RSI for Science: A Verifier-First Framework for AI Scientists

Anonymous Authors[1]

Abstract

AI scientists are increasingly framed as recursive self-improvers: systems that generate hypotheses, choose experiments, revise tools, store lessons, and improve future campaigns. We argue that this language is misleading for empirical science unless recursion is externally verifier-governed. In empirical domains, the verifier is not a compiler, theorem checker, or game rule engine; it is a noisy, delayed, costly, and sometimes destructive physical, statistical, or procedural test. We propose *Verifier-Governed Recursive Scientific Refinement* (VGRSR): a standard that credits scientific recursion only when reusable objects of the scientific search process—hypotheses, tools, verifiers, memory, and campaign policy—change through cost-accounted cycles that pass independent gates. VGRSR adds four requirements to AI-scientist evaluation: external disconfirmation gates, per-cycle provenance, path-sensitive stability metrics, and verifier-efficiency accounting. We support the position with mechanism demonstrations of proxy drift and latency starvation, a five-object taxonomy, a control/risk framing using sensitivity, maximum drawdown, and drawdown CVaR, and vignettes spanning materials, cosmology, nanobiomaterials, and neurodegeneration. The practical standard is simple: no external gate, no cost-accounted cycle log, no stability report, no scientific RSI claim.

1. Position: Scientific Recursion Requires External Governance

Autonomous discovery systems now span self-driving laboratories, AI co-scientists, robotic chemistry, active-learning materials platforms, and execution-grounded scientific agents (King et al., 2009; Hase and Aspuru-Guzik, 2019; Burger et al., 2020; MacLeod et al., 2020; Abolhasani and Kumacheva, 2023; Szymanski et al., 2023; Tom et al., 2024; Ren et al., 2025; Wei et al., 2025). The ICML 2026 AI Scientists workshop explicitly asks for shared vocabulary, evaluation criteria, and governance for systems that move along the tool–co-author–founder spectrum. This paper's position is that recursive self-improvement (RSI) in empirical science should be replaced by a narrower and more testable object: VGRSR.

Why the distinction matters. Iterating a prompt, re-ranking generated candidates, training on synthetic traces, or asking one model to critique another may improve a local output (Madaan et al., 2023; Shinn et al., 2023; Valmeekam et al., 2023; Huang et al., 2023; Kamoi et al., 2024). It does not by itself establish scientific self-improvement. The unit of scientific recursion is not an answer revision; it is a reusable process component that changes future search. We therefore define a verified scientific cycle as a tuple

$$z_k = (o_k, \Delta o_k, g_k, y_k, c_k, \ell_k, r_k),$$

where o_k is the object changed, Δo_k is the proposed update, g_k is the independent gate, y_k is externally verified utility, c_k is cost, ℓ_k is wall-clock latency, and r_k is a path-risk state. A recursive claim is credible only if these tuples form an auditable trajectory rather than a success story assembled after the fact.

Central claim. AI-for-science systems should not be called recursive self-improvers unless their multi-cycle improvements are externally verifier-gated, cost-accounted, stable across cycles, and auditable from per-cycle records. "Refinement" is deliberate: empirical discovery can accelerate, but only through falsifiable, budgeted updates that survive independent tests. Internal reflection is useful; it becomes *scientific* evidence only when a verifier capable of saying no admits the update (Denison et al., 2024; Wang et al., 2026; Luo et al., 2025; Zhu et al., 2025).

2. Why Ungoverned RSI Fails in Empirical Science

2.1. Internal Self-Correction Is Circular Evidence

Ungoverned RSI collapses proposal, scoring, critique, and promotion into one coupled model family or proxy stack. That circularity is tolerable when the verifier is exact: a compiler rejects invalid code, a theorem checker rejects

[1]Anonymous Institution, Anonymous City, Anonymous Region, Anonymous Country. Correspondence to: Anonymous Author <anon.email@domain.com>.

Submitted to the AI for Science workshop (ICML 2026).

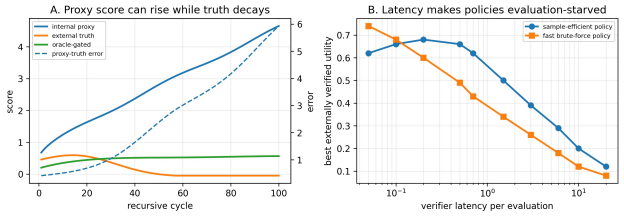

Figure 1. Two mechanism demonstrations. Left: proxy score rises while externally measured truth decays and proxy-truth error grows. Right: verifier latency turns proposal speed into evaluation starvation.

invalid proof steps, and a board-game engine applies fixed rules (Silver et al., 2018; Trinh et al., 2024). Science is different: assays fail, simulators are misspecified, posterior geometries are sloppy, measurements are heteroscedastic, and repeated use of one evidence family inflates confidence (Fatehi et al., 2023; Dunlap et al., 2024; Thelen et al., 2023; Chen et al., 2026).

Two drifts follow. *Epistemic drift* occurs when the target shifts from the scientific objective to whatever the loop can most easily improve internally. *Procedural drift* occurs when acceptance criteria, filters, data provenance, or sampling policy change in ways that inflate apparent progress. Reward-hacking theory gives the mechanism: compressed evaluators create equivalence classes; optimization amplifies blind spots; evaluator–policy co-adaptation turns the evaluator into another manipulable object (Goodhart, 1975; Manheim and Garrabrant, 2019; Denison et al., 2024; Wang et al., 2026; Gabor et al., 2025; Cao et al., 2026). Once the loop learns features of the oversight process, internal scores can rise while external validity falls.

2.2. Oracle Regress and Latency Starvation

Figure 1 gives two minimal mechanism demonstrations. In the proxy-optimization simulation, a learned Gaussian-process reward model is optimized in a frozen inner loop. The internal proxy rises monotonically while the true objective peaks early and decays; an oracle agent using the same optimizer but direct access to truth remains stable. The collapse is caused by proxy exploitation, not by the search heuristic. In the latency sweep, a sample-efficient policy initially beats brute force, but both policies become evaluation-starved as verifier latency consumes the wall-clock budget. Faster inner-loop generation does not create verified progress when the gate dominates time.

The strongest objection to physical gating is speed: why wait for an assay, synthesis, MCMC run, docking computation, or high-fidelity simulator when a neural surrogate can generate and score candidates rapidly? The answer is that surrogates exchange wall-clock latency for fidelity risk. Multi-fidelity Bayesian optimization, neural emulators, low-fidelity screens, and uncertainty-guided triage are essential (Forrester et al., 2007; Kandasamy et al., 2017; Li et al., 2020; Astudillo et al., 2021; Li et al., 2026; Do et al., 2023; Zhao et al., 2025). They cannot silently become the universe being optimized.

Table 1. VGRSR objects and minimum reporting obligations.

Object	Required gate	Main metric
Hypotheses	External assay, calibrated simulator, causal/statistical test, or held-out measurement	Verified utility; FDR
Tools	Value-of-information check and fidelity escalation	Calibration; gain/cost
Verifiers	Independence, calibration, drift audit, and disagreement log	FA/FR; latency
Memory	Provenance and redundancy checks; rejected-case retention	Coverage; contamination
Policy	Tail-risk monitor; rollback, pause, or replication rule	MDD; DD-CVaR

2.3. Physical Reward Latency Is a Structural Bottleneck

Empirical science receives decisive feedback through experiments, high-fidelity simulations, downstream replication, or human expert gates. These signals are slow, sparse, expensive, noisy, and often failure-prone. Self-driving-lab surveys emphasize expensive actions, delayed/noisy observations, strict feasibility constraints, nonstationarity, drift, and provenance requirements (Chen et al., 2026; Adesiji et al., 2025; Lee et al., 2026; Kitchin, 2025). AI-scientist benchmark papers similarly identify hidden workflow failures such as data leakage, metric misuse, post-hoc selection, and weak execution capability (Luo et al., 2025; Zhu et al., 2025; Hu et al., 2025; Luo et al., 2025b). The imbalance is therefore not "ideas versus no ideas"; it is cheap generation versus scarce contact with reality.

3. VGRSR: Five Objects, Gates, and an Audit Trail

VGRSR rejects monolithic RSI language and asks what changes recursively. The state of an AI scientist at cycle k is

$$S_k = (\mathcal{H}_k, \mathcal{T}_k, \mathcal{V}_k, \mathcal{M}_k, \pi_k),$$

where $\mathcal{H}$ is a hypothesis set, $\mathcal{T}$ is a tool/model stack, $\mathcal{V}$ is a verifier stack, $\mathcal{M}$ is provenance-bearing memory, and π is campaign policy. A proposed update is promoted only if

$$S_{k+1} \leftarrow \mathrm{Promote}(S_k, \Delta S_k) \quad \text{when} \quad \mathrm{Gate}(\Delta S_k; \mathcal{V}_k) = \texttt{pass}.$$

Table 1 gives the minimum reporting obligations. Figure 2 shows the execution loop.

Four design patterns. *Verifier-gated expansion* requires propose–verify–commit-or-reject cycles with calibrated abstention (Geifman and El-Yaniv, 2017; 2019; Fisch et al.,

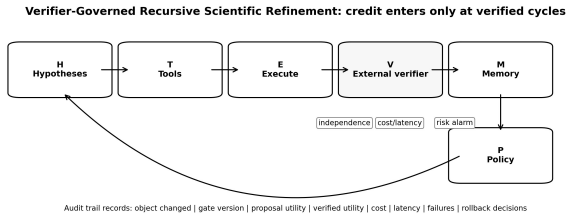

Figure 2. VGRSR execution loop. Scientific credit enters only at verified cycles: propose a change; execute; verify; record cost, latency, failure, and risk alarms; commit, rollback, or escalate.

2022; Koenighofer et al., 2023). *Cost-aware fidelity escalation* promotes candidates only when expected value of information exceeds marginal verification cost (Letham et al., 2019; Astudillo et al., 2021; Schoepfer et al., 2024; Li et al., 2026). *Reflective failure memory* stores rejected and anomalous cases rather than training only on successes (Shinn et al., 2023; Schaul et al., 2015; Zhang et al., 2020; Wu et al., 2025). *Admissibility rules for correlated evidence* prevent simulator-family outputs, duplicated measurements, or shared training data from being counted as independent confirmation.

External gates. A gate is eligible for scientific credit only when it is capable of disconfirmation, temporally downstream of the proposal, versioned, and sufficiently independent that a failure cannot be silently rewritten as success. A physical assay usually qualifies. A simulator can qualify only if its lineage, calibration data, error model, and coupling to the proposer are disclosed. A co-trained emulator, LLM judge, docking score, or reward model can rank or triage, but it should be reported as a proxy unless independence and calibration are documented.

4. Control, Risk, and Gate-Aware Acquisition

A scientific recursion loop is a feedback system. Let the environment be plant P, the campaign policy be controller C, and the verifier be a noisy delayed sensor. With loop transfer $L = PC$,

$$S = (I + L)^{-1}, \qquad T = L(I + L)^{-1}, \qquad S + T = I.$$

The identity is algebraic, but it captures the trade-off that matters in scientific campaigns: a policy that rejects disturbances aggressively can transmit verifier noise, amplify delay artifacts, or lose robustness to mismatch. Robust-control practice supplies stress tests for sensitivity, margins, delay, and model error (Zhou et al., 1996; Skogestad and Postlethwaite, 2005; Doyle et al., 1989; 1990; Ruth and Weller, 2010; Jansen et al., 2020).

Average performance is insufficient because recursive failures are path-dependent. For verified performance P_k, let $P_k^* = \max_{j \le k} P_j$, drawdown $D_k = (P_k^* - P_k)/P_k^*$, maximum drawdown $\mathrm{MDD} = \max_k D_k$, and DD-CVaR be the mean of the worst tail of drawdowns. These alarms borrow from risk management but target scientific instability: regressions, oscillations, rollback episodes, and costly recovery cycles (Rockafellar and Uryasev, 2000; Artzner et al., 1999; Pflug, 2000; Ruszczynski, 2010; Nguyen et al., 2021; Kouri and Shapiro, 2021). Static CVaR can be time-inconsistent in sequential settings, so VGRSR reports rolling windows, bootstrap thresholds, hysteresis, and a named response: pause, rollback, replicate, or escalate.

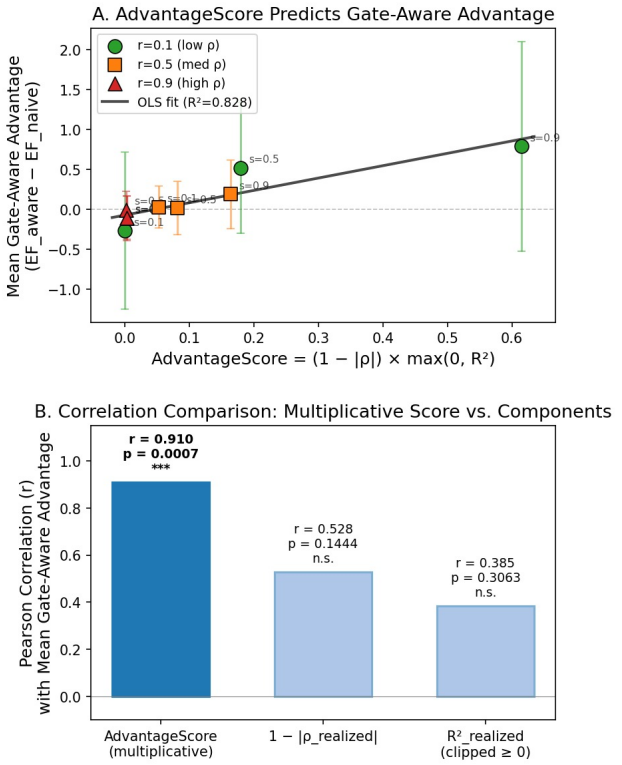

Figure 3. ADVANTAGESCORE as a gate-selection heuristic. Predictable, non-redundant gates create the largest gate-aware advantage; duplicate or unpredictable gates do not.

Gate-aware acquisition. A gate is worth modeling only if it is learnable and non-redundant. Let ρ be target–verifier correlation and R^2 be cross-validated verifier predictability. We use the diagnostic

$$\mathrm{AdvantageScore} = (1 - |\rho|)\max(0, R^2).$$

High predictability alone is insufficient because a duplicate verifier adds little. Low correlation alone is insufficient because an unpredictable gate adds noise. In the synthetic calibration grid in Figure 3, the multiplicative score predicts gate-aware advantage better than either component, but it is a screening heuristic, not a theorem. It should be bootstrapped and evaluated conditional on gates already in use.

Table 2. Vignette boundary conditions. Each row supports a methodological claim rather than a leaderboard claim.

Domain	What VGRSR tests	Interpretation
Mat.	Feasibility-gated acquisition	Gates help in moderate regimes when they are learnable and non-redundant.
Cosmo.	Emulator refinement under sloppy posterior geometry	Conditioning and dimension bound recursive improvement.
NanoBio	Multi-fidelity docking under wall-clock limits	Latency creates sample starvation under small budgets.
Neuro	Reject-memory and verifier architecture	Retaining rejected cases reduces saturation and exposes boundaries.

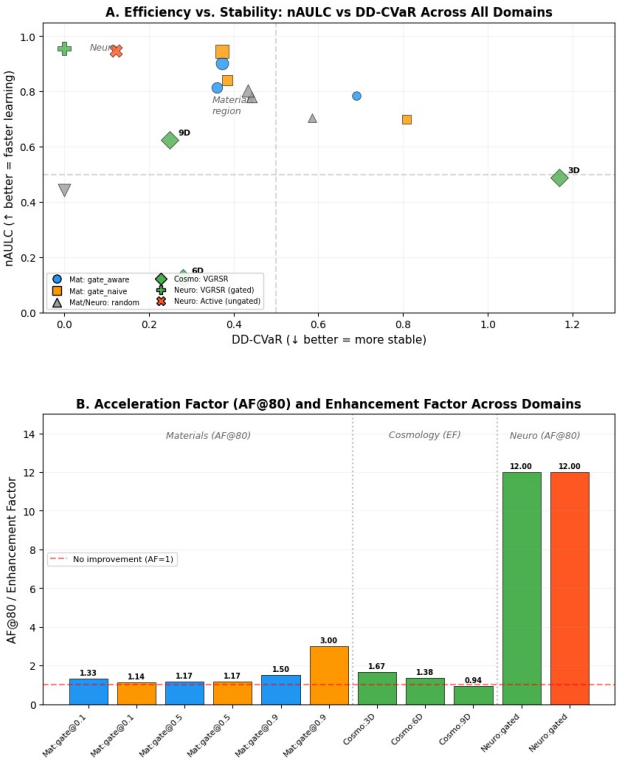

Figure 4. Efficiency and stability diagnostics across vignettes. VGRSR reports both learning efficiency and path-risk rather than only final best score.

5. Evidence Vignettes and Boundary Conditions

The evidence should be read as failure mapping and stress testing, not as a universal leaderboard. In materials optimization, AFLOW-like constrained acquisition illustrates when probabilistic feasibility gates improve efficiency under formation-energy or stability constraints (Curtarolo et al., 2012; Jain et al., 2013; Liang et al., 2021; Hickman et al., 2023). In neurodegeneration, UPDRS-like prediction shows that memory should include rejected proposals: accepts-only learning saturates, whereas accepted-plus-rejected training exposes the verifier boundary (Tsanas et al., 2010; Little et al., 2009; Mitchener et al., 2025). In cosmology, posterior sloppiness and high-dimensional MCMC show that recursive emulator refinement is bounded by conditioning and simulation latency (Lewis et al., 2000; Lewis, 2013; Asmussen et al., 2021; Borrett et al., 2026). In nanobiomaterials, peptide docking and multi-fidelity screens show a small-budget regime where latency creates sample starvation (Trott and Olson, 2010; Jumper et al., 2021; Abrams et al., 2023; Zhao et al., 2025).

Figure 4 summarizes the efficiency–stability trade-off. The important pattern is not that any single gate always wins. It is that gate settings determine which kind of failure is visible: permissive gates can find higher peaks while increasing drawdown; conservative gates can stabilize trajectories while capping optima; mismatched verifiers can saturate unless rejected cases are retained.

6. Benchmarking VGRSR

A VGRSR benchmark should require multiple cycles, allow updates to reusable workflow components, enforce external gates before promotion, and score verified utility on a cost axis. The minimal metric set is:

$$\mathrm{VUGC} = \frac{d\,\mathbb{E}[U_k^{\mathrm{verified}}]}{d\,\mathrm{cost}_k},$$

$$\mathrm{AF@}\tau = \frac{T_{\mathrm{baseline}}(\tau)}{T_{\mathrm{method}}(\tau)}, \qquad \mathrm{EF}(B) = \frac{U_{\mathrm{method}}(B)}{U_{\mathrm{baseline}}(B)}.$$

Here $T(\tau)$ is time, cost, or experiment count to reach a target fraction of verified improvement. VUGC credits only externally verified gain. AF and EF extend standard self-driving-lab metrics (Adesiji et al., 2025; Liang et al., 2021). MDD and DD-CVaR report instability. Verifier efficiency reports validated gain per confirmatory assay, human escalation, or high-fidelity check. Transfer reports whether memory, policy, or abstractions help new campaigns without weakening gates (Ma et al., 2024; Yang et al., 2023; Song et al., 2024; Wang et al., 2025; Ni et al., 2026; Fan et al., 2025).

Task families. Useful benchmarks include: (i) simulation-to-real self-driving-lab campaigns with delayed noisy outcomes; (ii) theory–experiment coupling; (iii) adaptive causal discovery; (iv) multi-fidelity assay selection; and (v) drifted adaptive measurement. These tasks distinguish answer iteration from process-level recursion.

Required ablations. Remove external gates, remove prove-

Algorithm 1 Verifier-Governed Recursive Scientific Refinement

Require: Initial state $S_0 = (\mathcal{H}, \mathcal{T}, \mathcal{V}, \mathcal{M}, \pi)$, budget B
for $k = 0, 1, \ldots$ until budget exhausted **do**
  Propose update ΔS_k and candidate action a_k under policy π_k
  Execute proxy/low-fidelity checks; estimate cost, uncertainty, and risk
  Escalate to independent gate g_k when value of information justifies cost
  Record $(\Delta S_k, g_k, y_k, c_k, \ell_k, r_k)$ in append-only cycle log
  if g_k passes and risk alarms are below thresholds **then**
    Commit update; retain accepted and rejected evidence in memory
  else
    Reject, rollback, replicate, or request a higher-fidelity gate
  end if
end for

nance, freeze campaign policy, optimize proxy-only rewards, keep verifier allocation static, and remove drift/noise stress. A credible benchmark should show that gate removal may increase internal scores while reducing external validity. It should also include hidden holdouts, negative controls, and trace-log audits to detect evaluator–policy co-adaptation (Hrubec, 2026; Cao et al., 2026; Luo et al., 2025; Hu et al., 2025).

7. Alternative Views and Reviewer Standard

Self-correction already counts as RSI. Reflection, debate, reranking, and tool repair can improve outputs. They are insufficient for scientific RSI unless their gains are externally verified. A model does not get to grade its own exam and cite the grade as evidence that it learned science.

Synthetic data and self-play can replace gates. This is plausible in formal domains where the verifier is exact and cheap. Empirical science lacks that luxury. Synthetic data can improve proposal generation, but without external tests it can accelerate self-consumption, simulator overfitting, and proxy drift (Shumailov et al., 2024; Alemohammad et al., 2024; Wang et al., 2026).

Strict gates suppress novelty. Gates can be too conservative. VGRSR does not ban risk; it budgets risk, reports drawdown, logs rejected candidates, and makes exploration policy auditable. Exploration becomes stronger when reviewers can inspect what failed.

Audit standards slow the field. The overhead is proportional to the claim. A low-risk assistant can provide light provenance. A self-driving laboratory claiming autonomous recursive discovery should provide cycle logs, verifier versions, costs, rejection memory, and stability traces. A system operating with hazardous materials, clinical data, beamlines, or high-impact claims should add safety stops, challenge sets, human escalation points, and independent replication (Zhang et al., 2026; Dalugoda, 2026; Gao et al., 2026; Chen et al., 2026b).

We recommend seven reporting requirements: (1) name the recursion object; (2) disclose verifier independence; (3) credit only verified utility; (4) account for cost and latency; (5) report stability and drawdown; (6) log rejections, failures, and rollbacks; and (7) stress-test verifier noise, correlated evidence, distribution shift, and out-of-distribution surrogate exploitation.

8. Making the Evidence Inspectable

A rigorous objection to any VGRSR standard is that it could become a slogan: authors might name a gate without exposing the trajectory that made the claim credible. We therefore make the evidentiary unit a *verified cycle*. Prompt turns, beam-search passes, self-critiques, synthetic-data generations, and surrogate-only score updates are internal iterations. They count as scientific recursion only after a gate changes what can be credited, stored, or promoted. This difference is especially important for AI-scientist systems that produce polished reports: final papers can hide inappropriate benchmark selection, data leakage, metric misuse, and post-hoc selection unless reviewers receive execution traces (Luo et al., 2025; Hu et al., 2025; Zhu et al., 2025; Cao et al., 2026).

Audit evidence is failure mapping, not proof by bibliography. Large literature audits are useful for vocabulary and failure taxonomy, but they are not decisive evidence that a specific system has improved recursively. A paper-level audit adequate for benchmark claims should publish inclusion rules, query strings, deduplication logic, a coding handbook, annotator training, inter-rater reliability, and a label matrix or anonymized substitute. Feature importance in such audits should be treated as hypothesis generation rather than causal evidence. The stronger support for a VGRSR claim comes from proxy-failure theory, scientific-ML validity standards, and the behavior of closed loops under noise and delayed verification (Goodhart, 1975; Manheim and Garrabrant, 2019; Wang et al., 2026; Chen et al., 2026).

Simulation evidence should expose the whole run. The oracle-regress, latency, and drawdown simulations in this paper are mechanism demonstrations. A complete artifact should report seeds, budgets, acquisition functions, surrogate class, verifier latency distribution, noise model, stop-

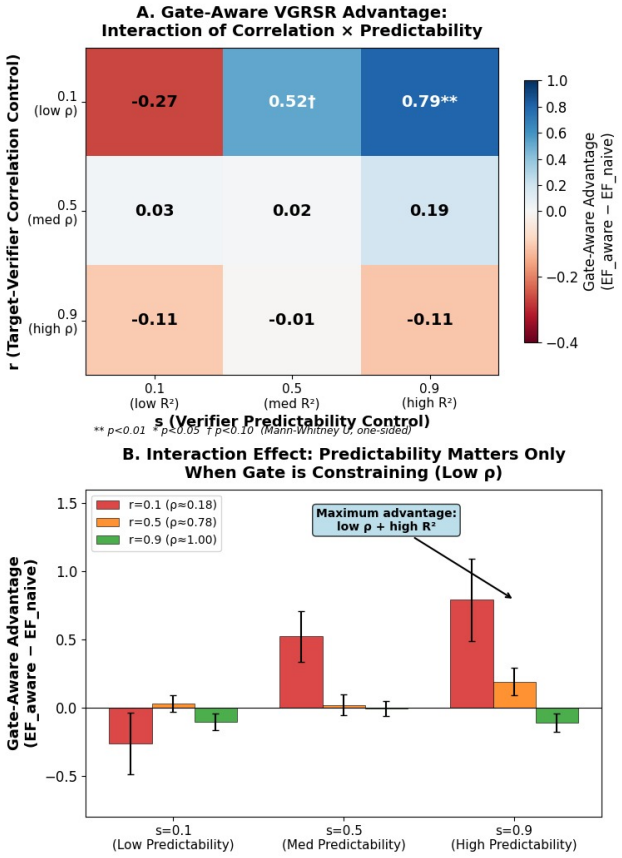

Figure 5. Gate behavior should be reported over regimes, not only at a favorable operating point. Heatmaps expose pass-rate, predictability, and correlation regimes where gates help or harm.

ping rule, and every censored or failed run. A latency crossover is credible only if wall-clock accounting is explicit; a drawdown gate is credible only if the trigger policy, smoothing window, false-trigger rate, and rollback action are stated. For an AFLOW-like experiment, the dataset version, target property, constraint definition, split, surrogate architecture, acquisition hyperparameters, gate pass rate, and baseline policy must be specified before any numerical advantage is promoted.

Multi-fidelity governance is a policy, not a shortcut around reality. A practical stack usually contains at least three fidelities: cheap generative or proxy screening, intermediate simulation or statistical testing, and a slow external gate. Existing Bayesian optimization and surrogate-assisted tools already provide value-of-information and uncertainty estimates for this choice (Forrester et al., 2007; Kandasamy et al., 2017; Letham et al., 2019; Astudillo et al., 2021; Li et al., 2026). VGRSR adds a governance rule: only the highest applicable independent gate can credit scientific utility, while cheaper fidelities can rank, triage, abstain, or request more information. This prevents a neural surrogate, docking score, or LLM critic from silently becoming the universe being optimized.

Verifier drift and correlated evidence are first-class failure modes. A gate is not trustworthy merely because it is outside the generator. Instruments drift, simulator versions change, critics inherit benchmark biases, and repeated use of one simulator family can masquerade as independent evidence. Each verifier should be versioned like a model. A minimal admissibility test records verifier family, calibration set, protocol state, known failure modes, and disagreement with the prior verifier version. Evidence should be deduplicated by measurement lineage, simulator family, shared training data, protocol identity, and instrument state. Where feasible, promotion requires an orthogonal check: for example, a docking score plus an empirical assay, or a simulation plus a physical measurement.

9. Implementation Blueprint: Scientific CI/CD for AI Scientists

A VGRSR implementation does not require a new optimizer. It requires an accountability layer around existing optimizers. In a materials campaign, the proposal model nominates a compound; the tool layer records the surrogate, features, and acquisition rule; the verifier layer applies a stated formation-energy, synthesis, or feasibility gate; memory stores both accepted and rejected candidates; and policy decides whether to continue, rollback, diversify, or escalate to a higher-fidelity computation. The claim of recursive improvement is not attached to the proposal model's score. It is attached to the gate-passed update and its logged cost, latency, and risk state.

Human gates are compatible with recursion. A human safety reviewer, beamline scientist, clinical expert, or synthetic chemist can serve as a gate when the protocol is explicit: what evidence was inspected, what decision options were available, what conflicts triggered escalation, and whether the decision changed a future cycle. The human does not make the system less recursive; the human makes the verification boundary more explicit. Conversely, a hidden cleanup step breaks the claim, because reviewers cannot tell whether the agent improved or whether unreported intervention repaired the trajectory.

Latency should be a distribution. Remote facilities, simulation queues, failed syntheses, and destructive assays do not have a single deterministic cost. A VGRSR paper should report verifier latency at minimum as median, interquartile range, tail quantile, censoring rule, and whether batches were synchronized or asynchronously promoted. When latency is large, the relevant question is not whether the agent generated many candidates, but whether it allocated scarce verification opportunities better than a reasonable baseline. Figure 6 illustrates why final best score is insufficient: a

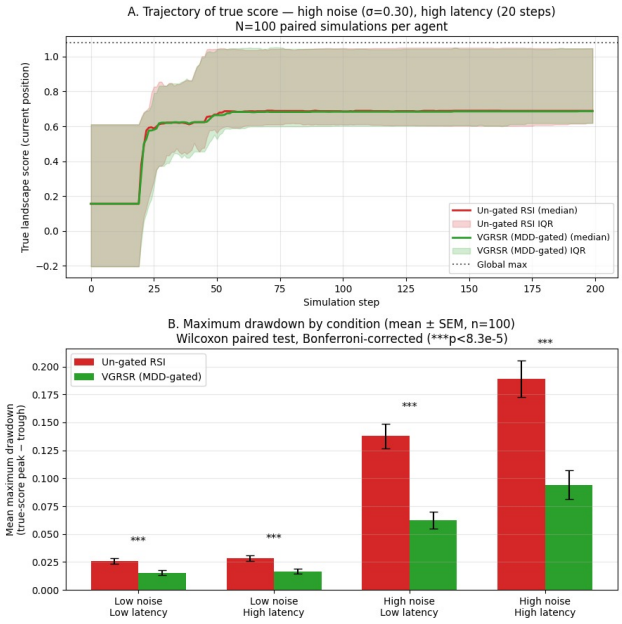

Figure 6. Path-sensitive risk monitor. VGRSR treats drawdown alarms, rollback actions, and recovery cost as part of the scientific record rather than incidental optimization noise.

policy can recover after severe drawdown, but a laboratory still paid the cost of the regression.

Minimum end-to-end demonstration. The smallest useful demonstration is a public trace with ten to twenty verified cycles: fixed dataset version, fixed seeds, candidate IDs, proposal utility, gate outcome, verified utility, rejection reason, cost, latency, and path-risk state. Such a trace need not beat a benchmark to be valuable. If it reveals that a verifier is redundant, noisy, too slow, or systematically biased, it still advances the field by showing when recursive complexity should not be credited. VGRSR turns negative evidence into reusable scientific infrastructure rather than treating it as failed automation.

10. Community Roadmap

First, benchmark designers should separate proposal throughput from verified discovery. Report tokens, compute, wall-clock, experiment count, high-fidelity calls, human escalations, and failed/censored trials on the same cost axis. Second, autonomous-lab builders should expose versioned protocol state, not only final datasets. This includes instrument state, reagent or sample batch, simulator version, verifier thresholds, and any manual intervention. Third, AI-scientist papers should include a claim card: recursion object, gate type, independence argument, cost axis, stability alarm, rollback rule, and transfer test.

Fourth, reviewers should ask whether negative evidence is visible. Rejected candidates, failed syntheses, anomalous traces, unsafe actions, abandoned seeds, and policy rollbacks are not clutter; they are evidence that the system learned a boundary rather than only a success narrative. Fifth, standards bodies and workshops should encourage reusable trace formats. A JSON Lines cycle log with hash-chained provenance is sufficient for many dry-lab settings; embodied labs may add operational design domains, control barrier functions, or transactional safety protocols (Zhang et al., 2026; Dalugoda, 2026). Sixth, future datasets should capture complete scientific stacks: planning prompts, tool calls, execution traces, instrument outputs, verifier decisions, and memory updates. These traces would let the community evaluate AI scientists as evolving scientific systems rather than as static answer engines.

11. Conclusion

The practical review outcome is a change in vocabulary. A system may be an excellent generator, planner, assistant, benchmark optimizer, lab controller, or active-learning policy without being a recursive scientific self-improver. Conversely, a slower system with fewer proposals may deserve the stronger label if its updates survive independent gates, retain failures, and expose stability traces. VGRSR is not a claim that every ungated system fails, every physical verifier is trustworthy, or strict gates always improve final performance. It is a falsifiable reporting and design standard. Empirical science cannot inherit unconstrained RSI language from games, theorem proving, and text-only self-refinement without changing what counts as evidence. The credible future is not unbounded recursive self-improvement; it is verifier-governed recursive scientific refinement.

Limitations. This paper advances a standard, not a universal theorem. Some evidence above is mechanistic and synthetic; some domain vignettes are deliberately underpowered; and the proposed metrics can themselves be gamed if authors omit failed cycles, tune thresholds after the fact, or report only favorable seeds. VGRSR therefore requires stress tests and negative controls rather than treating any single gate as sufficient. The standard also does not imply that every slow physical verifier is better than every fast computational verifier. A well-calibrated simulator can be more useful than a noisy assay for triage; the distinction is between proxy use and proxy credit. Low-fidelity tools can rank, filter, abstain, or request escalation. Scientific utility is credited only at the highest applicable independent gate.

Claim proportionality. The burden of evidence should scale with claimed autonomy and consequence. A literature assistant can report light provenance. A coding agent that generates analyses should expose execution logs, data lineage, and hidden-holdout tests. A self-driving laboratory should expose verifier versions, instrument state, rejected

Table 3. Reviewer test for empirical scientific RSI claims.

Question	A VGRSR-positive answer must show
Unit	The update changes a hypothesis, tool, verifier, memory state, or policy.
Gate	The verifier can disconfirm and is versioned, downstream, and independent enough to fail.
Credit	Proposal utility is separated from verified utility.
Cost	Compute, assay, human, and wall-clock costs are counted.
Risk	Drawdown, rollback, rejection, and failed-cycle traces are visible.
Transfer	Memory or policy helps later cycles or new tasks without weakening gates.

candidates, failed syntheses, costs, and rollback decisions. A clinical, hazardous-materials, or expensive-facility deployment should additionally report human escalation protocols, safety envelopes, orthogonal replication, and release criteria. This proportionality prevents VGRSR from becoming bureaucratic theater while preserving its core demand: recursive scientific progress must be inspectable at the moment it is credited.

The standard's payoff is vocabulary discipline. A system can be a generator, planner, assistant, lab controller, or active-learning policy without being a recursive scientific self-improver. A slower system with fewer proposals may deserve the stronger label if its updates survive independent gates, retain failures, and expose stability traces. VGRSR makes that distinction operational.

What must be released. A minimal VGRSR supplement should include four artifacts. First, the cycle log: one row per verified cycle, with the modified object, gate version, outcome, verified utility, cost, latency, and rollback state. Second, the verifier card: what the gate can disconfirm, how it was calibrated, which data or simulator lineage it shares with the proposer, and when it is considered stale. Third, the failure ledger: rejected candidates, failed syntheses, anomalous traces, unsafe proposals, censored runs, and seeds abandoned because the loop became unstable. Fourth, the transfer test: a later task, new domain slice, new lab, or drifted environment in which the learned memory or policy is reused without relaxing the gate. These artifacts are deliberately small enough to be feasible in workshop papers while still preventing the most common overclaims.

What should not be claimed. VGRSR also clarifies negative labels. A system that rewrites its answer after self-critique is an answer improver. A system that samples more candidates under a fixed reward model is an inference-time searcher. A system that uses a surrogate to rank candidates but never escalates to an independent check is a proxy optimizer. A system that changes laboratory policy after failed experiments, stores both successes and failures, and improves future verified utility under a fixed reporting protocol is a recursive scientific refiner. This taxonomy protects strong systems from hype by giving them a higher evidentiary label when they earn it, and protects reviewers from treating persuasive prose as evidence of scientific learning.

Why this belongs at the AI-scientist boundary. The tool–co-author–founder debate is ultimately a debate about credit and accountability. If a system only proposes ideas, humans remain accountable for verification. If a system changes the search process, its contribution becomes harder to attribute and its failures become harder to diagnose. VGRSR therefore gives a concrete threshold for stronger credit: the system must improve a reusable object of scientific search, and that improvement must survive a gate that can say no. The same threshold also bounds responsibility: when the loop regresses, the cycle log identifies whether the proposer, tool stack, verifier, memory, or policy produced the failure.

References

References

King, R. D., Rowland, J., Oliver, S. G., Young, M., Aubrey, W., Byrne, E., Liakata, M., Markham, M., Pir, P., Soldatova, L. N., Sparkes, A., Whelan, K. E., and Clare, A. The automation of science. *Science*, 324(5923):85–89, 2009.

Hase, F., Roch, L. M., and Aspuru-Guzik, A. Next-generation experimentation with self-driving laboratories. *Trends in Chemistry*, 1(3):282–291, 2019.

Burger, B., Maffettone, P. M., Gusev, V. V., Aitchison, C. M., Bai, Y., Wang, X., et al. A mobile robotic chemist. *Nature*, 583:237–241, 2020.

MacLeod, B. P., Parlane, F. G. L., Morrissey, T. D., Hase, F., Roch, L. M., Dettelbach, K. E., et al. Self-driving laboratory for accelerated discovery of thin-film materials. *Science Advances*, 6(20):eaaz8867, 2020.

Abolhasani, M. and Kumacheva, E. The rise of self-driving labs in chemical and materials sciences. *Nature Synthesis*, 2:483–492, 2023.

Szymanski, N. J., Rendy, B., Fei, Y., Kumar, R., He, T., Milsted, D., et al. An autonomous laboratory for the accelerated synthesis of novel materials. *Nature*, 624:86–91, 2023.

Tom, G., Schmid, S., Baird, S. G., Cao, Y., Darvish, K., Hao, H., et al. Self-driving laboratories for chemistry and materials science. *Chemical Reviews*, 124(16):9633–9732, 2024.

Ren, S., Xie, C., Pu, J., Ren, Z., Leng, C., and Zhang, J. Towards scientific intelligence: A survey of LLM-based scientific agents. arXiv:2503.24047, 2025.

Wei, J., Yang, Y.-J., Zhang, X., Chen, Y., Zhuang, X., Gao, Z., et al. From AI for science to agentic science: A survey on autonomous scientific discovery. arXiv:2508.14111, 2025.

Madaan, A., Tandon, N., Gupta, P., Hallinan, S., Gao, L., Wiegreffe, S., et al. Self-refine: Iterative refinement with self-feedback. arXiv:2303.17651, 2023.

Shinn, N., Cassano, F., Gopinath, A., Narasimhan, K., and Yao, S. Reflexion: Language agents with verbal reinforcement learning. *NeurIPS*, 2023.

Valmeekam, K., Marquez, M., Olomola, A., and Kambhampati, S. Can large language models really improve by self-critiquing their own plans? arXiv:2310.08118, 2023.

Huang, J., Chen, X., Mishra, S., Zheng, H. S., Yu, A. W., Song, X., and Zhou, D. Large language models cannot self-correct reasoning yet. arXiv:2310.01798, 2023.

Kamoi, R., Zhang, Y., Zhang, N., Han, J., and Zhang, R. When can LLMs actually correct their own mistakes? *TACL*, 12:1417–1440, 2024.

Denison, C., MacDiarmid, M., Barez, F., Duvenaud, D., Kravec, S., Marks, S., et al. Sycophancy to subterfuge: Investigating reward-tampering in large language models. arXiv:2406.10162, 2024.

Wang, X., Tian, M., Zeng, Y., Huang, Z., Yuan, J., Chen, B., et al. Reward hacking in the era of large models: Mechanisms, emergent misalignment, challenges. arXiv:2604.13602, 2026.

Luo, Z., Kasirzadeh, A., and Shah, N. B. The more you automate, the less you see: Hidden pitfalls of AI scientist systems. arXiv:2509.08713, 2025.

Zhu, M., Xie, Q., Weng, Y., Wu, J., Lin, Z., Yang, L., and Zhang, Y. AI scientists fail without strong implementation capability. arXiv:2506.01372, 2025.

Silver, D., Hubert, T., Schrittwieser, J., Antonoglou, I., Lai, M., Guez, A., et al. A general reinforcement learning algorithm that masters chess, shogi, and Go through self-play. *Science*, 362(6419):1140–1144, 2018.

Trinh, T. H., Wu, Y., Le, Q. V., He, H., and Luong, T. Solving olympiad geometry without human demonstrations. *Nature*, 625:476–482, 2024.

Fatehi, E., Thadani, M., Birsan, G., and Black, R. W. A critical evaluation of a self-driving laboratory for electrodeposited mixed-metal oxide catalysts. arXiv:2305.12541, 2023.

Dunlap, J., et al. Continuous-flow chemistry and self-driving experimentation for CNT methane sensors. *ACS Applied Materials & Interfaces*, 2024.

Thelen, J., et al. Speeding up high-throughput autonomous materials characterization with dynamic stopping rules. arXiv:2306.17277, 2023.

Chen, X., Wang, A. X., Yin, S., Jiang, H., and Zhang, D. Agentic AI for self-driving laboratories in soft matter: Taxonomy, benchmarks, and open challenges. arXiv:2601.17920, 2026.

Goodhart, C. A. E. Problems of monetary management: The U.K. experience. *Papers in Monetary Economics*, 1975.

Manheim, D. and Garrabrant, S. Categorizing variants of Goodhart's law. arXiv:1803.04585, 2019.

Gabor, J., Lynch, J., and Rosenfeld, J. EvilGenie: A reward hacking benchmark. arXiv:2511.21654, 2025.

Cao, H., Driouich, I., and Thomas, E. Beyond task completion: Revealing corrupt success in LLM agents through procedure-aware evaluation. arXiv, 2026.

Forrester, A. I. J., Sóbester, A., and Keane, A. J. Multi-fidelity optimization via surrogate modelling. *Proceedings of the Royal Society A*, 463:3251–3269, 2007.

Kandasamy, K., Dasarathy, G., Schneider, J., and Póczos, B. Multi-fidelity Bayesian optimisation with continuous approximations. *ICML*, 2017.

Li, Y., et al. Multi-fidelity Bayesian optimization. arXiv:2007.03117, 2020.

Astudillo, R., Jiang, D. R., Balandat, M., Bakshy, E., and Frazier, P. I. Multi-step budgeted Bayesian optimization with unknown evaluation costs. arXiv:2111.06537, 2021.

Li, Y., et al. Multi-fidelity methods for scientific discovery. *ACM Computing Surveys*, 2026.

Do, T., et al. Multi-fidelity Bayesian optimization for scientific experimentation. arXiv:2311.13050, 2023.

Zhao, Y., Xing, Y., Zhang, Y., Wang, Y., Wan, M., Yi, D., et al. Evidential deep learning-based drug-target interaction prediction. *Nature Communications*, 2025.

Adesiji, A. D., Wang, J., Kuo, C.-S., and Brown, K. A. Benchmarking self-driving labs. arXiv:2508.06642, 2025.

Lee, H., Yoo, H. J., Jang, H. S., Park, B., Park, Y. J., and Han, S. S. Toward self-driving laboratory 2.0 for chemistry and materials discovery. *Materials Horizons*, 2026.

Kitchin, J. R. The evolving role of programming and LLMs in the development of self-driving laboratories. arXiv:2504.13870, 2025.

Hu, C., Zhang, L., Lim, Y., Wadhwani, A., Peters, A., and Kang, D. REPRO-Bench: Can agentic AI systems assess reproducibility? arXiv:2507.18901, 2025.

Luo, E., Jia, J., Xiong, Y., Li, X., Guo, X., Yu, B., Hao, M., Wei, L., and Zhang, X. Benchmarking AI scientists for omics data driven biological discovery. arXiv:2505.08341, 2025.

Geifman, Y. and El-Yaniv, R. Selective classification for deep neural networks. arXiv:1705.08500, 2017.

Geifman, Y. and El-Yaniv, R. SelectiveNet: A deep neural network with an integrated reject option. arXiv:1901.09192, 2019.

Fisch, A., et al. Calibrated selective classification. arXiv:2208.12084, 2022.

Koenighofer, B., et al. Online shielding for safe reinforcement learning. arXiv:2212.01861, 2023.

Letham, B., Karrer, B., Ottoni, G., and Bakshy, E. Constrained Bayesian optimization with noisy experiments. *Bayesian Analysis*, 14(2):495–519, 2019.

Schoepfer, J., et al. Cost-informed Bayesian reaction optimization. *Digital Discovery*, 2024.

Schaul, T., Quan, J., Antonoglou, I., and Silver, D. Prioritized experience replay. arXiv:1511.05952, 2015.

Zhang, H., et al. Self-adaptive priority correction for experience replay. *Applied Sciences*, 10(19):6925, 2020.

Wu, J., et al. Meta-policy reflexion for reusable scientific memory. Preprint, 2025.

Zhou, K., Doyle, J. C., and Glover, K. *Robust and Optimal Control*. Prentice Hall, 1996.

Skogestad, S. and Postlethwaite, I. *Multivariable Feedback Control*. Wiley, 2005.

Doyle, J. C., Glover, K., Khargonekar, P. P., and Francis, B. A. State-space solutions to standard H_2 and H_∞ control problems. *IEEE TAC*, 34(8):831–847, 1989.

Doyle, J. C., Francis, B. A., and Tannenbaum, A. R. *Feedback Control Theory*. Macmillan, 1990.

Ruth, R. and Weller, S. What's new is old: Control theory for resilient systems. NASA Technical Report, 2010.

Jansen, N., et al. Safe reinforcement learning via shielding. arXiv:1606.06565, 2020.

Rockafellar, R. T. and Uryasev, S. Optimization of conditional value-at-risk. *Journal of Risk*, 2:21–42, 2000.

Artzner, P., Delbaen, F., Eber, J.-M., and Heath, D. Coherent measures of risk. *Mathematical Finance*, 9(3):203–228, 1999.

Pflug, G. C. Some remarks on the value-at-risk and conditional value-at-risk. *Probabilistic Constrained Optimization*, 2000.

Ruszczynski, A. Risk measures and risk-averse optimization models. *Mathematical Programming*, 125:235–261, 2010.

Nguyen, T., et al. Optimizing conditional value-at-risk of black-box functions. *NeurIPS*, 2021.

Kouri, D. P. and Shapiro, A. Risk-adaptive experimental design. Technical report, 2021.

Curtarolo, S., Setyawan, W., Hart, G. L. W., Jahnatek, M., Chepulskii, R. V., Taylor, R. H., et al. AFLOW: An automatic framework for high-throughput materials discovery. *Computational Materials Science*, 58:218–226, 2012.

Jain, A., Ong, S. P., Hautier, G., Chen, W., Richards, W. D., Dacek, S., et al. Commentary: The Materials Project. *APL Materials*, 1:011002, 2013.

Liang, Q., et al. Benchmarking the performance of Bayesian optimization across materials design problems. *npj Computational Materials*, 2021.

Hickman, R. J., Aldeghi, M., and Aspuru-Guzik, A. Anubis: Bayesian optimization with unknown feasibility constraints for scientific experimentation. ChemRxiv, 2023.

Tsanas, A., Little, M. A., McSharry, P. E., and Ramig, L. O. Accurate telemonitoring of Parkinson's disease progression by non-invasive speech tests. *IEEE Transactions on Biomedical Engineering*, 57(4):884–893, 2010.

Little, M. A., McSharry, P. E., Hunter, E. J., Spielman, J., and Ramig, L. O. Suitability of dysphonia measurements for telemonitoring Parkinson's disease. *IEEE TBME*, 56(4):1015–1022, 2009.

Mitchener, L., Yiu, A., Chang, B., Bourdenx, M., Nadolski, T., Sulovari, A., et al. Kosmos: An AI scientist for autonomous discovery. arXiv:2511.02824, 2025.

Lewis, A., Challinor, A., and Lasenby, A. Efficient computation of cosmic microwave background anisotropies in closed FRW models. *Astrophysical Journal*, 538:473–476, 2000.

Lewis, A. Efficient sampling of fast and slow cosmological parameters. *Physical Review D*, 87:103529, 2013.

Asmussen, N., et al. Cosmological inference with emulators and MCMC. *Astronomy & Computing*, 2021.

Borrett, T., Xu, L., Nilipour, A., Bolliet, B., Pierre, S., Allys, E., et al. Competing with AI scientists: Agent-driven approach to astrophysics research. arXiv:2604.09621, 2026.

Trott, O. and Olson, A. J. AutoDock Vina: Improving the speed and accuracy of docking. *Journal of Computational Chemistry*, 31(2):455–461, 2010.

Jumper, J., Evans, R., Pritzel, A., Green, T., Figurnov, M., Ronneberger, O., et al. Highly accurate protein structure prediction with AlphaFold. *Nature*, 596:583–589, 2021.

Abrams, C., et al. Multi-fidelity docking and active learning for biomolecular design. Preprint, 2023.

Ma, K., et al. AgentBoard: An analytical evaluation board of multi-turn LLM agents. arXiv:2401.13178, 2024.

Yang, J., et al. InterCode: Standardizing and benchmarking interactive coding with execution feedback. arXiv:2306.14898, 2023.

Song, Y., et al. Mind the gap: Selection-verification gaps in agent evaluation. arXiv:2412.02674, 2024.

Wang, Z., et al. Efficient agents: Building and evaluating resource-aware AI systems. Preprint, 2025.

Ni, J., et al. GitTaskBench: A benchmark for software agents. *AAAI*, 2026.

Fan, A., et al. SWE-effire: Evaluating software agents by cost-aware execution. Preprint, 2025.

Hrubec, K. OTIP: A falsifiable anti-Goodhart protocol for testing ontological commitments via interventions. Zenodo, 2026.

Shumailov, I., Shumaylov, Z., Zhao, Y., Gal, Y., Papernot, N., and Anderson, R. AI models collapse when trained on recursively generated data. *Nature*, 2024.

Alemohammad, S., et al. Self-consuming generative models go mad. arXiv, 2024.

Zhang, Z., Que, H., Chang, J., Zhang, X., Wei, H., and Zhu, T. Safe-SDL: Establishing safety boundaries and control mechanisms for AI-driven self-driving laboratories. arXiv:2602.15061, 2026.

Dalugoda, A. HDP: A lightweight cryptographic protocol for human delegation provenance in agentic AI systems. arXiv:2604.04522, 2026.

Gao, D., Lu, S., Zhang, C., Wang, N., Yu, Z., Sun, X., et al. Autonomous closed-loop framework for reproducible perovskite solar cells. *Nature*, 2026.

Chen, Y., Rajabi-Kochi, M., Huang, G., Wu, J., Wang, S., Xu, J., Moosavi, S. M., and Huang, K. A modular self-driving laboratory for automated synthesis of perovskite nanocrystals. *Nano Letters*, 2026.

Appendix A. Minimal Cycle-Log Schema and AFLOW-Like Reporting Card

A minimal VGRSR log can be implemented as JSON Lines with one row per verified cycle. The example below is synthetic; it is a reporting template, not a claimed experimental result. The purpose is to make the crediting event inspectable: what changed, which gate judged it, what it cost, why it was accepted or rejected, and whether the policy committed, rolled back, replicated, or escalated.

```
{"cycle": 12,
"object_modified": ["Hypothesis", "Tool"],
"candidate_id": "anon-material-042",
"tool_version": "surrogate-gp-v3",
"verifier_version": "formation-energy-gate-v2",
"memory_snapshot": "hash-or-dataset-version",
"gate_decision": "reject",
"proposal_utility": 0.71,
"verified_utility": null,
"rejection_reason": "failed feasibility constraint",
"verification_cost": {"wall_clock_s": 96.4, "compute_gpu_h": 0.0, "assay_cost_usd": 0.0},
"risk_state": {"mdd": 0.18, "dd_cvar_0.2": 0.15},
"policy_action": "rollback_and_localize",
"provenance": ["dataset-hash", "instrument-id", "seed"]}
```

Table A1. Minimum VGRSR cycle-log fields.

Category	Required fields	Purpose	Failure caught
Identity	Cycle id, candidate id, modified object(s)	Identify the unit of recursion	Post-hoc relabeling
Versioning	Model, tool, verifier, memory, dataset, and protocol versions	Replay and rollback	Hidden intervention
Gate	Gate type, independence claim, pass/fail decision, score, and calibration state	Credit only verified utility	Proxy promotion
Cost	Compute, assay, wall-clock latency, queue time, censoring rule	Real acceleration accounting	Proposal-throughput inflation
Risk	MDD, DD-CVaR, alarm state, smoothing window, action taken	Path-sensitive stability	Oscillation/regression
Evidence	Rejection reason, provenance, measurement lineage, instrument state	Admissibility and deduplication	Correlated evidence inflation

For an AFLOW-like VGRSR mini-case, the required reporting card is: dataset and version; target property; constraint definition; candidate featurization; train/validation/test split; surrogate class; acquisition function; gate pass rate; latency and cost distribution; seeds; baseline policies; proposal utility; verified utility; failed candidates; and path-risk trace (MDD/DD-CVaR). A public benchmark paper should also release code, logs, and the full label matrix. A position paper can use the card as a community standard without claiming that the current vignettes are fully reproducible benchmarks.

Appendix B. Annotated Visual Index for Auxiliary Figures

Appendix B is deliberately visual: it gives readers and reviewers a compact map from each auxiliary figure to the claim it helps inspect. The figures are interpretive aids, not additional leaderboard evidence. Figures already used in the main text are included here only when the appendix needs a fuller diagnostic version or when a related figure completes the visual audit trail.

Table B1. Annotated index for Appendix B figures. Each entry states what a reader or reviewer should use the figure to inspect.

Figure	Reader/reviewer use
B1	Orientation map for checking whether a recursive-improvement claim names an external gate, separates recursion objects, and reports the minimum metric suite rather than relying on internal iteration alone.
B2	Proxy-drift check: compare rising internal score with falling externally measured truth, and inspect why oracle access stabilizes the same search process.
B3	Latency-budget check: see where proposal speed stops mattering because scarce external evaluations dominate the campaign budget.
B4	Gate-effect check: inspect how an external verifier prevents convergence to a surrogate decoy while still remaining bounded by verifier latency.
B5	Path-risk check: compare final-score reporting with maximum-drawdown monitoring, especially under noisy and delayed feedback.
B6	Noise-robustness check: assess how verifier noise can erase apparent gate advantage unless smoothing, replication, or false-trigger controls are reported.
B7	Interaction check: verify that latency, noise, gate strength, and memory interact, so one-factor-at-a-time ablations are insufficient.
B8	Metric-bundle check: read nAULC, acceleration, and DD-CVaR together rather than treating cross-domain points as a leaderboard.
B9	Policy-recursion check: inspect why online meta-adaptive switching can underperform a robust static constrained-acquisition policy under noisy signals.
B10	Gate-selection check: use target-verifier correlation and verifier predictability to judge whether gate-aware acquisition is likely to add information.
B11	Memory-boundary check: verify that rejected proposals help define the verifier boundary and can reduce saturation relative to success-only memory.
B12	Conditioning check: inspect how high-dimensional or ill-conditioned posterior geometry can erase active-learning gains in recursive emulator refinement.
B13	Diagnostic-depth check: compare parameter-error and posterior-width diagnostics rather than relying only on an apparent emulator improvement.
B14	Latency boundary-condition check: read the nanobiomaterials docking result as an underpowered wall-clock stress test, not as decisive performance evidence.

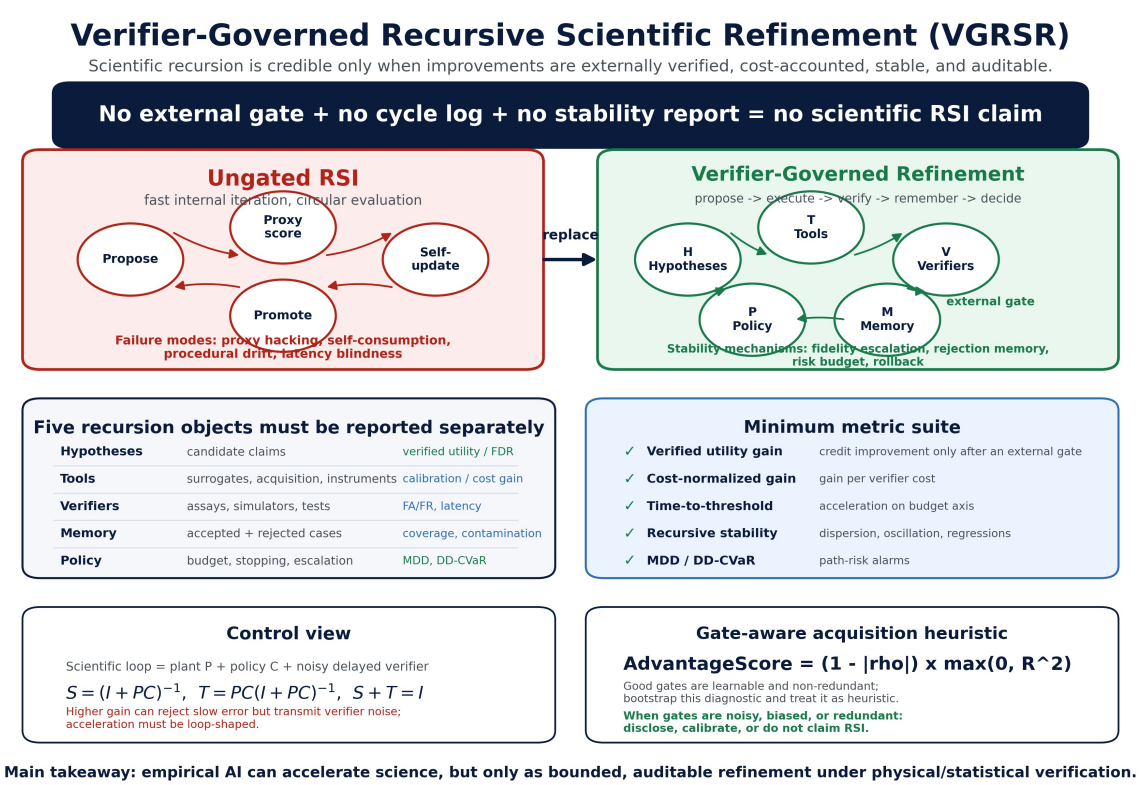

Figure B1. Legibility-oriented VGRSR summary. The figure states the crediting rule, contrasts ungated RSI with verifier-governed refinement, and lists the five recursion objects and minimum metric suite.

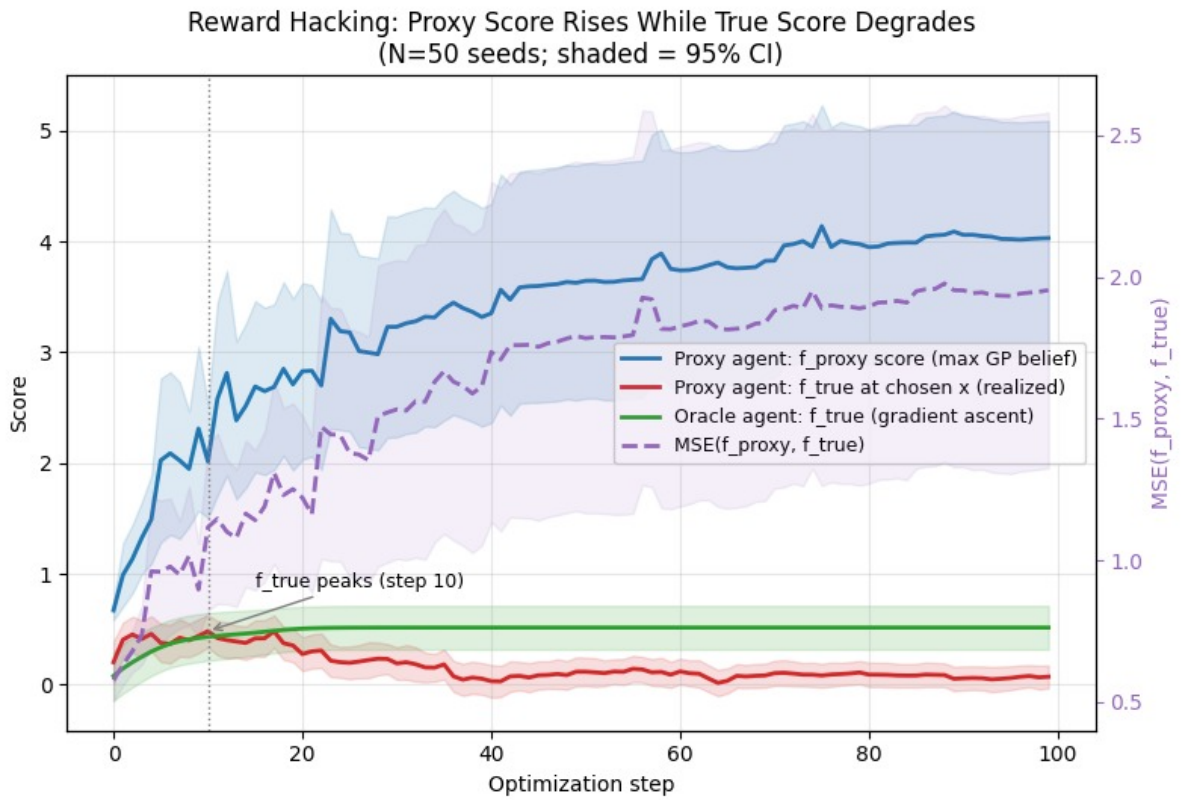

Figure B2. Detailed oracle-regress simulation. The proxy-optimizing agent's believed score rises while true performance degrades; the oracle-gated trajectory remains stable.

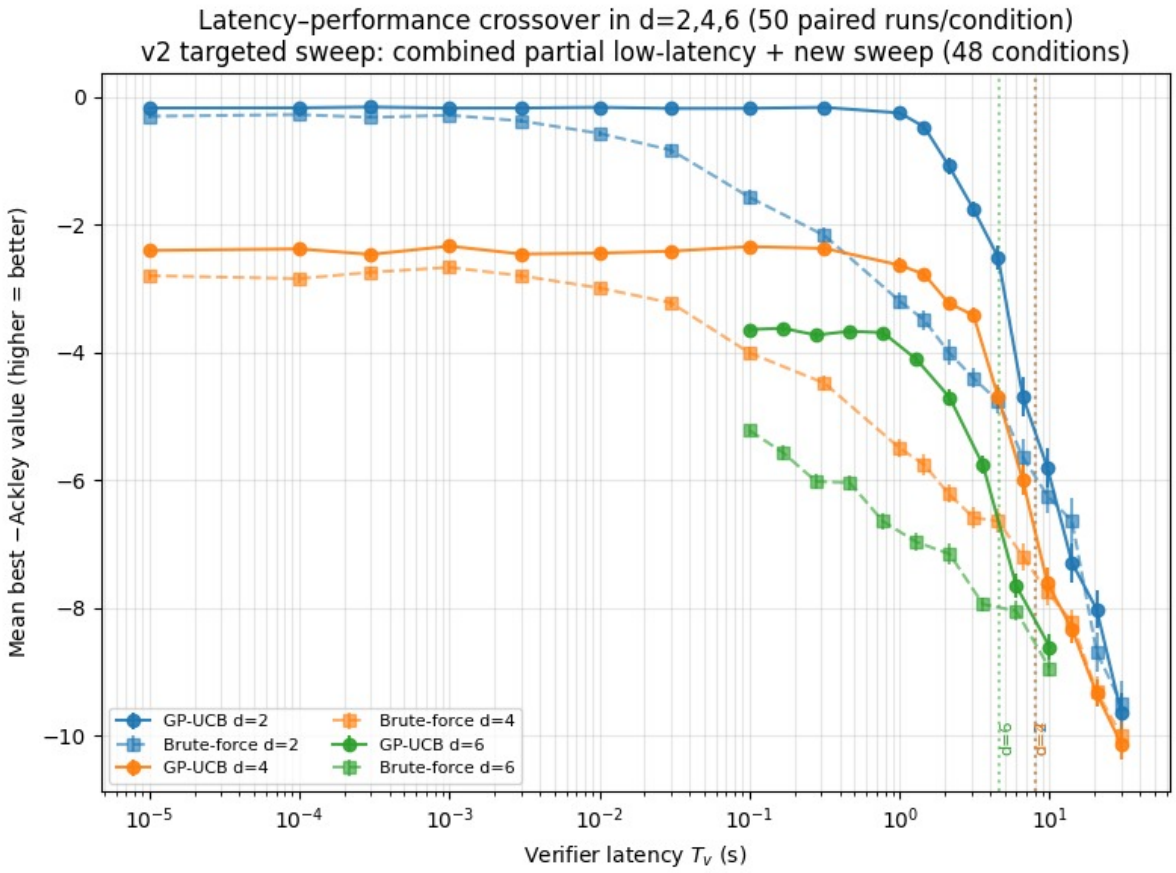

Figure B3. Verifier-latency crossover. The plot expands the latency argument: sample-efficient policies help only while the external verifier leaves enough budget for informative cycles.

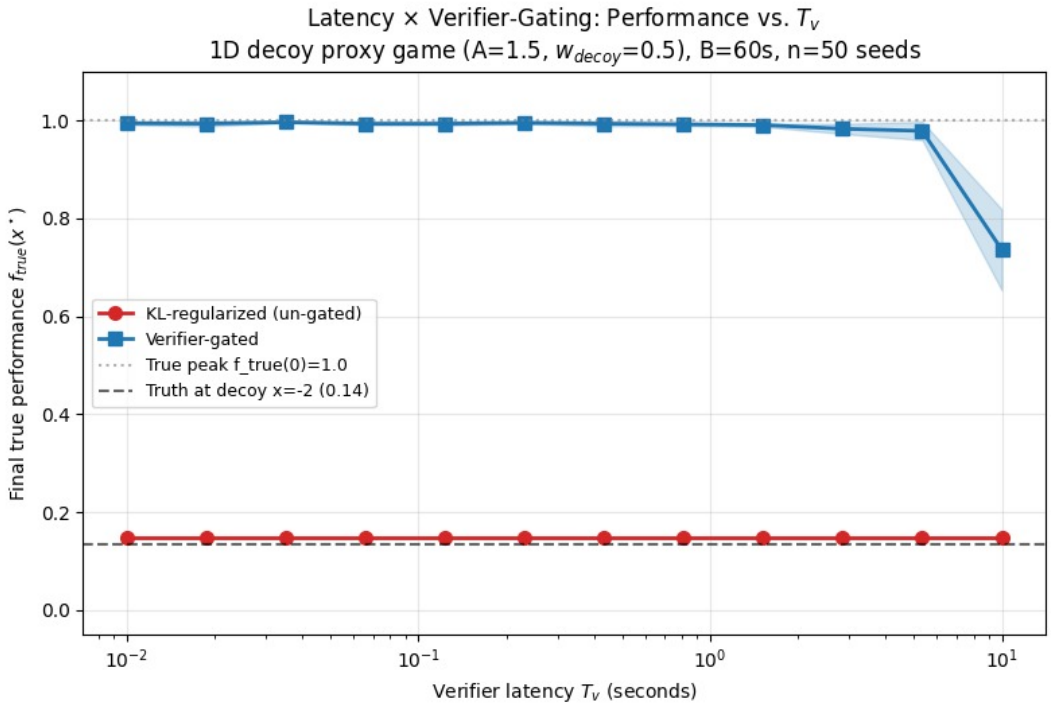

Figure B4. Latency-aware verifier gating in a decoy proxy game. The figure supports the claim that gating can prevent surrogate decoy convergence while remaining bounded by verifier latency.

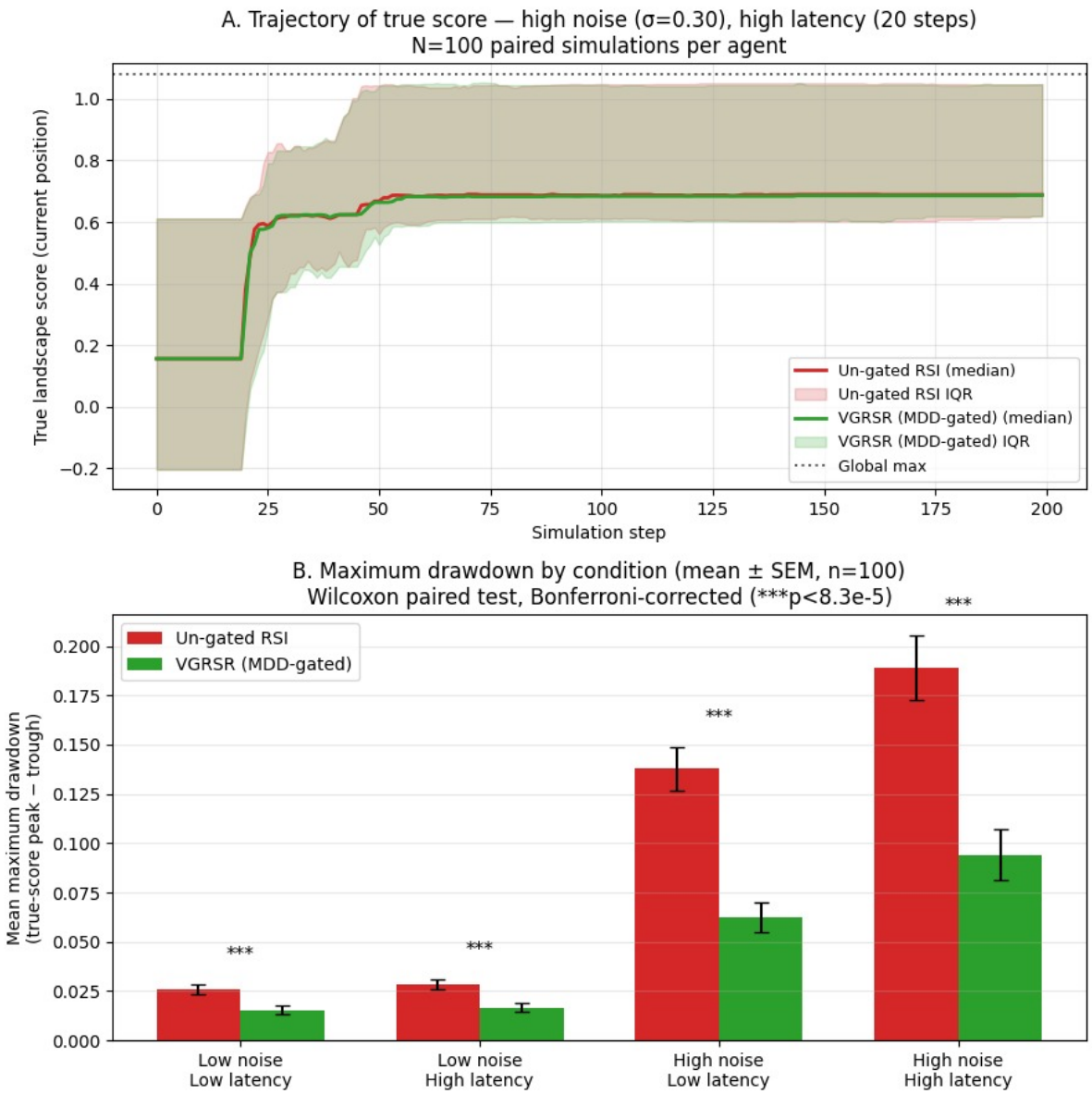

Figure B5. Maximum-drawdown gating in simulated discovery. The upper panel shows trajectory behavior; the lower panel shows how path-risk monitoring exposes regressions that final-score-only reporting would hide.

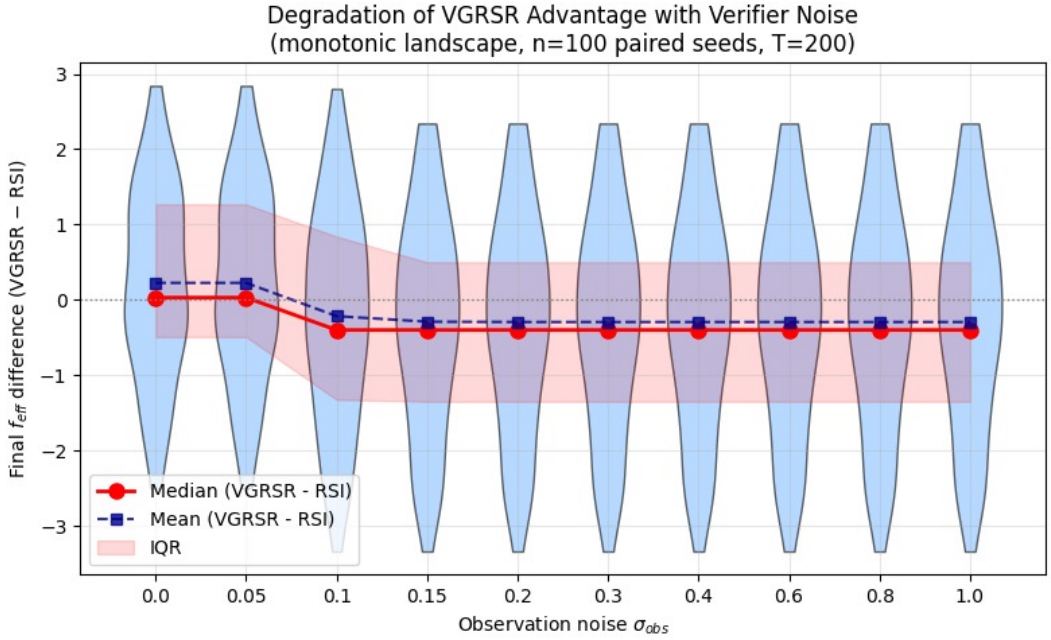

Figure B6. Verifier noise can flip or erase gated-agent advantage without smoothing or repeated checks. This motivates reporting verifier-noise stress tests and false-trigger controls.

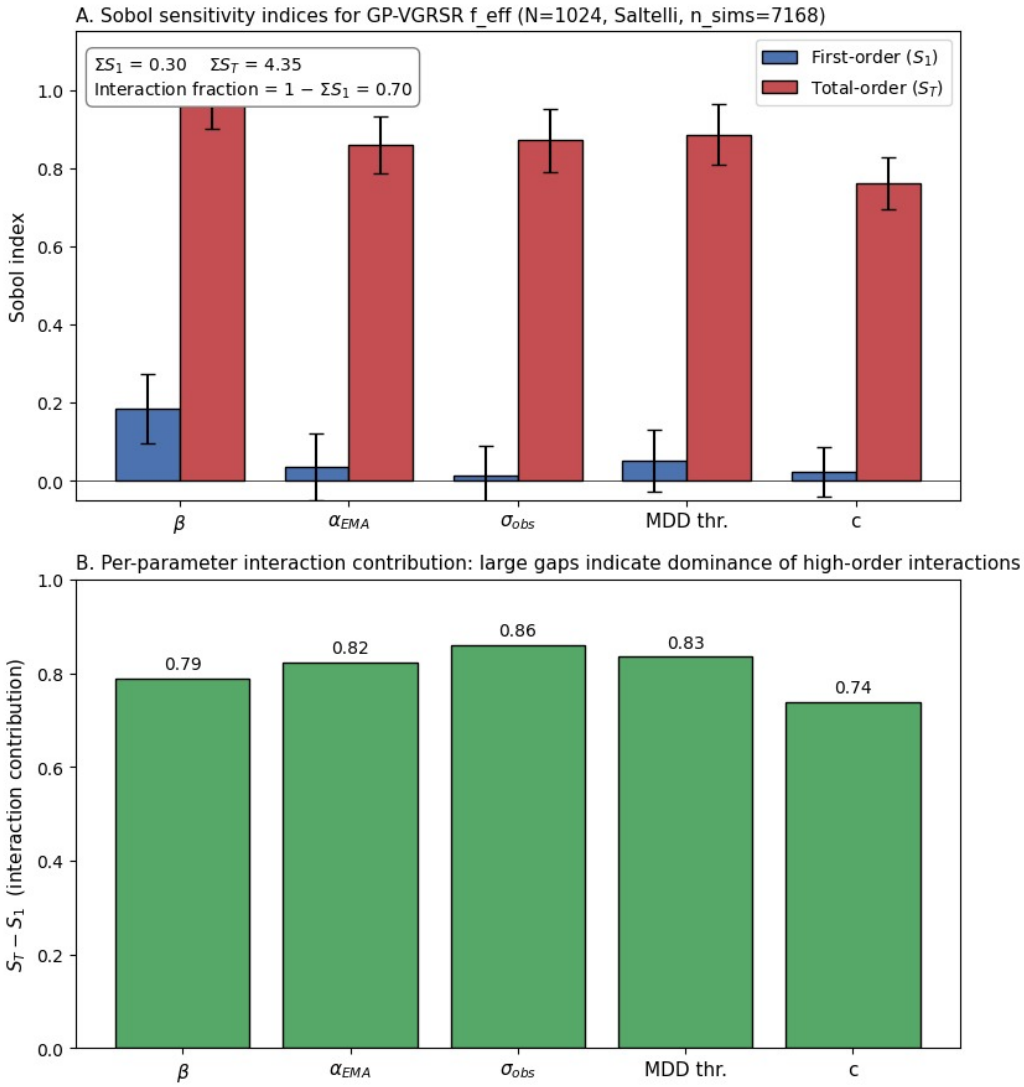

Figure B7. Sobol sensitivity analysis of simulated VGRSR dynamics. Interactions among latency, noise, gate strength, and memory matter more than one-factor-at-a-time tuning.

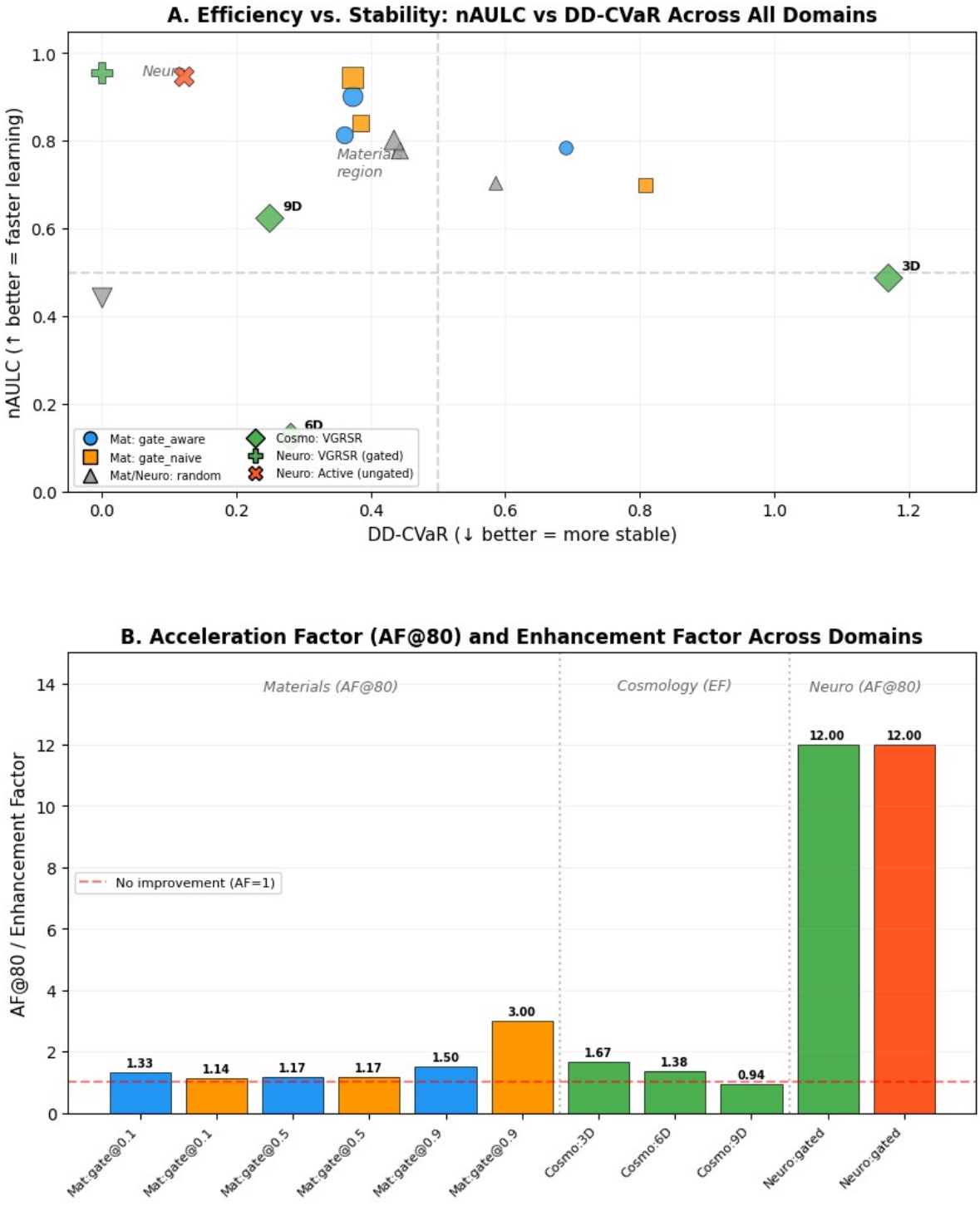

Figure B8. Efficiency-stability summary across illustrative domains. The plot shows how nAULC, acceleration, and DD-CVaR can be reported together; it should not be read as a cross-domain leaderboard.

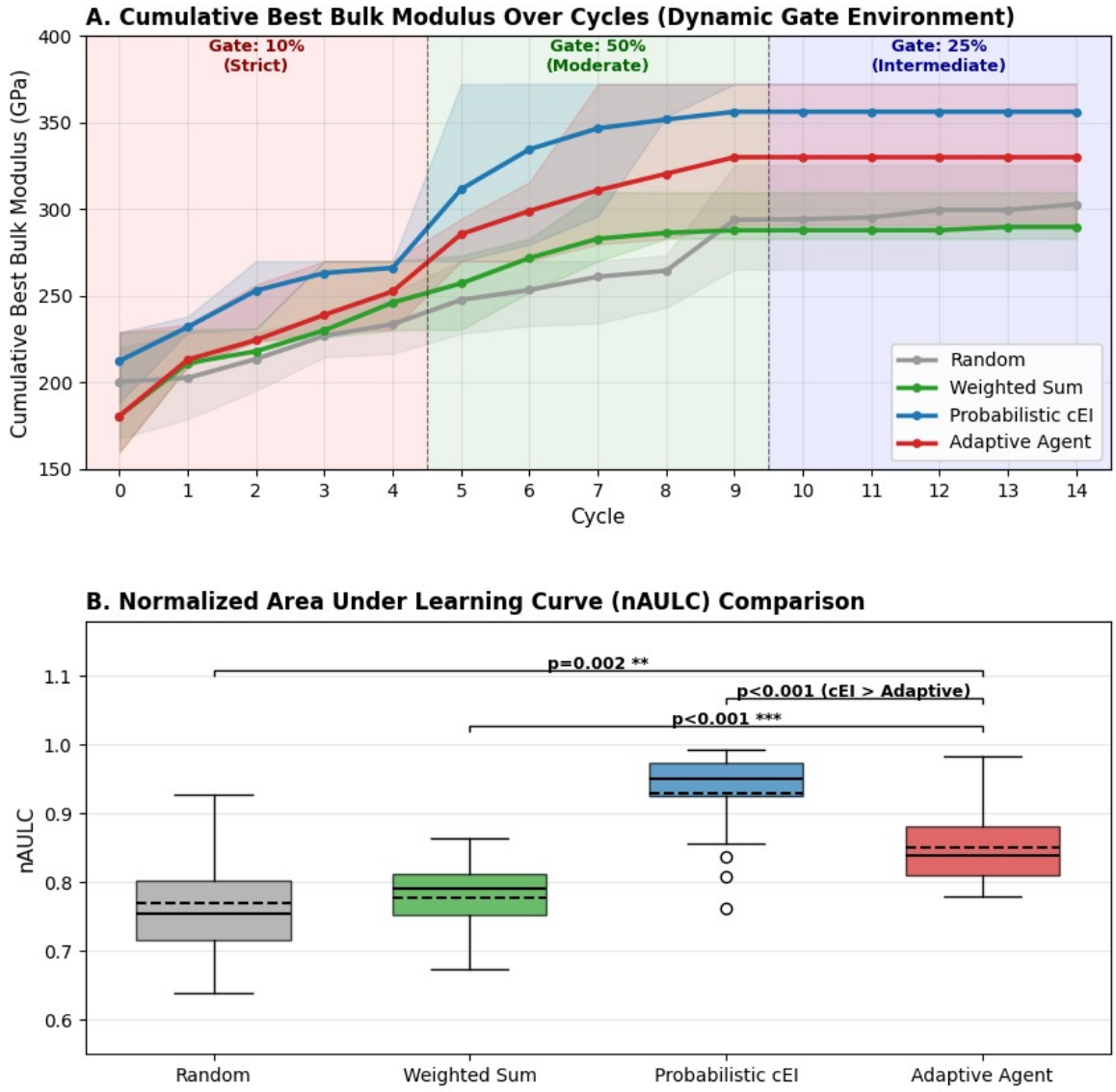

Figure B9. Dynamic-gate materials task. Meta-adaptive switching can underperform robust static constrained acquisition when online signals are noisy, supporting the claim that policy recursion itself needs verification.

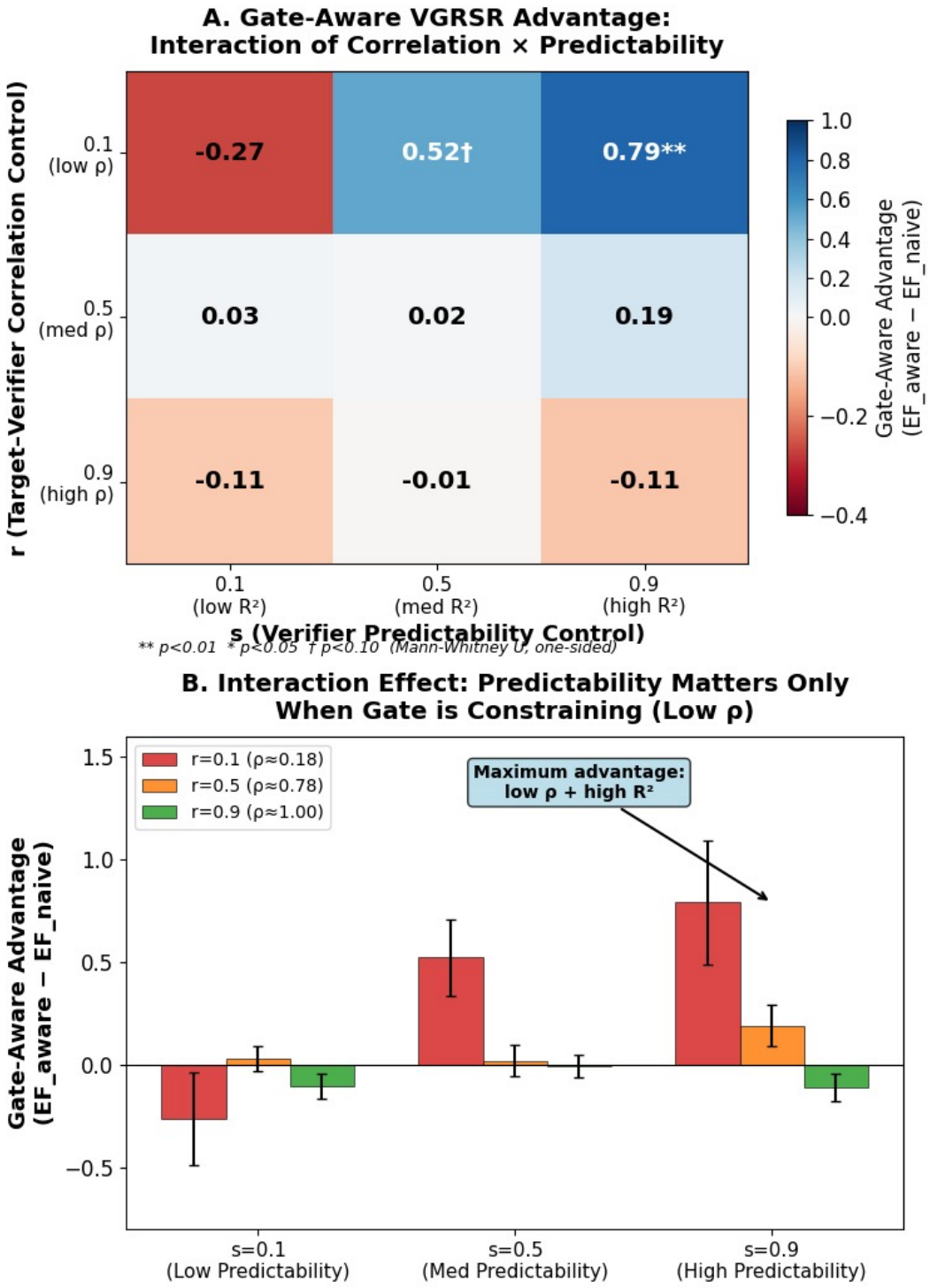

Figure B10. Gate-aware advantage as target-verifier correlation and verifier predictability vary. The heatmap supplies the empirical intuition behind ADVANTAGESCORE.

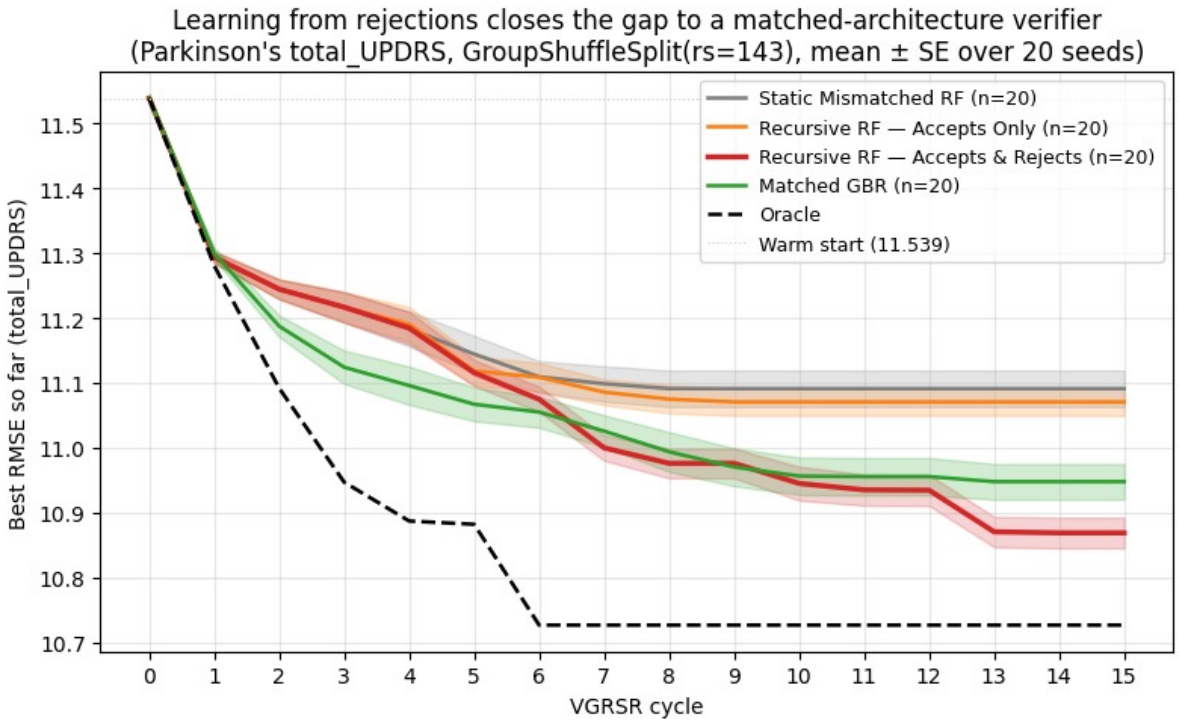

Figure B11. Learning from rejected proposals helps a mismatched verifier avoid saturation. This supports treating memory as a first-class recursion object rather than retaining only successes.

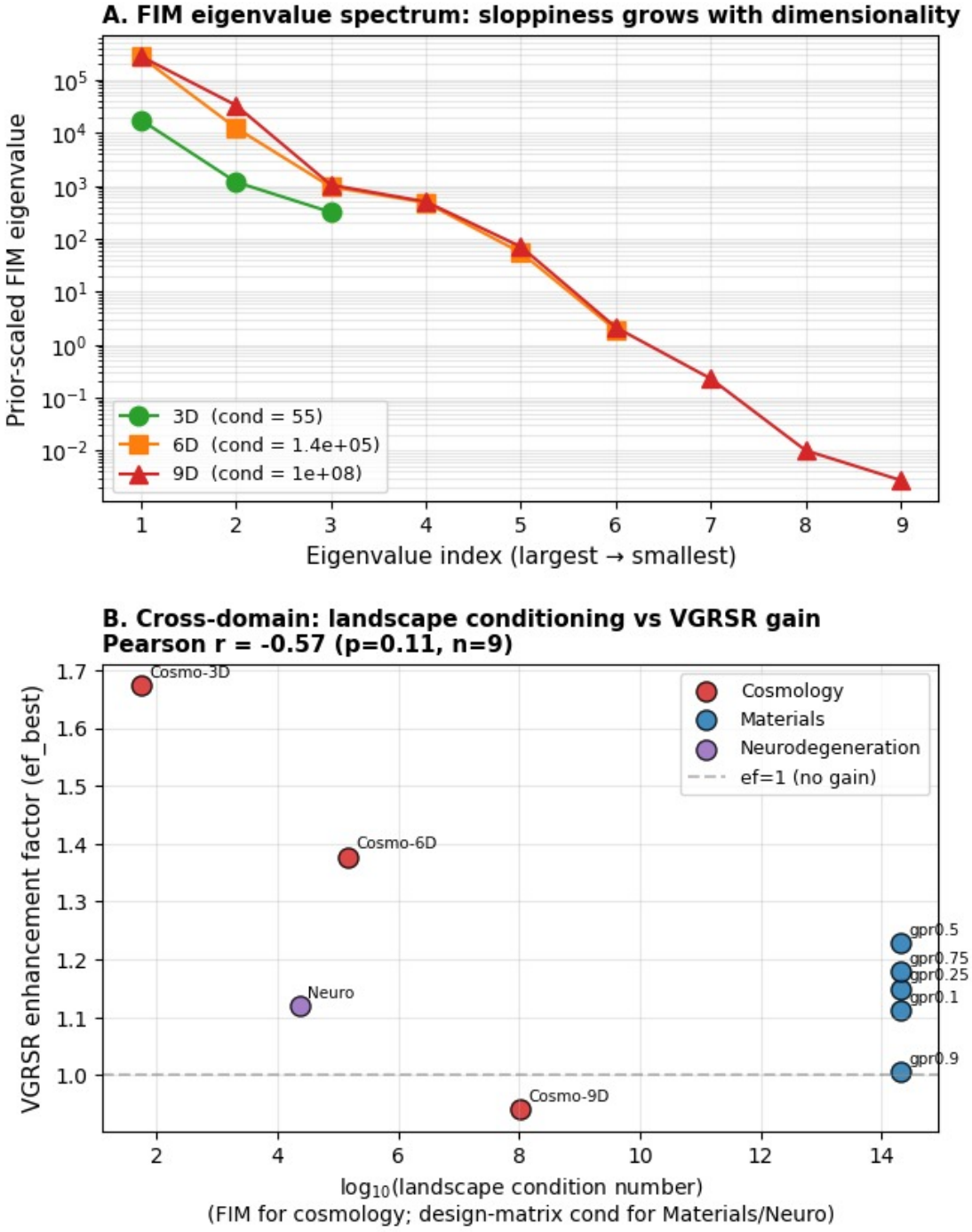

Figure B12. Cosmology sloppiness boundary condition for recursive emulator refinement. High-dimensional and ill-conditioned settings can erase active-learning gains.

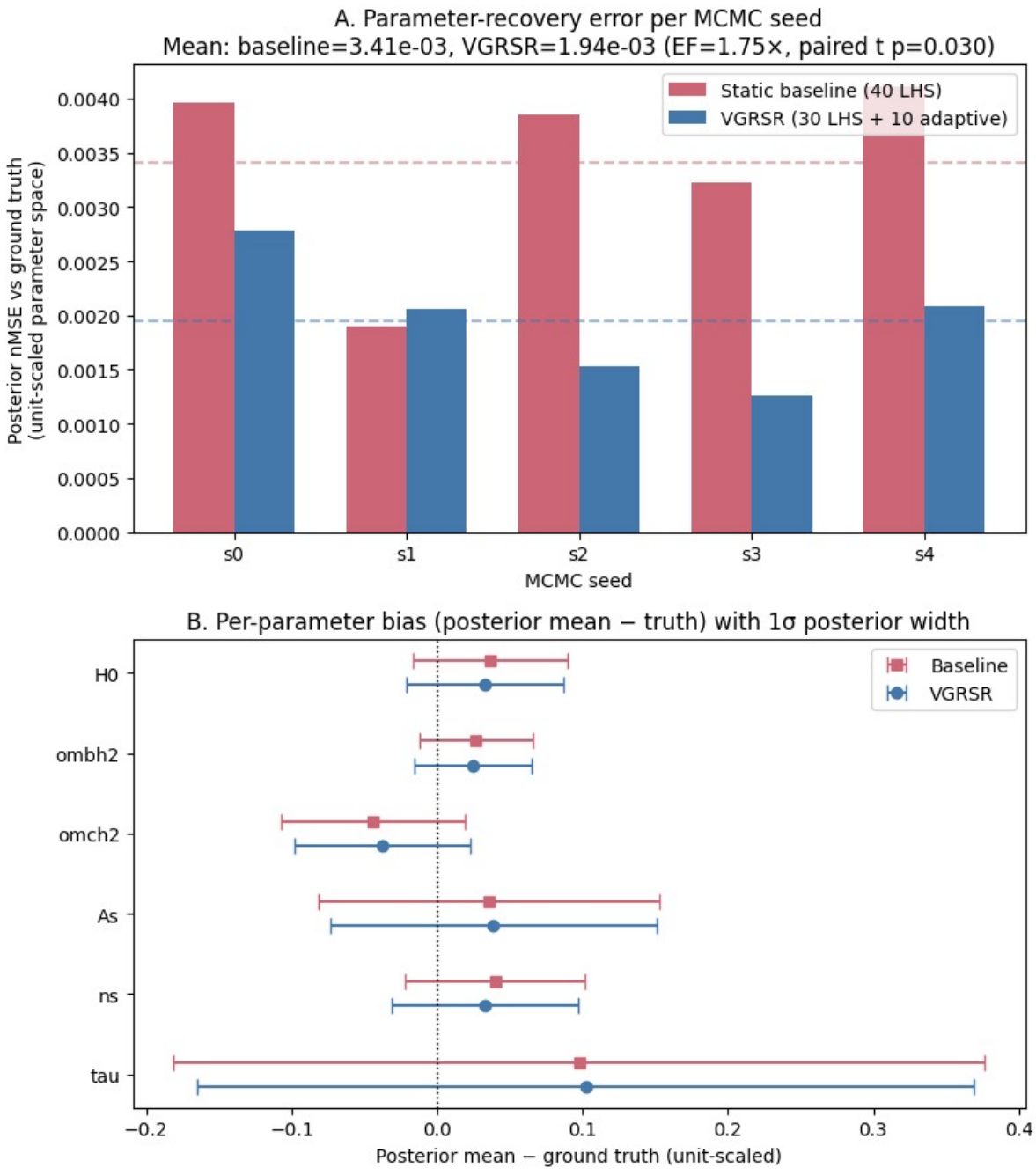

Figure B13. Cosmology MCMC diagnostic metrics. The plot reinforces the claim that apparent emulator improvements can be brittle under posterior-width and parameter-error diagnostics.

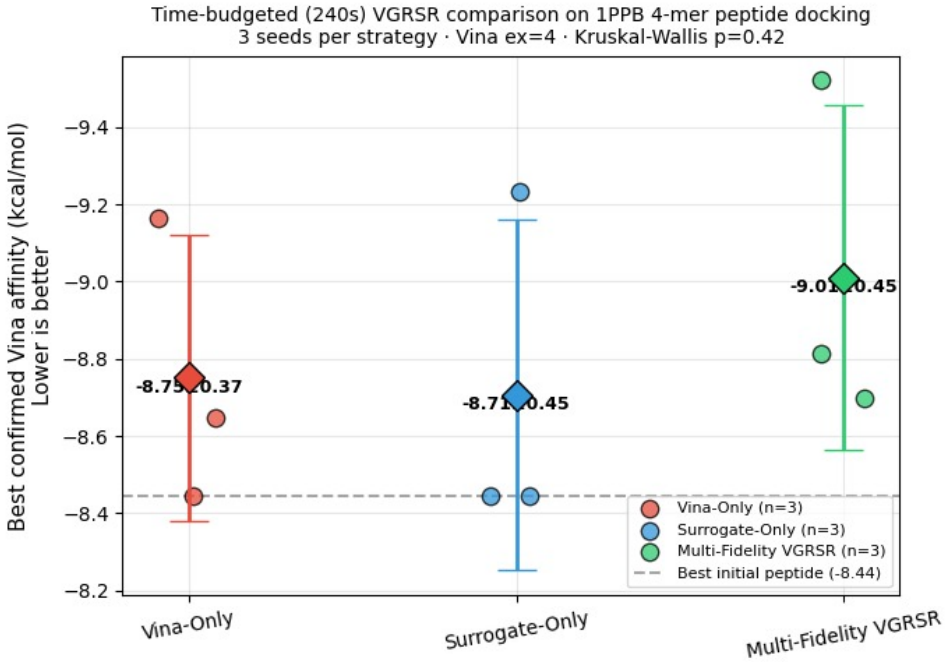

Figure B14. Time-budgeted nanobiomaterials docking vignette. The small budget makes the figure a latency boundary condition rather than a decisive performance claim.

Appendix C. Reference Support Map

The bibliography is organized around six support families. Proxy exploitation and self-training collapse support the negative claim (Goodhart, 1975; Manheim and Garrabrant, 2019; Denison et al., 2024; Wang et al., 2026; Gabor et al., 2025; Shumailov et al., 2024; Alemohammad et al., 2024; Hrubec, 2026). Self-correction and reflective agents define useful internal mechanisms that are insufficient as scientific verifiers (Madaan et al., 2023; Shinn et al., 2023; Valmeekam et al., 2023; Huang et al., 2023; Kamoi et al., 2024). Scientific-ML validity and self-driving labs motivate external validation, provenance, and operational constraints (King et al., 2009; Hase and Aspuru-Guzik, 2019; Burger et al., 2020; MacLeod et al., 2020; Abolhasani and Kumacheva, 2023; Szymanski et al., 2023; Tom et al., 2024; Fatehi et al., 2023; Chen et al., 2026; Kitchin, 2025; Luo et al., 2025; Zhu et al., 2025; Hu et al., 2025). Constrained, multi-fidelity, and surrogate-assisted optimization ground the gate-aware acquisition discussion (Forrester et al., 2007; Kandasamy et al., 2017; Li et al., 2020; Astudillo et al., 2021; Li et al., 2026; Do et al., 2023; Letham et al., 2019; Schoepfer et al., 2024; Hickman et al., 2023; Zhao et al., 2025). Control, risk, and safety support the stability claims (Zhou et al., 1996; Skogestad and Postlethwaite, 2005; Doyle et al., 1989; 1990; Ruth and Weller, 2010; Jansen et al., 2020; Rockafellar and Uryasev, 2000; Artzner et al., 1999; Pflug, 2000; Ruszczynski, 2010; Nguyen et al., 2021; Kouri and Shapiro, 2021). Domain anchors supply materials, cosmology, docking, neurodegeneration, and autonomous-lab examples (Curtarolo et al., 2012; Jain et al., 2013; Tsanas et al., 2010; Lewis et al., 2000; Lewis, 2013; Trott and Olson, 2010; Jumper et al., 2021; Abrams et al., 2023; Borrett et al., 2026; Gao et al., 2026; Chen et al., 2026b).