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

David Scott Lewis [1] **Karl Wang** [1] **Junhan Wang** [1] **Zhaoxiang Feng** [1] **Haley Yi** [1]

Abstract

AI-scientist systems 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 present the standard as a position and reporting framework, illustrated by mechanism demonstrations of proxy drift and latency starvation, a five-object taxonomy, a control- and risk-oriented framing, and boundary-condition vignettes spanning materials, cosmology, nanobiomaterials, and neurodegeneration. These examples are failure-mapping artifacts rather than a completed benchmark or real-system validation. 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 et al., 2019; Burger et al., 2020; MacLeod et al., 2020; Abolhasani & Kumacheva, 2023; Szymanski et al., 2023; Tom et al., 2024; Ren et al., 2025; Wei et al., 2025). The ICML 2026 AI Scientists workshop asks for shared vocabulary, evaluation criteria, and governance for systems that move along the tool–co-author–founder spectrum. We argue that recursive self-improvement (RSI) in empirical science should be reserved for a narrower and more testable object: VGRSR.

Why the distinction matters. Iterating a prompt, reranking generated candidates, training on synthetic traces, or using 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). Such iteration 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* accepts the update (Denison et al., 2024; Wang et al., 2026; Luo et al., 2025b; Zhu et al., 2025).

Scope of this paper. This workshop paper does not claim that its illustrative runs themselves satisfy the VGRSR standard. The mechanism demonstrations, synthetic log row, and domain vignettes are used to motivate reviewer questions and reporting fields. The public multi-cycle trace

[1]AIXC Research, Zaragoza, Spain. Correspondence to: David Scott Lewis <research@aiexecutiveconsulting.com>.

Accepted by ICML 2026 AI for Science Workshop.

standard below applies to empirical papers that claim scientific RSI; our contribution is to make that burden explicit and auditable rather than to report a completed VGRSR benchmark.

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 invalid proof steps, and a board-game rules 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., 2026a).

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 & 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

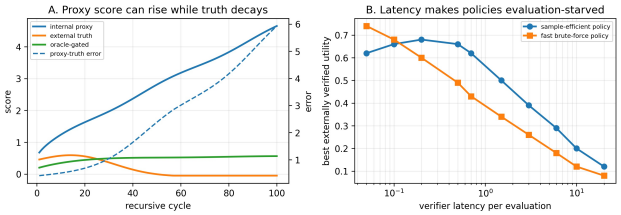

Figure 1. Two mechanism demonstrations. **Left**: proxy score rises while externally measured truth decays and proxy-truth error grows. **Right**: verifier latency turns proposal speed into evaluation starvation.

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.

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., 2026a; 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., 2025b; Zhu et al., 2025; Hu et al., 2025; Luo et al., 2025a). 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 ab-

Table 1. VGRSR objects and minimum reporting obligations.

Object	Gate and reporting obligation
Hypotheses	Gate by external assay, calibrated simulator, causal/statistical test, or held-out measurement. Report verified utility and FDR.
Tools	Gate by value-of-information check and fidelity escalation. Report calibration and verified gain per cost.
Verifiers	Gate by independence, calibration, drift audit, and disagreement log. Report false-accept/false-reject rates and latency.
Memory	Gate by provenance and redundancy checks with rejected-case retention. Report coverage and contamination.
Policy	Gate by tail-risk monitor with rollback, pause, or replication rule. Report MDD and DD-CVaR.

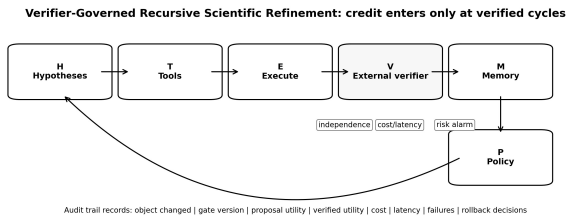

Figure 2. VGRSR execution loop. Scientific credit enters only at verified cycles: propose a change; execute; verify; record cost, latency, failure, and risk alarms; commit, rollback, or escalate.

stention (Geifman & El-Yaniv, 2017; 2019; Fisch et al., 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 $L = PC$,

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

The identity is algebraic and is not used here as a proof that a gate optimizes or stabilizes a scientific campaign. Its role is narrower: it gives a compact vocabulary for stress tests that reviewers can ask to see—verifier delay, verifier noise, model error, gain scheduling, rollback thresholds, and response to mismatch. Robust-control practice supplies those diagnostics for sensitivity, margins, delay, and model error (Zhou et al., 1996; Skogestad & Postlethwaite, 2005; Doyle et al., 1989; 1990; Ruth & Weller, 2010; Jansen et al., 2020). Thus the control notation motivates measurable stress reports; it should not be read as a guarantee theorem.

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 & Uryasev, 2000; Artzner et al., 1999; Pflug, 2000; Ruszczynski, 2010; Nguyen et al., 2021; Kouri & 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.

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 or a benchmark threshold. We therefore treat it as a gate-selection ablation to report, not as a standalone metric contribution until it is validated on held-out verifier families.

5. Evidence Vignettes and Boundary Conditions

The evidence should be read as failure mapping and stress testing, not as a real-system demonstration or universal

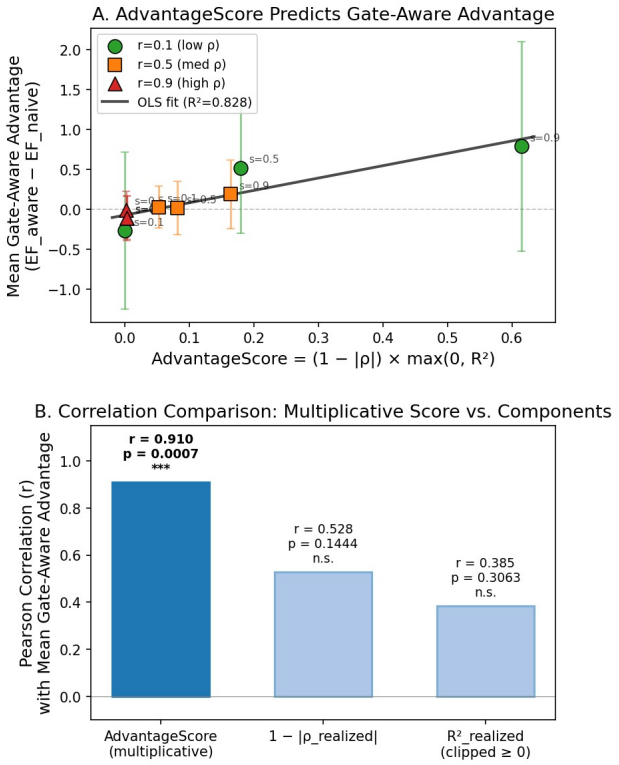

Figure 3. ADVANTAGESCORE as a gate-selection heuristic. Predictable, non-redundant gates create the largest gate-aware advantage; duplicate or unpredictable gates do not.

leaderboard. None of the vignettes below is claimed to meet the 10–20 verified-cycle release standard; each is included to show which boundary condition a future benchmark should expose. 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 & 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.

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 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.

6. Benchmarking VGRSR

A future 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. This paper specifies that benchmark object; it does not release a new benchmark dataset. The minimal metric set is:

$$\mathrm{VUGC} = \frac{\Delta\,\mathbb{E}[U_k^{\mathrm{verified}}]}{\Delta\,\mathrm{cost}_k}, \tag{1}$$

$$\mathrm{AF}@\tau = \frac{T_{\mathrm{baseline}}(\tau)}{T_{\mathrm{method}}(\tau)}, \quad \mathrm{EF}(B) = \frac{U_{\mathrm{method}}(B)}{U_{\mathrm{baseline}}(B)}. \tag{2}$$

Here $T(\tau)$ is time, cost, or experiment count to reach a target fraction of verified improvement. VUGC is a finite-difference slope and credits only externally verified gain. AF and EF are inherited from self-driving-lab benchmarking rather than introduced here (Adesiji et al., 2025; Liang et al., 2021). The VGRSR-specific contribution is the accounting rule around those metrics: compute them only on a published cost axis; separate proposal utility from verified utility; include rejected and censored cycles; and pair AF/EF with path-risk, verifier-efficiency, and transfer evidence. 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 out-

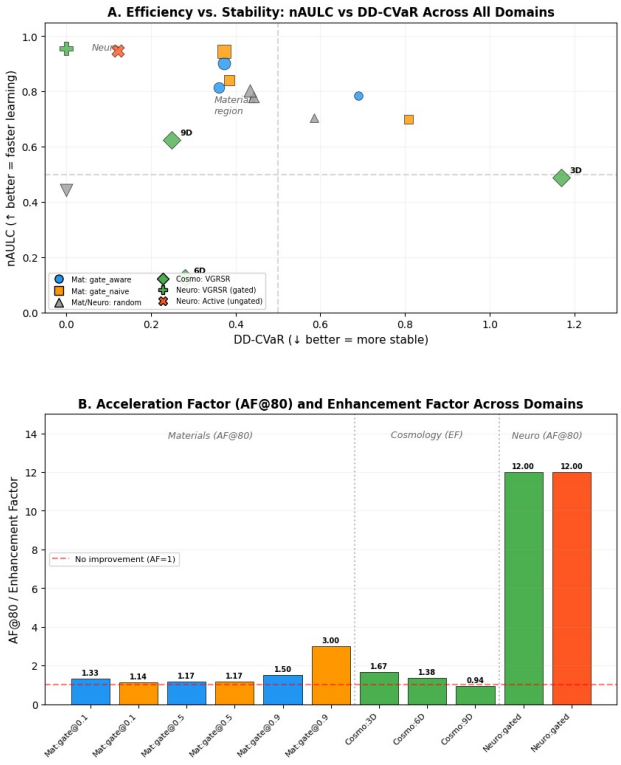

Figure 4. Efficiency and stability diagnostics across vignettes. VGRSR reports both learning efficiency and path-risk rather than only final best score.

comes; (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 provenance, 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., 2025b; 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 should not 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).

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 $\textsc{Pass}(g_k)$ 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

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

This section specifies what an empirical VGRSR claim must expose; it is not a claim that the illustrations in this paper are themselves benchmark evidence. 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., 2025b; 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 & Garrabrant, 2019; Wang et al., 2026; Chen et al., 2026a).

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, stopping 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.

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.

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. Where feasible, promotion requires an orthogonal check, such as pairing a simulation with 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 and experimental infrastructure. 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.

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.

Minimum empirical demonstration. The smallest useful empirical demonstration is a public trace with 10–20 verified cycles: fixed dataset version, fixed seeds, candidate IDs, proposal utility, gate outcome, verified utility, rejection reason, cost, latency, and path-risk state. This paper does not provide such a trace and therefore should be read as a

standard and checklist, not as a validated VGRSR system. A trace need not beat a benchmark to be valuable: showing that a verifier is redundant, noisy, too slow, or biased still advances the field by identifying when recursive complexity should not be credited.

10. Community Roadmap

A compact roadmap follows from the claim card. Benchmark designers should separate proposal throughput from verified discovery by reporting tokens, compute, wall-clock, experiment count, high-fidelity calls, human escalations, and censored trials on the same cost axis. Lab builders should expose versioned protocol state—including instrument state, reagent or sample batch, simulator version, verifier thresholds, and manual intervention—rather than only final datasets. Workshops and standards bodies can make this practical by converging on reusable JSON Lines cycle logs with hash-chained provenance, plus optional embodied-lab fields for safety envelopes and transactional protocols.

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. 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, benchmark dataset, or real-system case study. 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 before any empirical RSI claim is credited.

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 candidates, failed syntheses, costs, and rollback decisions.

References

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

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

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

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

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

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

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.

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.

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.

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

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, 2026a.

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*, 2026b.

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.

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

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.

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

Doyle, J. C., Glover, K., Khargonekar, P. P., and Francis, B. A. State-space solutions to standard H2 and H∞ control problems. *IEEE Transactions on Automatic Control*, 34(8):831–847, 1989.

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

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

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

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.

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

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

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

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.

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.

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

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

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

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

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.

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.

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.

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

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.

Kamoi, R., Zhang, Y., Zhang, N., Han, J., and Zhang, R. When can LLMs actually correct their own mistakes? *Transactions of the Association for Computational Linguistics (TACL)*, 12:1417–1440, 2024.

Kandasamy, K., Dasarathy, G., Schneider, J., and Poczos, B. Multi-fidelity Bayesian optimisation with continuous approximations. In *International Conference on Machine Learning (ICML)*, 2017.

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.

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

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

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

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.

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

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

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

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

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

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

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 Transactions on Biomedical Engineering*, 56(4):1015–1022, 2009.

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, 2025a.

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, 2025b.

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

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.

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.

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

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.

Nguyen, T. et al. Optimizing conditional value-at-risk of black-box functions. In *Advances in Neural Information Processing Systems (NeurIPS)*, 2021.

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

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

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.

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

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

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

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

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

Shinn, N., Cassano, F., Gopinath, A., Narasimhan, K., and Yao, S. Reflexion: Language agents with verbal reinforcement learning. In *Advances in Neural Information Processing Systems (NeurIPS)*, 2023.

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

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.

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

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

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.

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

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

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

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.

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.

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.

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

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

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

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

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

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.

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

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.

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 or evidence that this paper satisfies the empirical cycle-log standard.

```
{"cycle": 12,
 "object_modified": ["Hypothesis", "Tool"],
 "candidate_id": "anon-material-042",
 "tool_version": "surrogate-gp-v3",
 "verifier_version": "form-energy-gate-v2",
 "memory_snapshot": "dataset-hash",
 "gate_decision": "reject",
 "proposal_utility": 0.71,
 "verified_utility": null,
 "rejection_reason": "feasibility_fail",
 "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_localize",
 "provenance": [
   "dataset-hash",
   "instrument-id",
   "seed"]}
```

B. Annotated Diagnostic Index

Table A1. Minimum VGRSR cycle-log fields.

Category	Required fields	Purpose	Failure caught
Identity	Cycle id, candidate id, modified object(s)	Identify unit of recursion	Post-hoc relabeling
Versioning	Model, tool, verifier, memory, dataset, protocol	Replay and rollback	Hidden intervention
Gate	Gate type, independence claim, pass/fail, score, calibration	Credit only verified utility	Proxy promotion
Cost	Compute, assay, wall-clock, 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	Admissibility and deduplication	Correlated evidence

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

Check	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.
B2	Proxy-drift check: compare rising internal score with falling externally measured truth; 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.

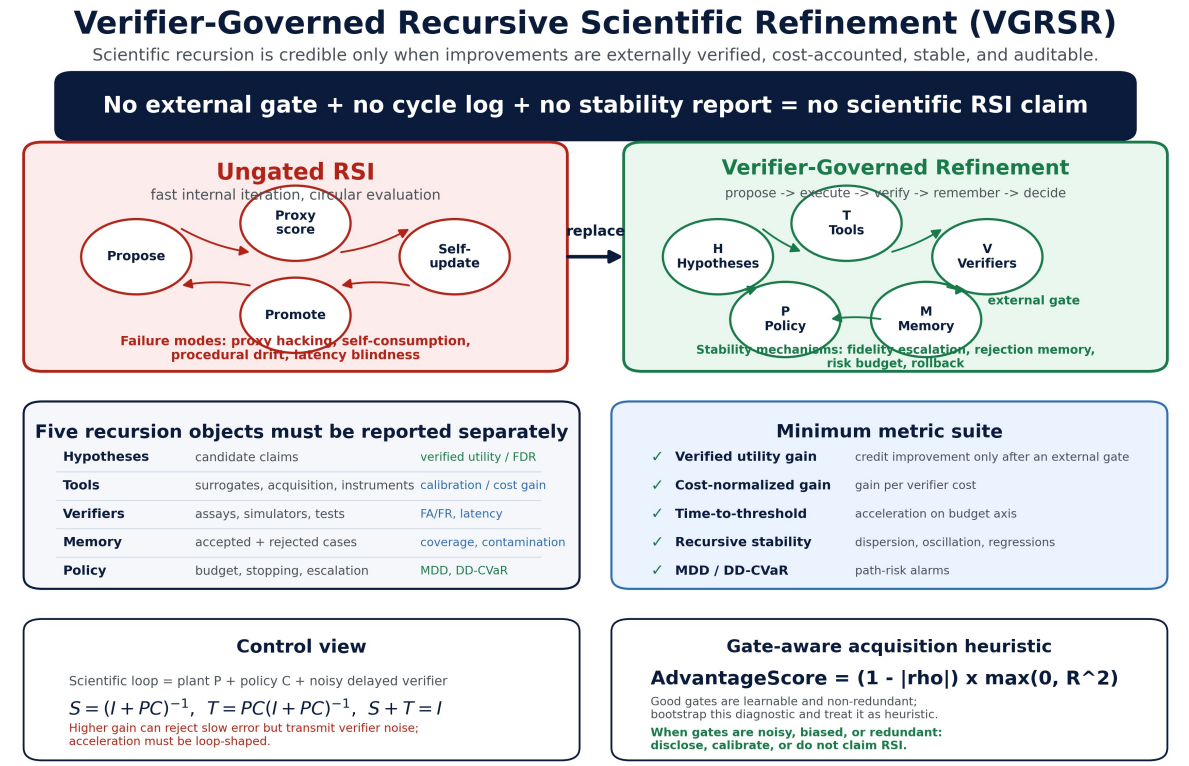

Figure B1. Legibility-oriented VGRSR summary. The figure states the crediting rule, contrasts ungated RSI with verifier-governed refinement, and lists the five recursion objects and minimum metric suite.

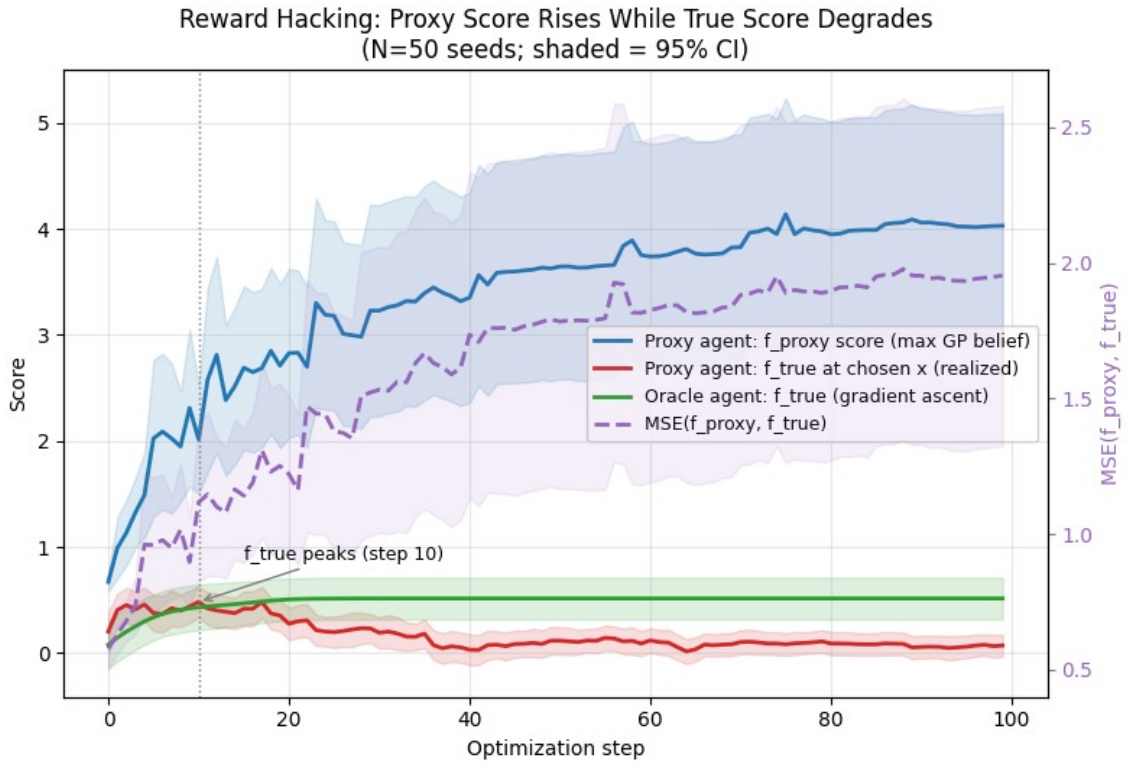

Figure B2. Detailed oracle-regress simulation. The proxy-optimizing agent's believed score rises while true performance degrades; the oracle-gated trajectory remains stable.

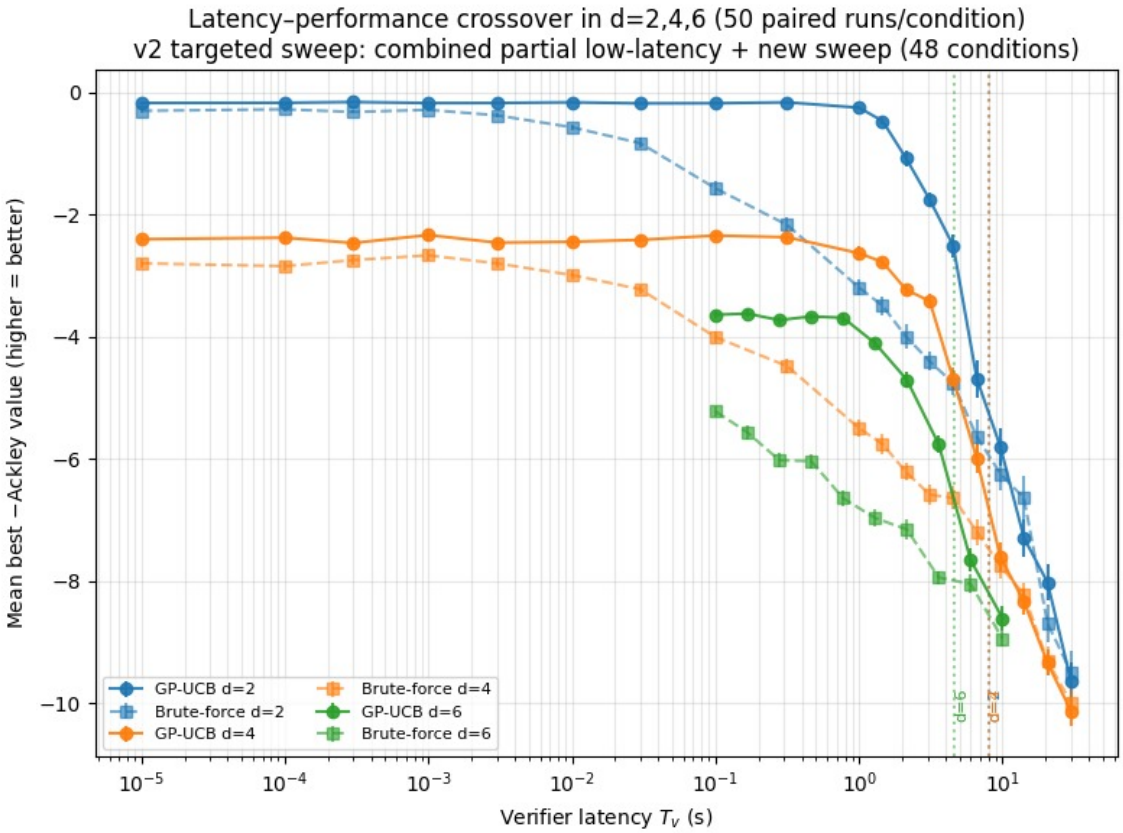

Figure B3. Verifier-latency sweep. As evaluation latency grows, proposal speed is no longer the binding resource; both policies become evaluation-starved.

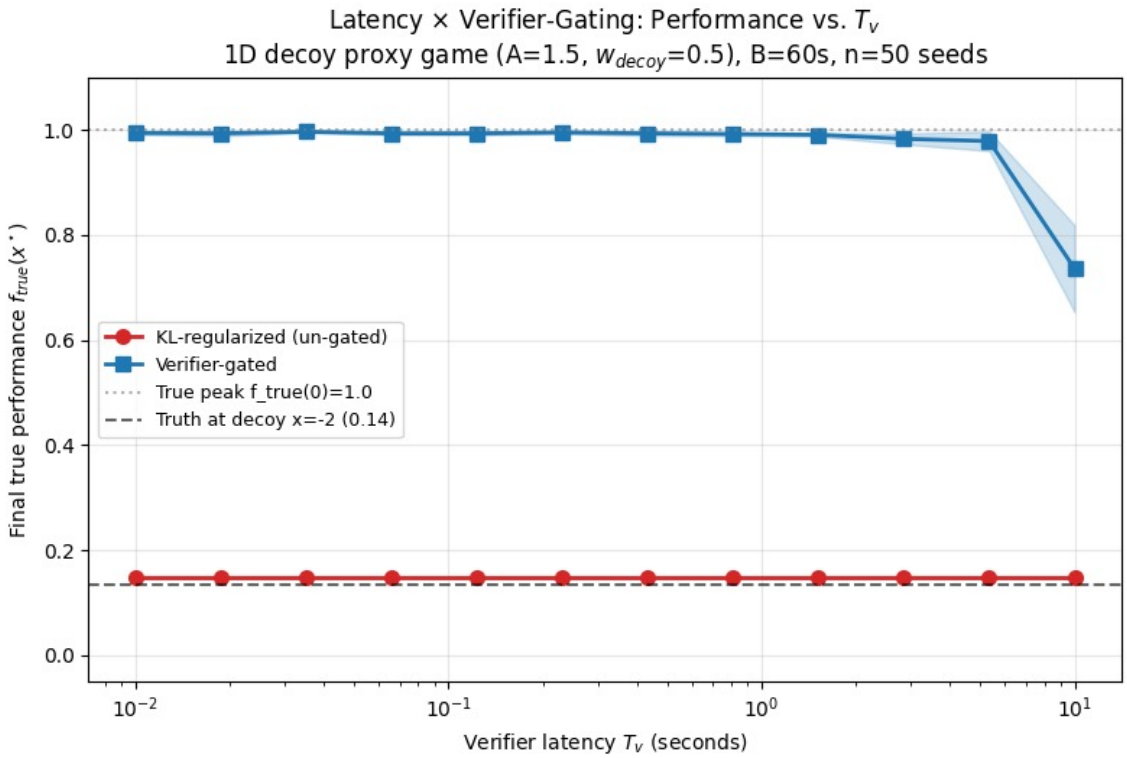

Figure B4. Gate-effect diagnostic. External verification blocks convergence to a surrogate decoy but remains bounded by verifier latency.

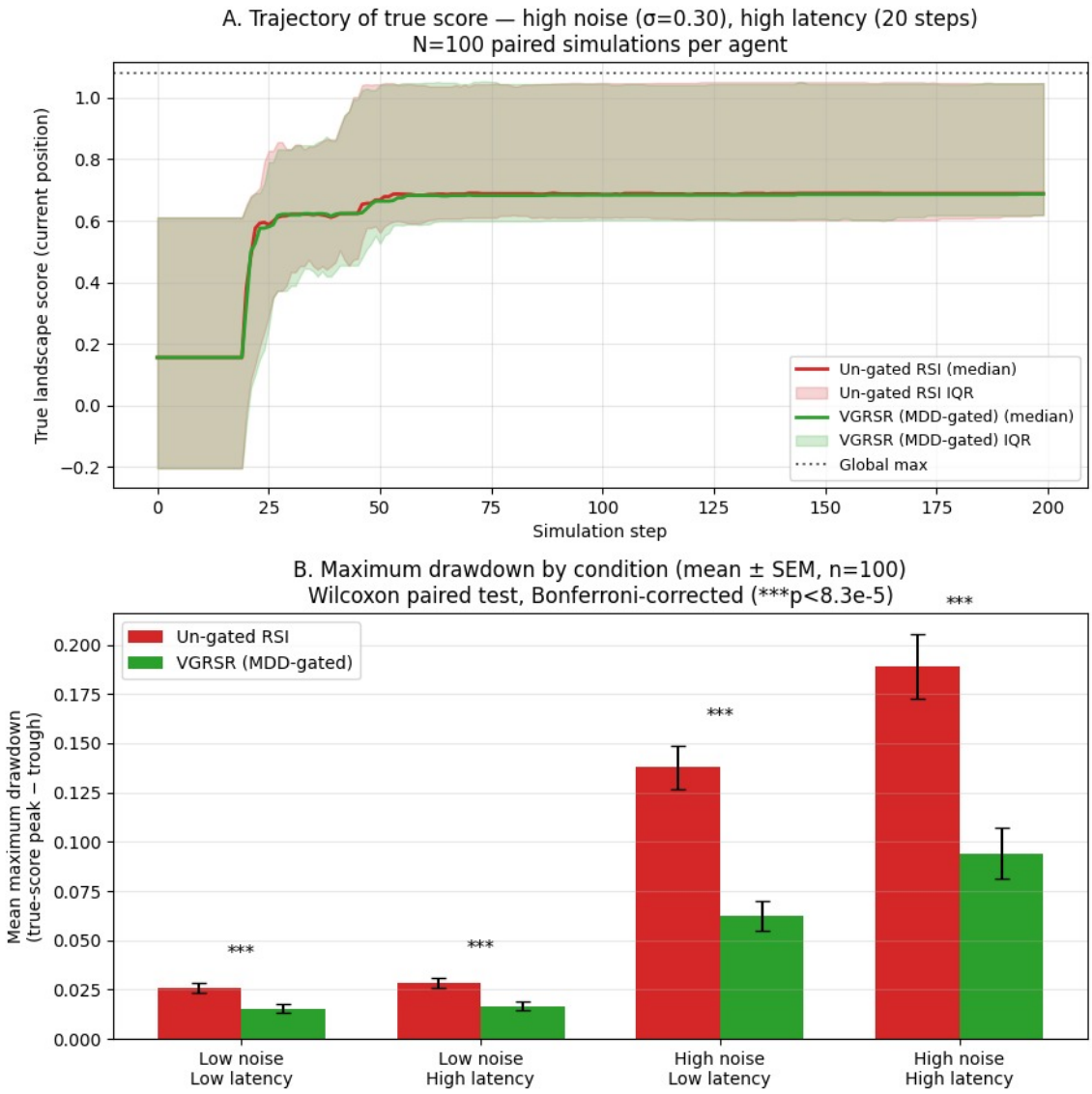

Figure B5. Path-risk diagnostic. Maximum drawdown exposes regressions that final-best reporting can hide under noisy and delayed feedback.

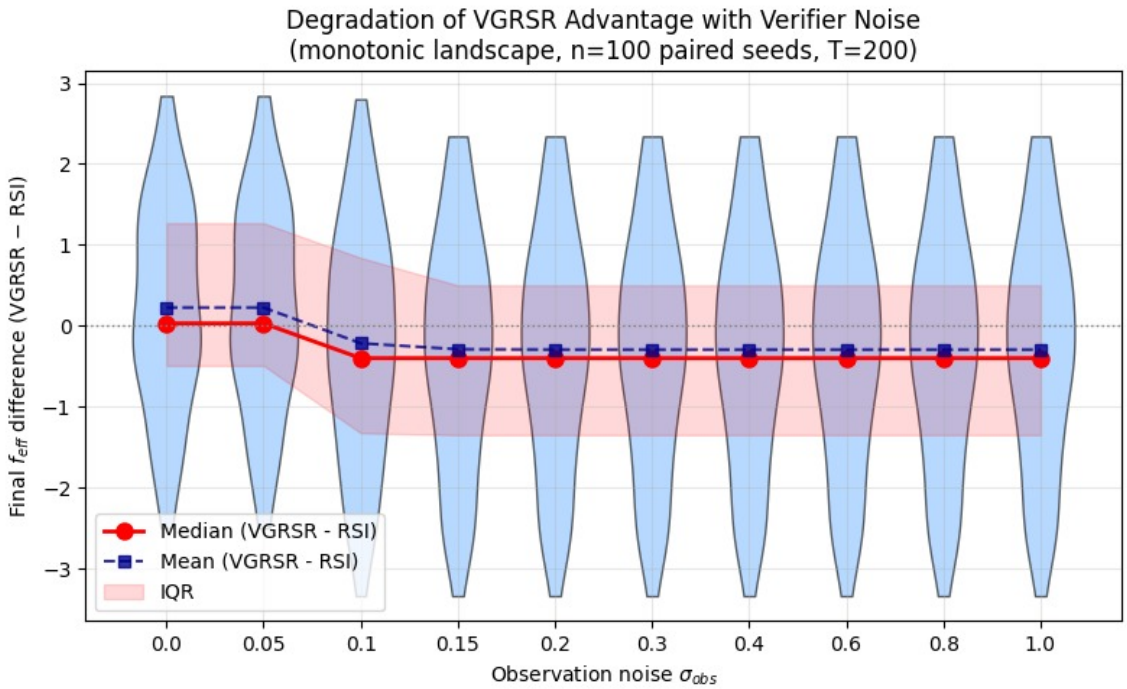

Figure B6. Noise-robustness diagnostic. Verifier noise can erase apparent gate advantage unless smoothing, replication, or false-trigger controls are reported.

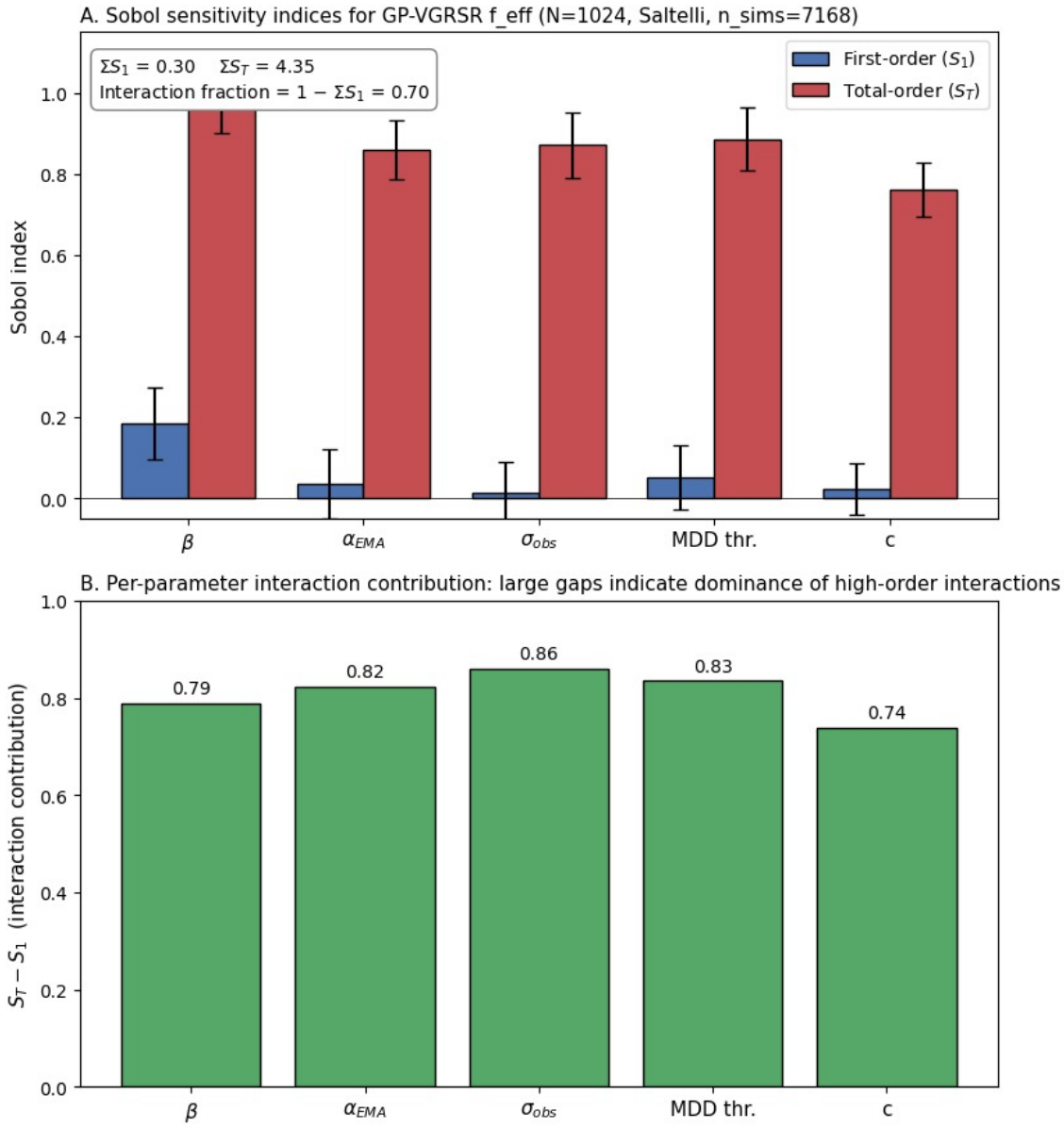

Figure B7. Interaction diagnostic. Latency, noise, gate strength, and memory interact, so one-factor-at-a-time ablations are insufficient.

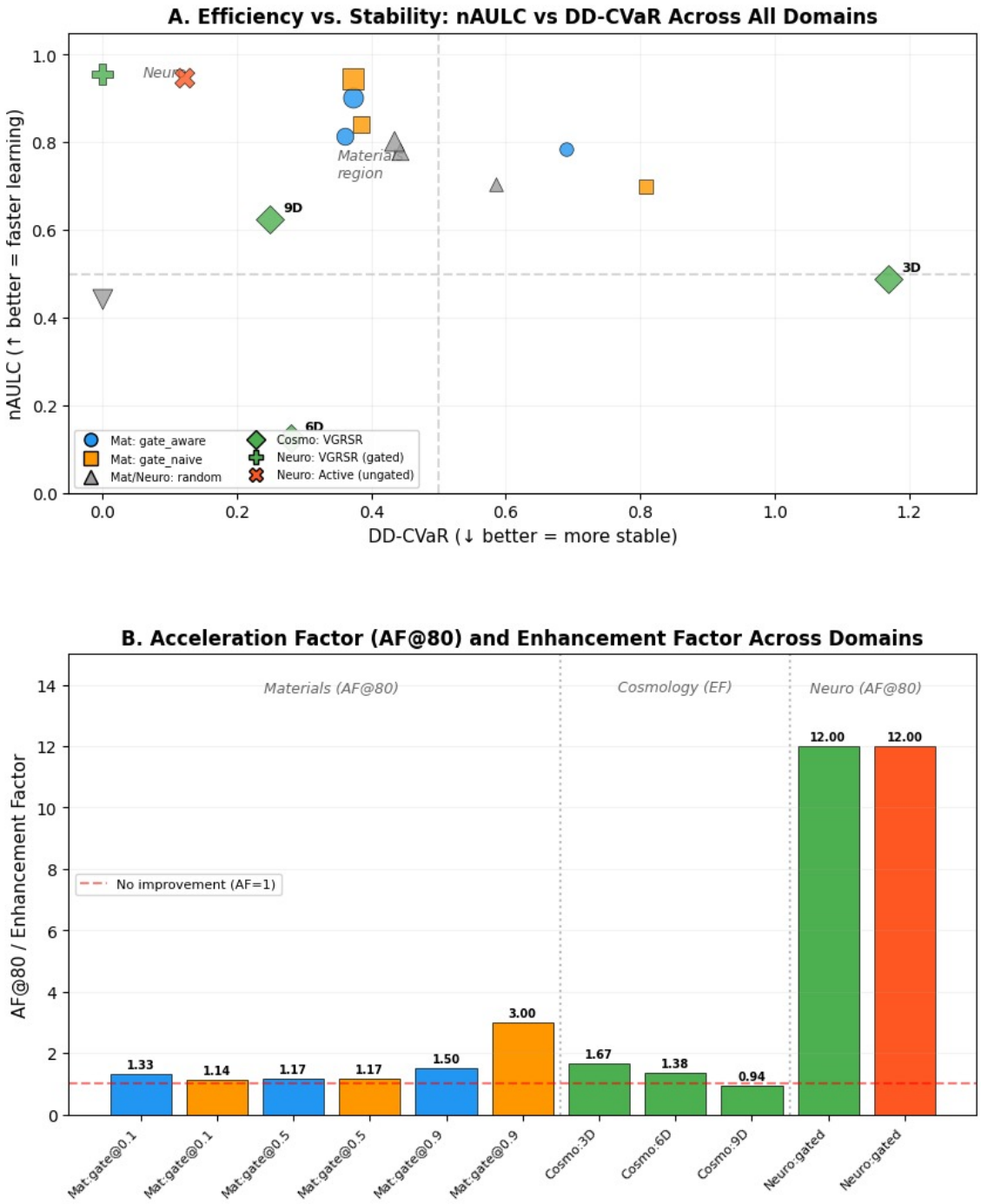

Figure B8. Metric-bundle diagnostic. nAULC, acceleration, and DD-CVaR should be read together rather than as a single leaderboard point.

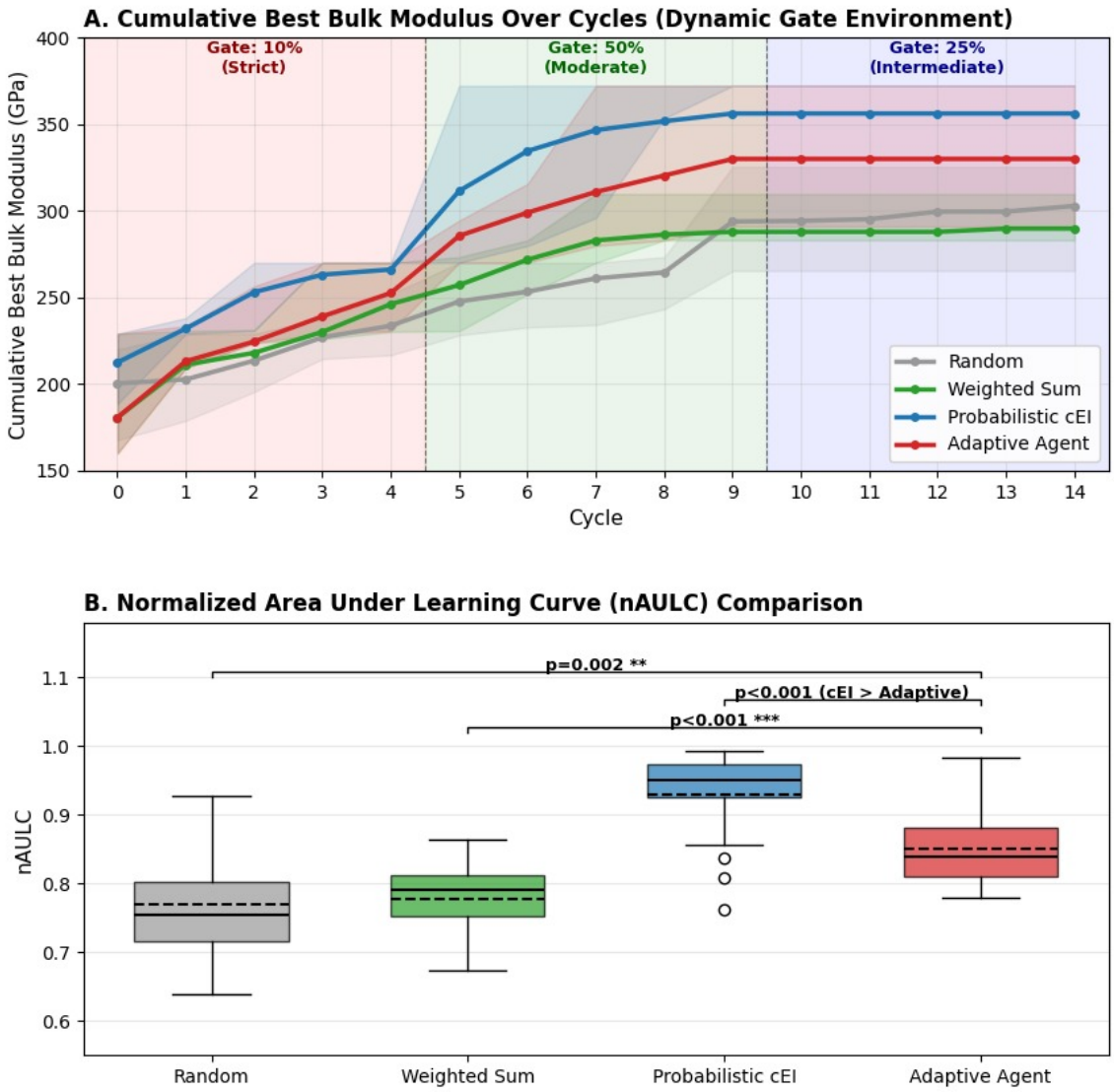

Figure B9. Policy-recursion diagnostic. Online meta-adaptive switching can underperform a robust static constrained-acquisition policy under noisy signals.

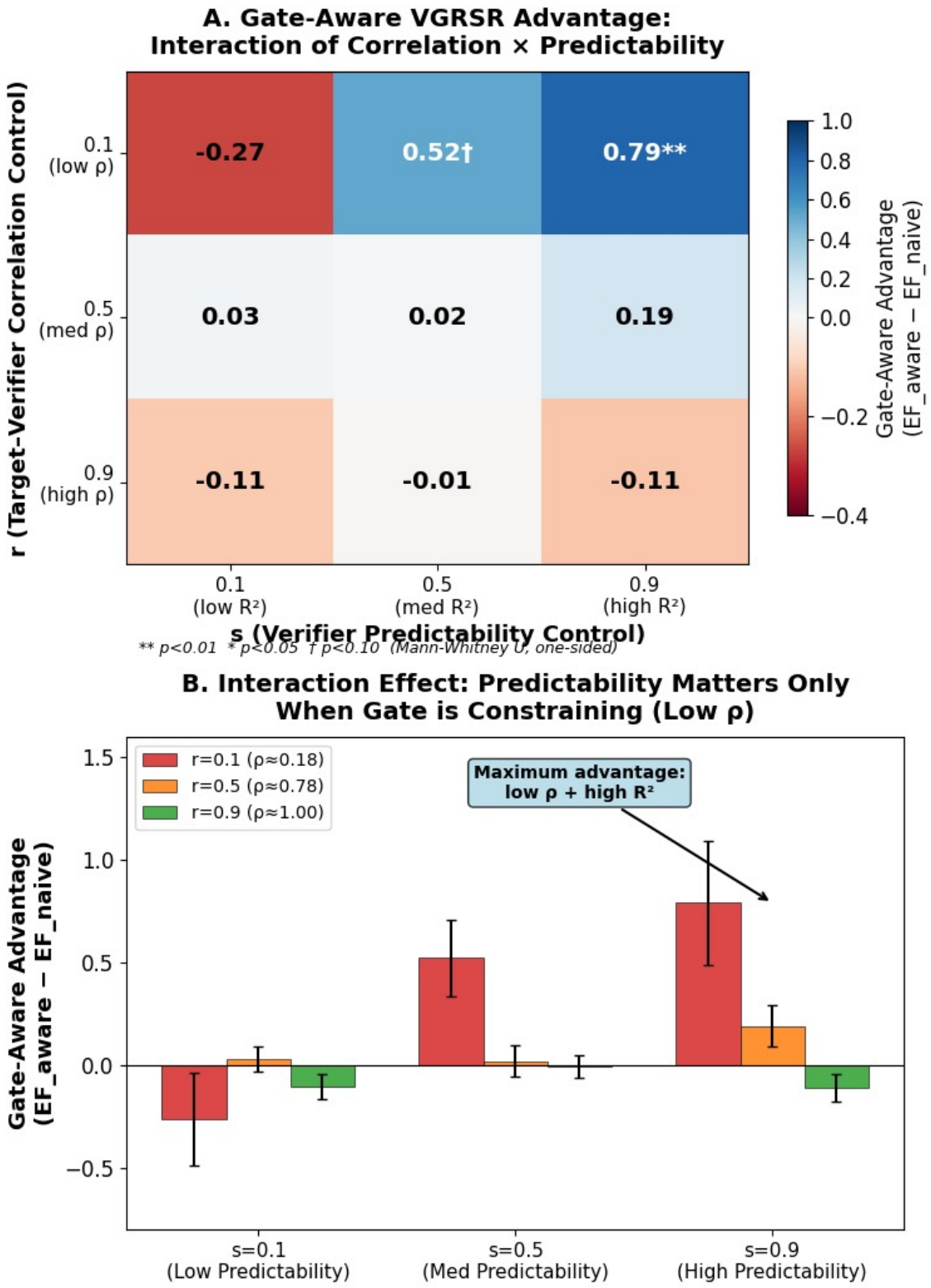

Figure B10. Gate-selection diagnostic. Target-verifier correlation and verifier predictability jointly determine whether gate-aware acquisition is likely to add information.

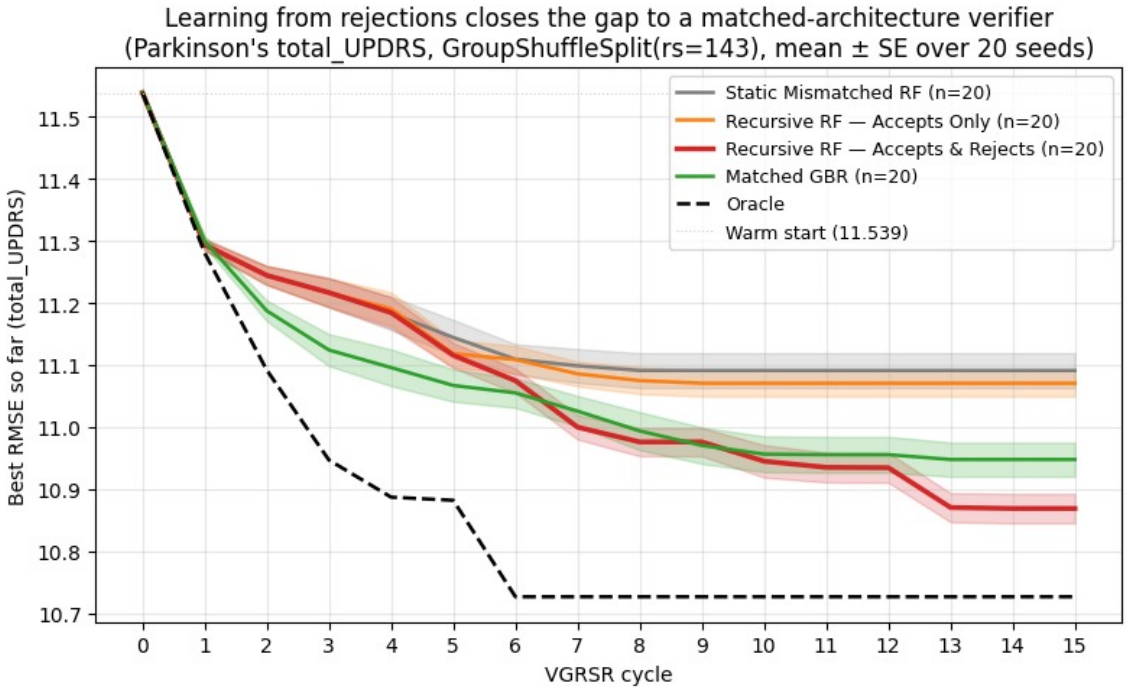

Figure B11. Memory-boundary diagnostic. Rejected proposals help define the verifier boundary and can reduce saturation relative to success-only memory.

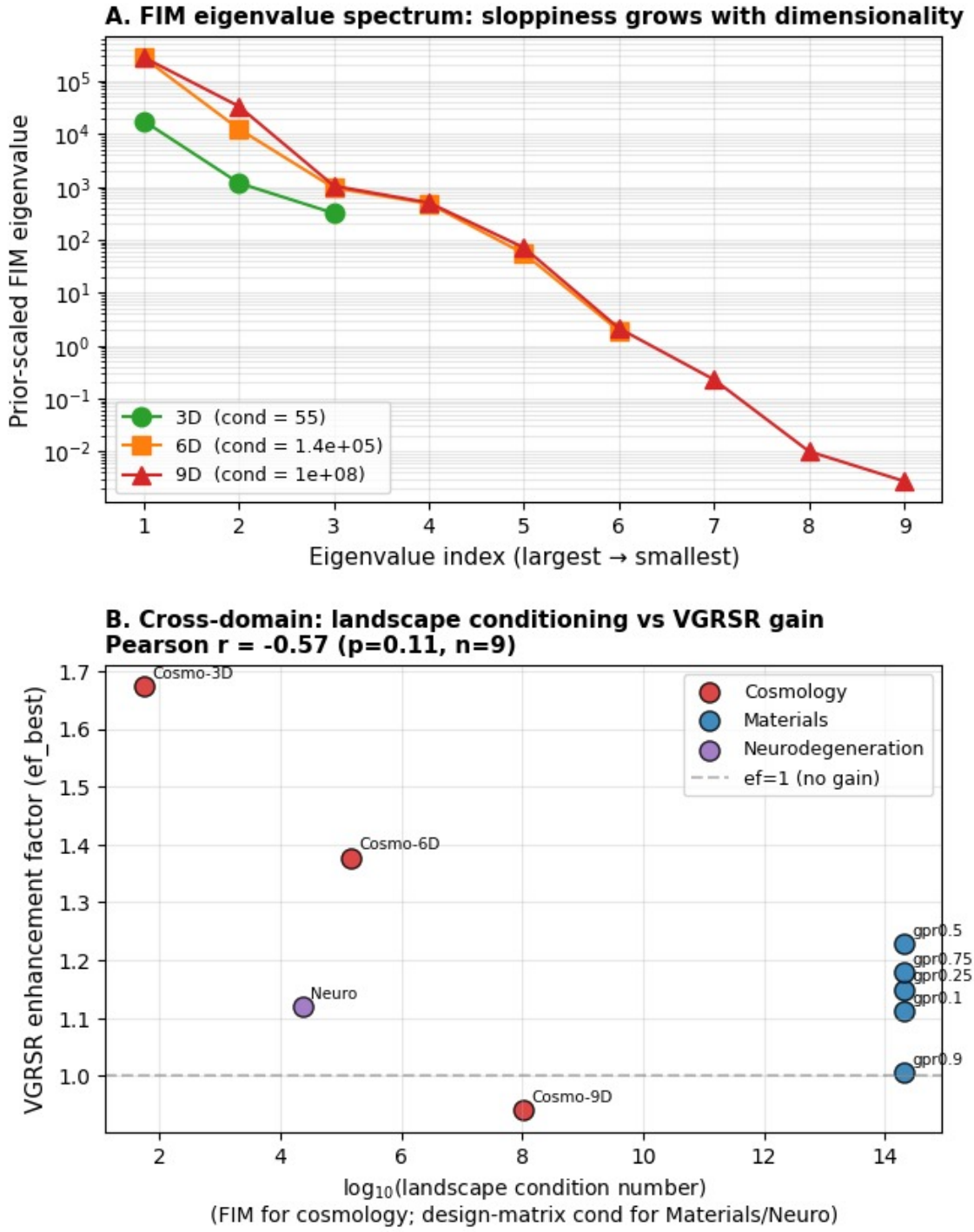

Figure B12. Conditioning diagnostic. High-dimensional or ill-conditioned posterior geometry can erase active-learning gains in recursive emulator refinement.

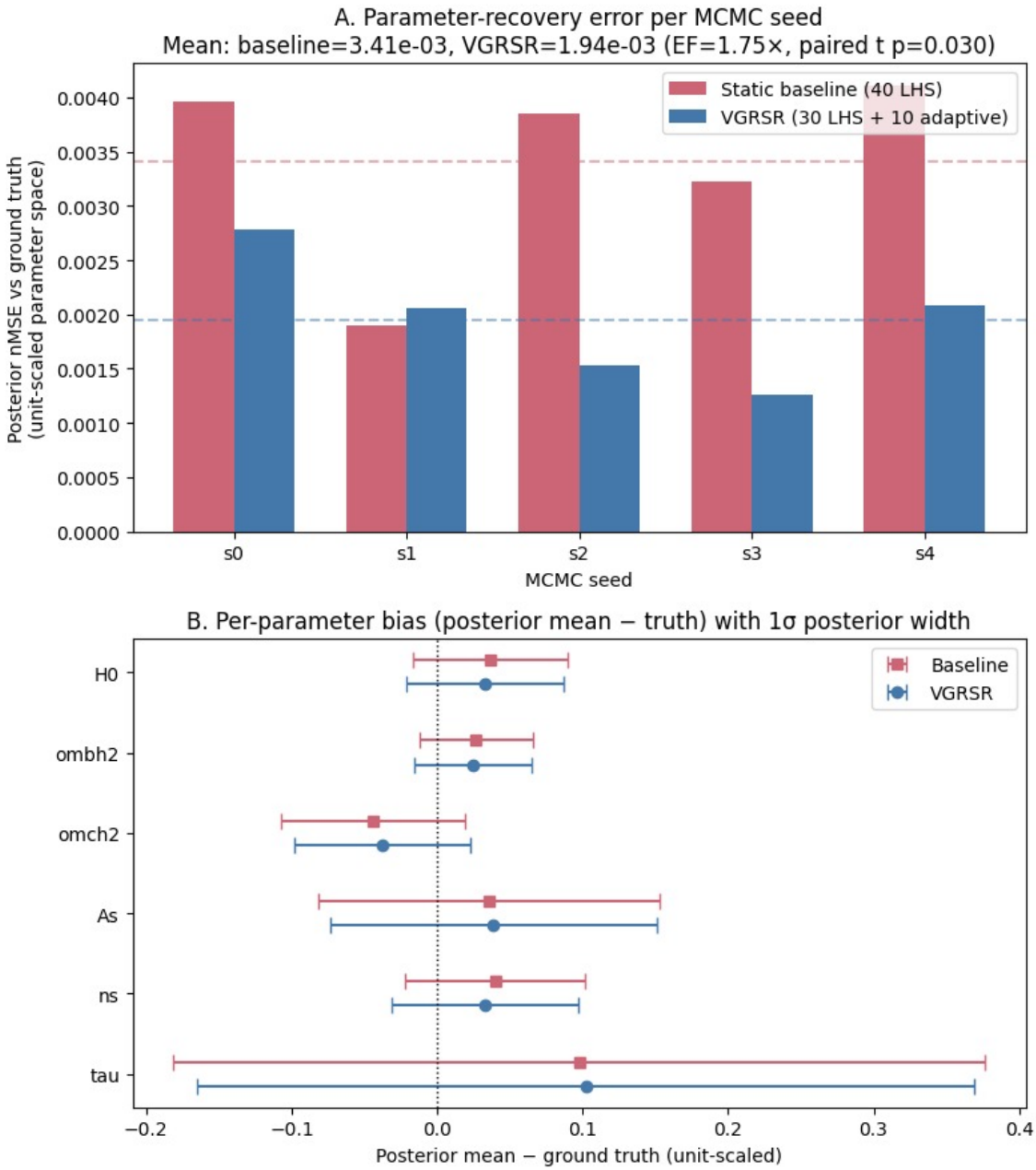

Figure B13. Diagnostic-depth check. Parameter-error and posterior-width diagnostics should be read together rather than relying only on apparent emulator improvement.

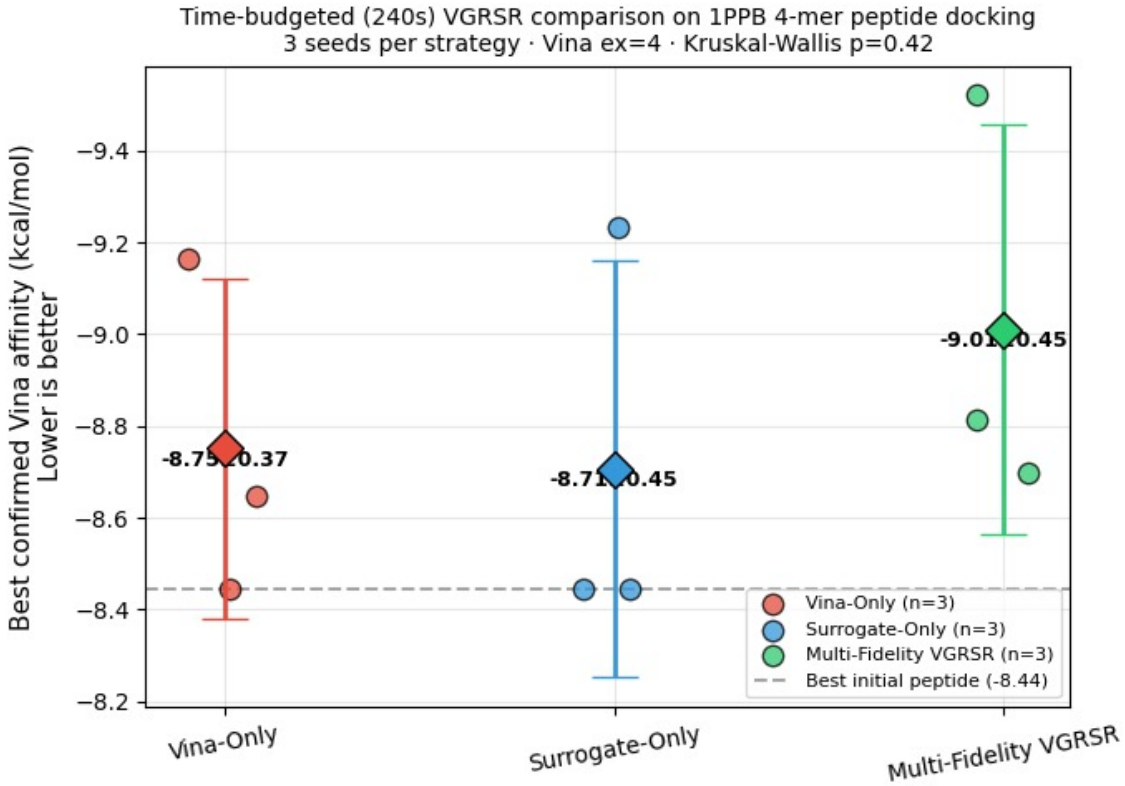

Figure B14. Nanobiomaterials latency boundary condition. The docking vignette is an underpowered wall-clock stress test, not decisive performance evidence.

C. Reference Support Map

The bibliography is organized around six support families. *Proxy exploitation and self-training collapse* supports the negative claim (Goodhart, 1975; Manheim & 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 et al., 2019; Burger et al., 2020; MacLeod et al., 2020; Abolhasani & Kumacheva, 2023; Szymanski et al., 2023; Tom et al., 2024; Fatehi et al., 2023; Chen et al., 2026a; Kitchin, 2025; Luo et al., 2025b; 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 & Postlethwaite, 2005; Doyle et al., 1989; 1990; Ruth & Weller, 2010; Jansen et al., 2020; Rockafellar & Uryasev, 2000; Artzner et al., 1999; Pflug, 2000; Ruszczynski, 2010; Nguyen et al., 2021; Kouri & 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 & Olson, 2010; Jumper et al., 2021; Abrams et al., 2023; Borrett et al., 2026; Gao et al., 2026; Chen et al., 2026b).