 Agent schedule, retained objects, and admission

Both initial AC arms receive one A40 GPU, one CPU thread, 720 generation requests, and 10,800 seconds, including search, installation, failed phases, compilation, and inference. Full acquisition first makes an inventory-aware decision from an empty scientific inventory. Its acquire branch enters the existing discovery/compiler/native-Biomni pipeline. Coding can acquire any applicable official pretrained artifact and retain its own helper, runtime, tokenizer, and weights. Different artifact choices are observed outcomes of the two routes.

Each of the five reserved AC batches permits 30 requests and 1,800 seconds per arm. The full route receives a manifest built from its actual initial execution receipt, with scientific accuracy still withheld. Its decision must select the retained capability before native Biomni registration and invocation occur. A different recommendation remains recorded and is not overridden. Coding receives its own frozen cache and executes its unchanged helper with new input/output arguments. Fresh workspaces exclude earlier task inputs, outputs, and predictions. Source, weight, tokenizer, and runtime membership and bytes are checked before and after retained use.

AD and AE are conditional on all twelve AC slots terminating and independent admission of each arm's initial scientific execution. Eligibility uses execution evidence, without any task-score threshold. Each eligible arm receives 30 requests and 3,600 seconds for one further retained invocation. Acquisition, package installation, helper edits, settings searches, and repairs are unavailable in this phase. An unavailable initial artifact leaves its scheduled slot unavailable. AE follows AD worker termination to avoid GPU contention; its eligibility does not depend on AD performance. Together, the studies distinguish acquiring an operation, executing it on reserved genomic inputs, and applying it to different biological probes.

Before scoring, programmatic checks and author-supervised trace inspection verify the selected official artifact, locked scientific source, checkpoint/tokenizer identity, actual weight reads, learned forward computation, likelihood convention, and provenance of the unique formally submitted file. Compilation tests and successful MCP registration are necessary engineering evidence, but cannot alone establish the scientific computation. Failed attempts and intermediate phases remain in the cost records. A terminal timeout or unsubmitted file cannot be rescued by finding a plausible prediction in its workspace.

Explicit engineering interventions. AC's full route compiled and independently validated its selected artifact, but its initial registered call failed before reading learned weights: the isolated runtime omitted the system path needed by the CUDA library loader.

After zero-prediction diagnostics verified the corrections, explicit authorization allowed one separate continuation per failed initial route. AC adds only read-only system loader paths and permits one further registered call within 32 requests/1,800 seconds; independent admission is required before its original five reserves and two biological probes. Original task data, selected scientific source/checkpoint, and scorers remain fixed. Coding episodes are preserved. Intervention outcomes are reported separately, all original and additional costs are charged, and neither continuation counts as an independent acquisition replicate. No further inference repair or intermediate-output salvage is permitted. A separately authorized zero-call submission intervention handles only a completed native final that explicitly lists the unique successful MCP prediction together with extra metadata files. It removes metadata paths from a derived envelope, preserving every file byte, the original failed terminal, and all costs. Independent scientific audit remains required. These manually organized submissions and all retention conditional on them are reported separately from uninterrupted autonomous outcomes.

H.3 Observed scientific implementations

AC's full route selected the official Evo 2 7B artifact (Brixi et al., 2026); coding selected the official human-reference 500M Nucleotide Transformer (Dalla-Torre et al., 2025). These artifacts share a scoring interface: Evo 2 returns a causal sequence log-likelihood, while Nucleotide Transformer returns a masked-token pseudo-log-likelihood. The Evo 2 wrapper calls the unchanged official causal scoring function with a beginning token, batch size sixteen, and a sum over all 350 target bases. The recorded forward precision is bfloat16; native log-softmax precedes float32 summation. The optional Transformer Engine was absent, and the official implementation used its bfloat16 projection fallback. Nucleotide Transformer uses the official tokenizer with exact sequence reconstruction, masks each of sixty nonspecial tokens once, and sums their original-token log probabilities in float32 inference. Thus raw values across the models have different interpretations; only each original benchmark's within-model contrasts are compared.

H.4 ORIGINAL SCORING AND DESCRIPTIVE ANALYSIS

All AC, AD, and AE slots terminate and their independent execution audits close before scientific scores are released. AE was declared after the initial coding submission and knowledge of the selected artifacts, while all biological scores remained withheld; AC and AD criteria are unchanged. AC replays the sealed finite values, bound by opaque identifier and sequence bytes, through the unchanged original `ZeroShotPairedControlEvaluator.evaluate`. This is a declared prediction-interface adaptation using float64 JSON values; it is not a bitwise reproduction of every original model's inference. The original evaluator supplies paired accuracy, signed-rank outputs, and likelihood-difference summaries. Its primary accuracy requires element likelihood to strictly exceed control likelihood, so exact ties are incorrect. Raw likelihood magnitudes are not compared across different model tokenizations.

AC reports each fixed chromosome, all-six and reserved-five means, and full-minus-coding differences wherever both original scorers yield outcomes. Descriptive 95% percentile intervals use 10,000 paired bootstrap draws within each fixed chromosome, seed 37. The resampling preserves the paired arm observations and chromosome sizes; sequence pairs do not multiply the number of acquisition episodes. Cumulative prefixes include actual initial acquisition and subsequent use costs.

A secondary annotation summary was declared after AC dispatch but before any scientific score release. It groups the existing original Task-1 correctness indicators by all eight unchanged ENCODE annotation labels, reporting scheduled and scored denominators, each arm's mean, and the difference on complete paired observations. It introduces no new labels, scorer, subset selection, subgroup significance test, or change to the primary analysis.

AD restores the sealed values to the exact original row order and calls the unchanged motif-footprinting CLI. All sixteen original per-motif accuracies and their unweighted mean are reported, alongside its background comparison and signed-rank values. This scorer uses ≥ 0, giving ties full credit; tie counts therefore accompany interpretation. Its convention is preserved rather than harmonized with AC. The synthetic motif probe measures the model's likelihood preferences, without establishing regulatory function. Original scorer warnings and nonfinite secondary outputs remain in the archived raw results.

AE restores the two sealed allele values and original labels, then calls the unchanged original `sig_ctrl_variants_Afr_CaQTLs`, `compute_change`, and `get_precision_recall_auc` functions. Only the module's hard-coded data directory is configured to the bound subset. The ranking score is $|s_{\mathrm{alt}} - s_{\mathrm{ref}}|$, the absolute difference between the two whole-sequence log scores. The original functions compute AUROC and average precision, reported as AUPRC in DART-Eval. Both metrics are descriptive point estimates over all 512 fixed variants; positive prevalence is $35/512 = 6.84\%$. Exact zero-change counts accompany them.

H.5 CLOSED LIKELIHOOD RESULTS

We report original and intervention-assisted outcomes separately. Table 9 summarizes all fixed regulatory batches. Sixteen unchanged original scorer calls produce the two sets of task outcomes. Original-only and assisted numerical exports remain separate, with 314 and 476 independent reconciliation checks respectively. These checks compare original metrics, paired indicators, bootstrap summaries, variant rank metrics, and request totals; they do not substitute for execution audit.

Figure 3(a,b) shows the regulatory, motif, and variant outcomes. These are task outcomes from different selected predictors, not independent acquisition replications.

Figure 8(a) shows cumulative request costs: 346 for the full route, including failed acquisition and authorized continuation, versus 89 for coding. The five reserves use 45 versus 60 requests; this lower incremental cost does not offset the larger first-use investment within the observed sequence. Manual effort and incomplete timing are outside this request measure.

On the 640 reserved pairs, assisted full and coding are uniquely correct on 126 and 13 pairs, respectively; both are incorrect on 11. Higher aggregate accuracy therefore coexists with coding-only successes. This post-hoc overlap characterizes the retained predictors; no selection or ensemble policy is evaluated.

Table 9: **Every fixed regulatory batch and paired error overlap.** Full denotes the separately assisted Evo 2 configuration; coding denotes the original Nucleotide Transformer configuration. Differences are percentage points, with paired within-chromosome percentile intervals. Initial full requests include the failed acquisition and authorized continuation; projections add no model requests. The lower panel cross-tabulates original pair-correctness indicators after study closure.

Chromosome	Phase	Full (%)	Coding (%)	Full req.	Coding req.	Paired Δ [95% CI]
chr5	Initial	95.31	86.72	301	29	$+8.59$ [2.34, 14.84]
chr10	Retained	96.09	79.69	10	12	$+16.41$ [9.38, 24.22]
chr14	Retained	95.31	78.12	8	10	$+17.19$ [10.16, 25.00]
chr18	Retained	94.53	79.69	14	12	$+14.84$ [7.81, 21.88]
chr20	Retained	96.09	78.12	7	14	$+17.97$ [10.16, 25.78]
chr22	Retained	99.22	77.34	6	12	$+21.88$ [14.84, 29.69]
Reserved mean		96.25	78.59	–	–	$+17.66$ [14.38, 21.09]
All-batch mean		96.09	79.95	–	–	$+16.15$ [13.28, 19.14]

Paired overlap	N	Both correct	Only full	Only coding	Both incorrect
Initial	128	108	14	3	3
Reserved	640	490	126	13	11

The descriptive annotation analysis retains all eight unchanged ENCODE classes, with sizes from 14 to 476. These groups define neither new primary comparisons nor independent acquisition observations.

Table 10: **Regulatory variant outcomes.** Both routes use their unchanged retained likelihood operation. AUPRC is the original scorer's average precision. All 512 variants remain, with positive prevalence 6.84%.

Arm	AUROC	AUPRC	Positive / total	Zero change	Requests	Scored
Full + interventions	0.5208	0.0751	35 / 512	1	10	1/1
Original coding	0.5037	0.0764	35 / 512	0	13	1/1

H.6 Same-function DNA interface replication (AH)

AH assigns the unchanged retained Evo 2 likelihood function to twelve fresh native Biomni A1 episodes, requesting GLM-5.3 in both conditions. The six 128-pair chromosome batches are the existing AC inputs. Their earlier scientific scores were already known to the experimenter when AH was declared; workers receive neither earlier predictions nor pair labels. This is a new invocation study on previously scored inputs, rather than an additional acquisition or independent biological validation.

All twelve attempts terminate before independent execution admission and original scoring. Source/runtime/weight membership and bytes are checked around each attempt. Admission checks the assigned invocation, actual source/weight/GPU/input/output access, all prediction-file writes, and the unchanged unique file explicitly submitted by the native episode. A separate inspection reads every executed command. Exact equality of parsed ordered identifier/likelihood records and maximum absolute difference after matching identifiers are reported on admitted pairs; numerical inequality is not silently rounded away or treated as execution failure. Each admitted submission is replayed once through the unchanged original Task-1 scorer.

Primary costs are all generation requests and agent-attempt elapsed seconds, including native startup, retrieval, scientific execution, and receipt handling. Original acquisition and external integrity checks are excluded; worker controller elapsed, token totals, and missing usage remain separate. Costs retain failed attempts independently of scientific admission. Paired means, medians, and 95% percentile intervals use 10,000 resamples of the six batch-level MCP-minus-direct differences, seed 37. The three direct-first and three MCP-first groups are also reported separately. The 1,536 sequences do not become independent interface replicates, and protein and DNA costs are not pooled.

MCP and direct use 50 and 55 requests in total, with mean agent-attempt times 167.62 and 131.91 seconds. The paired request mean is -0.83 (95% interval $[-2.83, 0.83]$), median -0.50; the elapsed

Table 11: **DNA interface replication: every scheduled batch.** Accuracy is shared when both admitted arms agree; the chromosome-20 cell is direct/MCP, with the rejected direct output unscored. Cost differences are MCP minus direct across all six pairs. Scientific missingness does not remove observed cost.

		Requests		Agent-attempt s			
Chr.	First	Direct	MCP	Direct	MCP	Accuracy (%)	Exact values
5	Direct	8	9	120.86	152.34	95.31	Yes
10	MCP	10	8	107.89	114.46	96.09	Yes
14	Direct	8	7	135.57	97.67	95.31	Yes
18	MCP	10	10	100.47	189.03	94.53	Yes
20	Direct	10	5	176.38	134.76	– / 96.09	–
22	MCP	9	11	150.27	317.42	99.22	Yes
Paired mean Δ		-0.83		35.71		–	–
Paired median Δ		-0.50		19.03		–	–

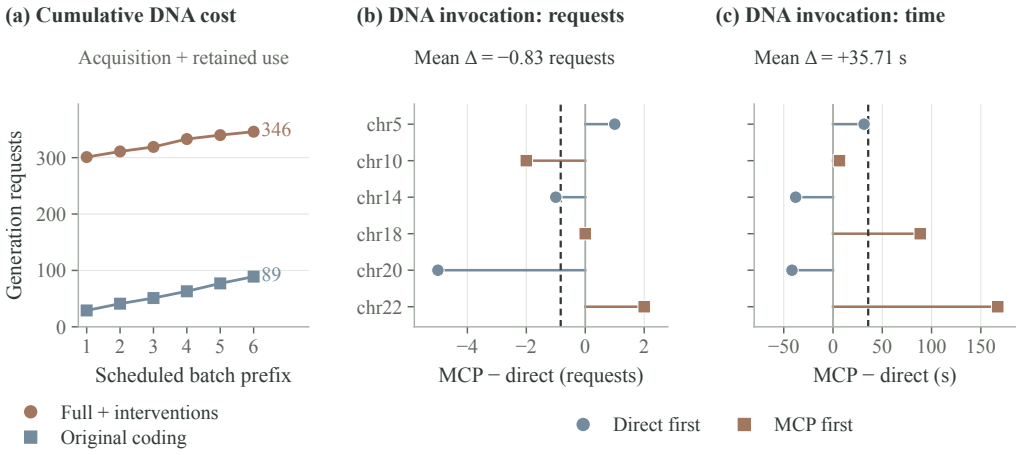

Figure 8: **DNA acquisition and invocation costs.** (a) Cumulative generation requests over the six regulatory batches (AC), including failed initial phases and authorized continuations. The retained Evo 2 configuration includes interventions; the comparator is the original coding-selected Nucleotide Transformer. Zero-call submission cleanup adds no request; manual effort and incomplete timing are outside this measure. (b,c) A separate same-function study (AH) shows MCP-minus-direct requests and agent-attempt seconds for all six batch pairs, including the pair with an unscored direct output. Five admitted pairs have identical likelihoods. Panels share chromosome rows; markers identify the first route and dashed lines mark paired means. Negative values favor MCP. These invocation costs exclude acquisition and external integrity checks, but include exploratory work and uncontrolled cache effects; both mean intervals span zero.

mean is $+35.71$ seconds ($[-19.93, 96.65]$), median $+19.03$. Direct-first batches have mean differences of -1.67 requests and -16.01 seconds, whereas MCP-first batches have 0.00 requests and $+87.43$ seconds. Each order group contains only three batches and different scientific inputs; this does not identify an order effect. Prompt-token totals are 360,833/466,545 and completion-token totals 55,746/39,495 (MCP/direct), with no missing usage.