# EUREKA IS A DISTRIBUTION: EXECUTABLE REGIME FEEDBACK FOR LANGUAGE-AGENT ALGORITHM DISCOVERY

**Anonymous authors**
Paper under double-blind review

## ABSTRACT

An algorithm earns its name by surviving beyond the evaluator that shaped it. We call this criterion *distributional survival*: retaining an advantage across generator and scale shifts that never shaped search. We introduce REGIMEGYM, a controlled four-domain testbed, and REGIMEFORGE, a training-free discovery loop that exploits a useful asymmetry: model proposals are scarce, while executable falsification is cheap. CPUs mine an 80-regime atlas, convert diverse failures into compact behavior dossiers, and direct a global explorer plus a targeted repair. Across three model families and 12 paired runs, REGIMEFORGE improves all nine nonzero cell effects and ties three; every domain and model family has a positive mean. Median hidden-regret reduction is 1.06 points, the library-independent geometric objective falls 6.82%, and mean regret drops from 22.43% to 12.74%. Frozen programs evaluated on 320 newly drawn hidden regimes retain 9/2/1 positive/tied/negative cells, a 1.84-point median, a 8.85% geometric gain, and a positive crossed-axis interval. Crossed mechanism replications show a 46.60-point full-atlas effect in GLM/cache and a 2.12-point, 4/4-positive effect in GLM/load. Matched removal gives behavior deltas a 4.04-point median advantage in 6/8 paths; ranked and random family-diverse dossiers improve all 16 matched arms. The 84-call main study costs $4.73 because models propose while CPUs falsify. Together, these results establish executable falsification as a practical, affordable foundation for agentic algorithm discovery.

## 1 INTRODUCTION

Consider an agent asked to discover a cache-replacement policy. It proposes a compact formula, executes it on a development workload, receives a scalar score, and revises the formula until the score improves. On steady popularity traces the result looks excellent. A popularity shift makes the policy retain yesterday's frequent objects, repeatedly evicts expensive new objects, and costs more than the policy it replaced. The agent completed every step of the usual discovery loop; what failed was the evidence standard. A single evaluator distribution produced a locally successful program that fails the algorithmic standard.

This distinction matters as language models move from solving instances to writing the algorithms that solve future instances. FunSearch and subsequent systems demonstrate that model-guided program search can produce competitive mathematical and optimization programs (Romera-Paredes et al., 2024; Liu et al., 2024; Ye et al., 2024; van Stein & B"ack, 2024). Recent agentic systems make this process increasingly autonomous: HeuriGym evaluates long-horizon heuristic construction, ShinkaEvolve improves proposal efficiency, TusoAI optimizes scientific methods, and InnoGym measures both novelty and performance (Chen et al., 2026; Lange et al., 2026; Turcan et al., 2026; Zhang et al., 2026). These advances make the next question unavoidable: *what evidence elevates an evaluator-winning program into an algorithm?*

Our answer is to make distributional falsification part of the discovery loop. A model proposal is expensive but an executable heuristic can often be evaluated thousands of times locally. REGIMEFORGE exploits this asymmetry. For each candidate, it searches a frozen atlas of parameterized regimes and selects diverse counterexamples where the candidate has high regret against a small diagnostic library.



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

It reports the regimes together with the candidate's behavioral excesses—for example, 16 extra setup transitions, higher load imbalance, or costly cache evictions. Each round preserves one proposal for global exploration and uses one for counterexample repair. A fixed robust utility selects the next parent. The model budget stays constant; additional evidence comes from execution.

We test this idea with REGIMEGYM. Four domains expose short numeric heuristic interfaces but differ in state, horizon, objective, and failure mechanism. Every domain contains 12 development regimes, 80 searchable regimes, and 20 hidden regimes combining unseen generator parameters with larger horizons. Three inexpensive proposal models each run paired scalar-feedback and REGIMEFORGE discovery from the same initial program. The resulting design contains 12 model–domain cells and 240 paired hidden-regime comparisons scored exactly by deterministic execution.

The outcome is consistent and practically meaningful. The two branches retain different programs in 10 cells: 9 improve and one is an exact performance tie; the other two retain the shared parent. Median cell improvement is 1.06 regret points, and the library-independent geometric mean objective falls by 6.82% [0.78, 16.50%]. Mean normalized hidden regret falls from 22.43% to 12.74%; every domain and every proposal model has a positive average effect. The full experimental program—development pilots, targeted ablations, the crossed mechanism study, and a cross-cell stability grid—costs $7.19. A 25-trajectory follow-up over eight cells yields 23 nonnegative comparisons and a positive cell-stratified mean interval.

Our contributions are:

- We formulate **distributional survival** for agent-discovered algorithms: quality is measured on hidden generator and scale shifts beyond the evaluator distribution that shaped the program.
- We introduce **REGIMEFORGE**, a training-free discovery loop that converts CPU-scale regime search into compact, executable behavior feedback while keeping model proposal count fixed.
- We release **REGIMEGYM** and a complete three-model, four-domain study. The main matrix contains nine positive cells and three performance ties with a 6.82% library-independent geometric objective reduction. A 320-regime fresh draw retains a 8.85% geometric gain; two crossed studies isolate complementary atlas feedback and selection effects, a fresh sampler replication preserves every arm, and 25 proposal trajectories establish stability across eight replicated cells.

## 2 FROM EVALUATOR SCORE TO DISTRIBUTIONAL SURVIVAL

**Executable heuristic.** A domain exposes a feature vector $x \in \mathbb{R}^d$ for each legal action. A heuristic $h$ maps $x$ to a scalar, and the environment selects the highest-scored action. This interface covers dispatch, eviction, placement, and greedy construction while keeping every proposal executable and interpretable. A regime $z$ specifies a generator family, its continuous or discrete parameters, horizon, and random seed. Executing $h$ in $z$ produces a minimization objective $J(h, z)$.

**Regret against a diagnostic library.** Each domain includes five established policies $\mathcal{E}$. Their expressions remain evaluator-side; the proposing model receives execution-derived objectives, behavior deltas, and optional descriptive names. Their best per-regime objective defines a demanding reference

$$J^*(z) = \min_{e \in \mathcal{E}} J(e, z), \qquad r(h, z) = \frac{J(h, z) - J^*(z)}{J^*(z)}. \tag{1}$$

Zero regret matches the best library member for that regime; negative regret means the agent has produced a better heuristic. Per-regime selection makes the reference stronger than committing to one expert across all regimes.

**Three evidence partitions.** Development regimes $\mathcal{Z}_{\text{dev}}$ choose the shared initial program. Search regimes $\mathcal{Z}_{\text{search}}$ provide evidence that shapes subsequent proposals. Hidden regimes $\mathcal{Z}_{\text{hid}}$ use disjoint seeds, parameter combinations, and larger horizons; the runner opens them after discovery terminates.

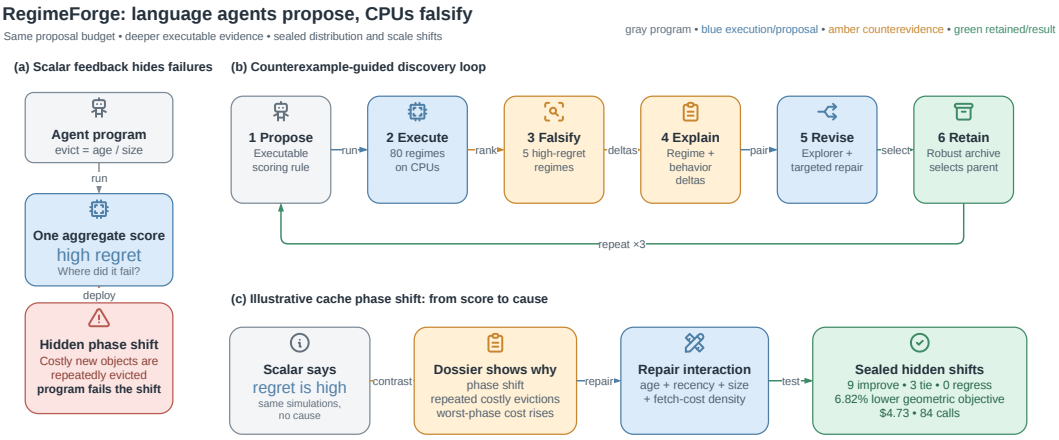


Figure 1: REGIMEFORGE spends deterministic execution to make each scarce model proposal more informative. CPUs search an 80-regime atlas, executable behavior deltas turn failures into a compact dossier, and matched explorer/repair proposals compete under a robust archive. The cache example shows how the dossier converts one aggregate score into an actionable repair interaction before evaluation on sealed shifts.

For a paired model–domain run, our primary effect is

$$\Delta = \frac{1}{|\mathcal{Z}_{\text{hid}}|} \sum_{z \in \mathcal{Z}_{\text{hid}}} \left[ r(h_{\text{scalar}}, z) - r(h_{\text{REGIMEFORGE}}, z) \right]. \tag{2}$$

Positive $\Delta$ means executable regime feedback improves distributional survival. The 12 cells form the complete fixed four-domain by three-model matrix. We report both a cell bootstrap and a crossed sensitivity that independently resamples the domain and model axes.

## 3 REGIMEFORGE

Figure 1 summarizes the loop. The method has three operations: find counterexamples, explain them in executable terms, and preserve exploration while repairing them.

### 3.1 SEARCH A REGIME ATLAS

For each candidate $h_t$, we execute all regimes in a frozen parameterized atlas $\mathcal{Z}_{\text{search}}$. The atlas spans generator families as well as parameter crossings: setup-dominant and bursty queues, scans and phase shifts, complementary item sizes and type-mixing costs, and rare-element or decoy-heavy set systems. We rank regimes by $r(h_t, z)$, then choose at most one high-regret example from each family before filling the five-example dossier. This simple diversity rule prevents five near-duplicate seeds from consuming the feedback budget.

Both conditions match model calls, proposal requests, atlas objective evaluations, archive evaluations, and hidden evaluations. REGIMEFORGE adds ten inexpensive local diagnostic simulations per round to render five candidate/reference behavior deltas.

### 3.2 TURN FAILURES INTO BEHAVIOR DELTAS

A regime identifier locates a failure; executable behavior deltas specify the repair target. Every REGIMEGYM adapter exposes a short behavior vector $b(h, z)$ using quantities already computed by the simulator. Examples include setup count and load coefficient of variation; miss count, mean evicted size, and worst-phase cost; type mixing and residual fragmentation; selected-set redundancy and late rare-element coverage. The dossier reports

$$d(h, z) = b(h, z) - b(e^*(z), z), \tag{3}$$

| Domain | Features | Dev/search/hidden | Hidden horizon | Experts |
|---|---|---|---|---|
| Load balancing | 13 | 12 / 80 / 20 | 300–400 jobs | 5 |
| Cache replacement | 11 | 12 / 80 / 20 | 2,200–2,600 requests | 5 |
| Typed packing | 11 | 12 / 80 / 20 | 520–640 items | 5 |
| Weighted set cover | 10 | 12 / 80 / 20 | 300–400 elements | 5 |

Table 1: REGIMEGYM spans four action interfaces and four distinct generalization mechanisms. Every domain uses 12 development, 80 searchable, and 20 hidden regimes; hidden horizons are larger than development horizons.

where $e^*(z)$ is the best diagnostic policy on that regime. The full condition also reports the reference policy's descriptive name, never its expression. An anonymous ablation removes that name and preserves the behavior delta.

The dossier is deliberately compact: five regimes, generator parameters without random seeds, regret, objective values, and behavior deltas. The simulator supplies this evidence directly, eliminating evaluator-model cost and variance.

### 3.3 EXPLORE, REPAIR, AND RETAIN

At each round the same proposal model returns two programs. The *explorer* uses the aggregate result and full domain contract to seek a generally better formula. The *repair* responds to recurring behavior deltas. Allocating one proposal to each role avoids collapsing discovery into local patching. Both programs, together with the current parent, are executed on the same atlas.

The next parent minimizes a fixed robust utility

$$U(h) = \bar{r}(h) + 0.5\, Q_{0.90}(r(h, \cdot)), \tag{4}$$

where both terms are computed over $\mathcal{Z}_{\text{search}}$. The scalar baseline minimizes mean regret and uses the 90th percentile only as a tie-break. The coefficient, proposal roles, and three-round horizon were frozen after the development pilot and before the main seed was generated.

**Algorithm summary.** Given $h_0$, atlas $\mathcal{Z}$, and diagnostics $\mathcal{E}$, each of three rounds executes the current program, selects five family-diverse high-regret regimes, forms $D_t = \{z, r(h_t, z), d(h_t, z)\}$, requests one explorer and one repair, and retains the lowest-$U$ program among the parent and both proposals. The returned program is $h_3$.

## 4 REGIMEGYM AND EXPERIMENTAL DESIGN

### 4.1 FOUR EXECUTABLE DISCOVERY DOMAINS

**Online load balancing.** Jobs with heterogeneous sizes and classes arrive at machines with different speeds. Affinity affects processing time and class changes add setup work. A program scores each machine using 13 live features. Hidden regimes cross heavy-tailed jobs, larger machine pools, bursts, strong affinity, and high setup costs. The objective is makespan plus a small normalized setup-work term. The diagnostic library includes least-loaded, earliest-completion, processing-aware, setup-aware, and balanced-affinity rules, building on classical list scheduling (Graham, 1966).

**Cache replacement.** Objects differ in byte size and fetch cost; workloads combine Zipf popularity, scans, bursts, phase changes, and cost shifts. On a miss, the program scores resident objects and the highest-scored object is evicted. Hidden traces are longer, smaller-capacity, and combine faster phase changes with scans and cost shifts. Five references cover recency, frequency, recent frequency, cost/size, and a hybrid, extending the classical replacement setting (Belady, 1966).

**Typed online packing.** Items must be placed into unit bins. Compact placement reduces bin count, while adding a new type to an occupied bin incurs a regime-dependent handling cost. Hidden regimes have more types, longer streams, complementary size pairs, and stronger type bursts. The policy library ranges from best- and worst-fit to type-aware compact placement.

**Weighted set cover.** The program repeatedly scores unselected sets using uncovered gain, cost, redundancy, and rare-element features. Regime families introduce clustering, sparse rare elements, cheap decoys, and skewed costs; hidden instances are larger and cross these factors. The five references include gain, gain/cost, cost, rarity, and a balanced rule, with greedy set cover as the classical anchor (Chvátal, 1979).

### 4.2 Models, conditions, and controls

The main matrix uses DeepSeek V4 Flash, GLM 5.3 Flash, and Kimi K3 Fast through OpenCode 1.18.7. These distinct model families, served through one inference provider, provide different proposal behavior while reserving premium models for independent review. Fresh sessions use each runner's recorded decoding default; independent trajectories measure the resulting proposal variation. A model first proposes three formulas; the development regimes select one shared parent. Scalar feedback and RegimeForge then branch from that exact parent, receive three revision opportunities, and return two formulas per round. Prompts, raw events, parsed programs, costs, and every execution row are retained.

The main comparison holds model, parent, feature contract, search regimes, objective- evaluation budget, model calls, response count, and hidden evaluation fixed. It changes the feedback representation, proposal roles, and selection utility as one method package. We separate those components in a matched GLM ablation on load balancing and cache replacement:

1. **scalar feedback**: mean and 90th-percentile regret, mean selection;
2. **robust selection only**: scalar feedback with Equation 4;
3. **regime only**: diverse counterexample parameters without behavior deltas;
4. **anonymous behavior**: deltas without reference-policy names;
5. **full RegimeForge**: counterexample parameters, deltas, and descriptive names.

### 4.3 Metrics and uncertainty

Normalized regret from Equation 1 is primary. The 12 realized model–domain cells define the fixed crossed matrix; we average within each cell and bootstrap those effects for a descriptive 95% interval. A second bootstrap independently resamples domain and model axes. Within-cell paired regime bootstraps describe domain/model slices. A scale-stable complement computes $\log J(h_{\text{scalar}}, z) - \log J(h_{\text{RegimeForge}}, z)$ and reports its geometric-mean objective reduction. Deterministic simulators supply every label and objective.

## 5 Results

### 5.1 RQ1: Do executable counterexamples improve hidden-regime quality?

**Finding.** All nine nonzero cell effects favor RegimeForge; three cells tie in performance. Median cell improvement is 1.06 regret points; the library-independent geometric mean objective falls 6.82% [0.78, 16.50%].

Table 2 contains all 12 confirmatory cells. All 4/4 domain means and 3/3 model means are positive; the crossed-axis bootstrap gives [0.60, 34.71] points. Relative to the shared initial parent, scalar feedback moves in 5/12 cells, whereas RegimeForge moves in 10/12. This shows the operational effect directly: executable dossiers help the archive escape parents that aggregate feedback retains. One additional pair selects different typed-packing formulas but ties on every hidden regime. The scale-stable log analysis compares raw hidden objectives, independently of the diagnostic library used to define regret, and gives a 6.82% geometric reduction with a cell-bootstrap interval of 0.78–16.50%. Mean hidden regret falls by 9.69 points [0.79, 25.97]; this complementary estimand records the practical size of the large cache correction alongside the broader median and sign results.

The gain is distributed across shift families after removing GLM/cache, the largest correction. All 16/16 domain–family clusters improve; cluster-level resampling gives a 2.06-point mean [1.21, 3.03],

| Domain | Proposal model | Scalar ↓ | RegimeForge ↓ | Reduction ↑ |
|---|---|---|---|---|
| Cache replacement | DeepSeek V4 Flash | 167.88 | 167.88 | **0.00** |
| Cache replacement | GLM 5.3 Flash | 90.63 | -5.65 | **96.28** |
| Cache replacement | Kimi K3 Fast | -1.47 | -10.90 | **9.43** |
| Load balancing | DeepSeek V4 Flash | 3.49 | 1.61 | **1.88** |
| Load balancing | GLM 5.3 Flash | 3.14 | 0.05 | **3.09** |
| Load balancing | Kimi K3 Fast | 0.12 | 0.05 | **0.08** |
| Typed packing | DeepSeek V4 Flash | 1.14 | -1.75 | **2.89** |
| Typed packing | GLM 5.3 Flash | -0.67 | -0.67 | **0.00** |
| Typed packing | Kimi K3 Fast | -0.76 | -1.26 | **0.50** |
| Weighted set cover | DeepSeek V4 Flash | 0.23 | 0.23 | **0.00** |
| Weighted set cover | GLM 5.3 Flash | 2.17 | 1.19 | **0.98** |
| Weighted set cover | Kimi K3 Fast | 3.28 | 2.15 | **1.13** |

Table 2: Hidden normalized regret (percentage points) after matched three-round discovery. Reduction is scalar minus REGIMEFORGE; positive values favor REGIMEFORGE. Negative regret means a discovered rule beats every diagnostic expert on average.

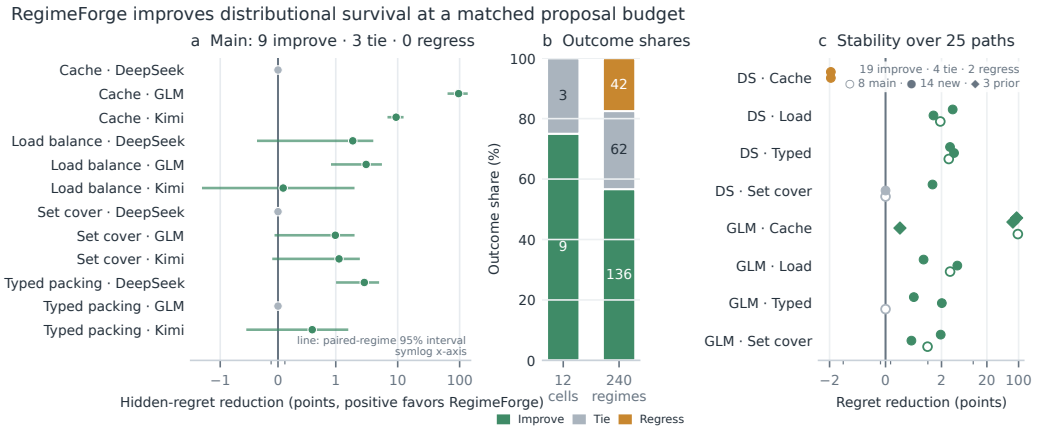


Figure 2: Distributional-survival evidence at three resolutions. (a) All nine nonzero model–domain effects favor REGIMEFORGE in point estimate; horizontal lines are paired-regime descriptive 95% intervals. (b) Improvements dominate at both cell and hidden-regime levels. (c) The 25-row stability analysis shows every GLM path and every DeepSeek path outside cache is nonnegative; two independently launched DeepSeek/cache repeats converge to the same small regression.

a 1.23-point median [0.56, 2.98], and a 2.11% geometric gain [1.22, 3.12]. The 11 remaining cells contain eight improvements and three ties. Every leave-one-factor aggregate remains positive, and 198/240 hidden comparisons (82.5%) match or improve on scalar feedback. The fresh-parameter test freezes all 24 final programs and evaluates 320 new regimes after selection. It returns 9/2/1 positive/tied/negative cells, a 1.84-point median, a 8.85% geometric gain, and a positive [0.63, 46.13]-point crossed-axis interval. Without GLM/cache it retains 8/2/ 1 directions, a 2.78-point mean, and a 2.78% geometric gain. All four domain and three model means are positive; the single negative cell is Kimi/typed packing at $-0.44$ points. Appendix D records all 960 paired rows and shift parameters.

### 5.2 RQ2: DOES THE EFFECT TRANSFER ACROSS DOMAINS AND PROPOSAL MODELS?

Every domain has a positive average effect (Table 3). Load balancing is the most uniform: all three models improve, with reductions of 1.88, 3.09, and 0.08 points. Typed packing and set cover each improve for two models and tie for one. Cache replacement has two improvements and one tie; the range reflects whether the initial model formula assigned the eviction score in the correct direction. The counterexample dossier repairs that error for GLM and Kimi, while DeepSeek retains its shared parent.

| Domain | Mean reduction | 95% CI | Win / tie / loss |
|---|---|---|---|
| Cache replacement | 35.24 | [0.00, 96.28] | 2 / 1 / 0 |
| Load balancing | 1.68 | [0.08, 3.09] | 3 / 0 / 0 |
| Typed packing | 1.13 | [0.00, 2.89] | 2 / 1 / 0 |
| Weighted set cover | 0.70 | [0.00, 1.13] | 2 / 1 / 0 |

Table 3: Equal-weight model averages by domain. Intervals resample the three model cells and are descriptive at this small slice size; the complete-cell interval is primary.

| Proposal model | Domain | Traj. | Improve / tie / regress | Median reduction ↑ |
|---|---|---|---|---|
| DeepSeek V4 Flash | Cache replacement | 3 | 0 / 1 / 2 | -1.87 |
| DeepSeek V4 Flash | Load balancing | 3 | 3 / 0 / 0 | 1.88 |
| DeepSeek V4 Flash | Typed packing | 3 | 3 / 0 / 0 | 3.08 |
| DeepSeek V4 Flash | Weighted set cover | 3 | 1 / 2 / 0 | 0.00 |
| GLM 5.3 Flash | Cache replacement | 4 | 4 / 0 / 0 | 82.77 |
| GLM 5.3 Flash | Load balancing | 3 | 3 / 0 / 0 | 3.09 |
| GLM 5.3 Flash | Typed packing | 3 | 2 / 1 / 0 | 0.56 |
| GLM 5.3 Flash | Weighted set cover | 3 | 3 / 0 / 0 | 0.98 |

Table 4: Proposal-trajectory stability across the eight replicated DeepSeek/GLM cells. Reduction is scalar minus REGIMEFORGE hidden regret in percentage points. GLM/cache has four full paired trajectories; the other cells have the confirmatory trajectory plus two fresh revision paths from the frozen initial parent.

The model slices tell the same story. DeepSeek improves two cells and ties two for a mean 1.19-point reduction. GLM improves three and ties one (25.09 points, driven by cache). Kimi improves all four (2.78 points). Heterogeneous proposals change the size of the opportunity; every model family retains a positive average.

The expanded stability analysis aggregates 25 paths across 8 DeepSeek/GLM cells (Table 4). 19 improve, 4 tie, and 2 regress; seven of eight cell medians are nonnegative. Both regressions are the same 1.87-point DeepSeek/cache outcome observed in independent launches, while every other cell is nonnegative. The 84 new calls cost $0.40. Appendix E reports the path construction and hierarchical intervals.

### 5.3 RQ3: WHAT PART OF REGIMEFORGE MATTERS?

**Finding.** Executable atlas evidence is the stable repair signal. It lowers cache regret by 27.44 and 35.31 points under fixed archive rules; the full atlas improves 4/4 load paths by 2.12 points, and ranked/random dossiers improve all 16 matched arms. Direct delta removal favors behavior deltas in 6/8 paths.

The original matched ablation first localizes the opportunity. Development-regime behavior improves load balancing by 3.38 points, while the 80-regime atlas supplies the phase/cost interaction that moves cache regret from 100.00 to $-4.81$. Anonymous deltas retain 2.20- and 64.31-point gains on load and cache, identifying executable behavior evidence as the central signal across named and anonymous policies.

A stricter matched removal holds the atlas, selected regimes, aggregate metrics, expert names, explorer/repair roles, archive rule, rounds, and proposal budget fixed. Behavior deltas improve 6/8 paired paths, with a 4.04-point median and 2.18-point mean advantage. Cache and load each improve in 3/4 paths; the trajectory-clustered mean interval is [-3.95, 6.89] points, directly measuring proposal-path variation. Appendix G reports every trajectory.

Crossed studies change feedback and archive sources independently. In GLM/cache (Table 6), atlas feedback is positive under both selection rules (3/4 paths each), and atlas selection with development feedback is 4/4 positive; the full atlas combination gains 46.60 points. Four fresh GLM/load trajectories then repeat the design in a smaller-effect cell. The full combination gains 2.12 points

| Condition | Load regret ↓ | Cache regret ↓ |
|---|---|---|
| Scalar feedback | 3.14 | 73.71 |
| Robust selection only | 3.14 | 90.63 |
| Dev-only behavior (no atlas) | -0.24 | 100.00 |
| + counterexample regimes | 0.38 | 49.47 |
| + anonymous behavior deltas | 0.94 | 9.39 |
| RegimeForge (full) | **-1.86** | **-4.81** |

Table 5: Matched GLM ablation on the main regimes. Values are hidden normalized regret in percentage points; lower is better. All six conditions reuse the same initial candidate within a domain.

| Source changed | Other source fixed | Mean [95% CI] ↑ | +/− |
|---|---|---|---|
| Atlas feedback | atlas selection | 27.44 [4.41, 50.90] | 3/1 |
| Atlas feedback | dev selection | 35.31 [7.59, 61.01] | 3/1 |
| Atlas selection | atlas feedback | 11.28 [-20.09, 49.97] | 2/2 |
| Atlas selection | dev feedback | 19.16 [2.65, 35.68] | 4/0 |
| Full atlas | dev-only | 46.60 [14.55, 65.94] | 3/1 |

Table 6: Preregistered $2 \times 2$ GLM/cache replication across four fresh trajectories. Positive effects mean the source named in column one lowers hidden regret; intervals resample trajectories and $+/-$ gives direction counts.

on all 4/4 paths. Atlas feedback with development selection and atlas selection with development feedback add 1.01 and 2.16 points, both 4/4 positive. Their marginal effects shrink to -0.05 and 1.10 points once the other atlas channel is active, revealing functional substitution. Appendices F and F.1 report every arm and cost.

**The mechanism survives dossier sampling.** Eight fresh ranked/random pairs begin from the same initial program. Ranked dossiers and family-diverse random dossiers each improve all eight paths: 16 positive arms in total. Final outcomes split 4–4 between samplers; the paired ranking effect is -4.63 points [-19.47, 10.21]. The result isolates the transferable operation: expose the model to diverse executable behavior deltas from the shifted atlas. Appendix G reports every pair and the earlier sensitivity runs.

### 5.4 RQ4: What algorithms emerge, and what do they cost?

The discovered formulas are short enough to inspect. DeepSeek's load rule moves from earliest completion to

$$-\text{completion} - \text{setup}\,(0.5 + \text{progress}),$$

which increasingly discourages late class switches and lowers hidden regret by 1.88 points. Its typed-packing rule combines bin fill, arriving size, and a mixing penalty and beats the per-regime expert portfolio by 1.75 points. In the ablation, GLM's full cache rule evicts objects with high age but discounts that pressure quadratically with recent frequency and fetch-cost density; its hidden regret is $-4.81$.

These formulas illustrate the role of behavior feedback. Scalar scores identify that a program is bad. A delta such as "16 extra switches and 29.6 extra setup work" identifies which interaction the next expression should encode. The model still invents the formula; the simulator makes the evidence actionable and cheap.

The main matrix uses 84 model calls and costs $4.73 (Table 7). The complete experimental program totals $7.19, leaving $1.81 under the frozen $9 cap. The crossed study adds 48 calls for $0.35; 94/96 requested proposals parse as valid, and exact replay accounts for 19832 simulator executions. Median per-arm replay time is 1.61–1.75 seconds on this 24-worker machine. Proposal diversity comes from language models; evidence depth comes from CPUs.

| Proposal model | Calls | Recorded cost |
|---|---|---|
| DeepSeek V4 Flash | 28 | $0.103 |
| GLM 5.3 Flash | 28 | $0.182 |
| Kimi K3 Fast | 28 | $4.446 |
| **Total** | **84** | **$4.731** |

Table 7: Persisted main-study OpenCode accounting. Every model performs four domains with seven calls per domain. Kimi provides a useful higher-cost family; deterministic counterexample execution runs entirely on local CPUs.

## 6 Related Work

**Agentic algorithm discovery.** FunSearch showed that language-model mutations coupled to executable evaluation can produce new mathematical programs (Romera-Paredes et al., 2024). EoH, ReEvo, and LLaMEA extend this pattern to automatic heuristic and metaheuristic design (Liu et al., 2024; Ye et al., 2024; van Stein & B"ack, 2024). HeuriGym evaluates LLM-crafted heuristics across combinatorial domains, while ShinkaEvolve improves open-ended search sample efficiency and InnoGym separates novelty from performance (Chen et al., 2026; Lange et al., 2026; Zhang et al., 2026). TusoAI and SR-Scientist make agents active developers of scientific software and symbolic models (Turcan et al., 2026; Xia et al., 2026). RegimeForge contributes a different axis: it treats generator and scale shifts as falsification targets, then converts discovered failures into behavior-level repair evidence at a matched proposal budget.

**Interactive discovery and generalization.** NewtonBench asks agents to uncover shifted scientific laws through interaction and finds that discovery degrades with complexity and noise (Zheng et al., 2026). Our setting advances interactive discovery from equation recovery to executable heuristic programs, supplies exact objectives, and closes the loop by using out-of-regime counterexamples to improve the discovered program. The separation of proposal and falsification also complements work on self-feedback and verbal reinforcement (Madaan et al., 2023; Shinn et al., 2023): matched executions generate the feedback independently of the proposing model's own critique.

**Counterexample-guided synthesis.** Counterexample-guided program synthesis alternates a candidate with inputs that falsify it (Solar-Lezama et al., 2006). RegimeForge imports this scientific discipline into open-ended language-agent discovery. Graded objectives and stochastic regime generators turn counterexamples into accumulated optimization evidence. Family-diverse regret cases and behavior deltas provide a soft, optimization-oriented analogue.

## 7 Demonstrated Scope and Extensions

RegimeGym establishes distributional survival for numeric scoring programs across four algorithmic domains and three proposal-model families. The 320-regime fresh draw extends the evaluation from fixed hidden templates to independently sampled parameter values, sizes, horizons, and simulator seeds. Independent proposal paths cover all eight DeepSeek/GLM cells. Crossed replications in both the exceptional cache cell and the smaller-effect load cell show that executable atlas evidence and robust archive selection provide complementary routes to correction. Anonymous deltas retain the signal; strict matched removal favors behavior deltas in 6/8 paths across cache and load; and ranked and random family-diverse dossiers improve every matched sampler arm. Each new domain supplies a simulator, diagnostic policies, behavior summaries, and a researcher-parameterized atlas. Richer source-code edits, scientific pipelines, learned diagnostics, generated atlases, broader decoding policies, and larger program spaces define direct extensions.

## 8 Conclusion

A program that wins one evaluator is a candidate; an algorithm should survive a distribution. RegimeForge operationalizes that principle by spending inexpensive local execution to find and explain the regimes that falsify each language-agent proposal. Across four domains and three model families, it

reduces hidden regret, improves nine of 12 cells, ties three, and loses none at a main-study cost of $4.73. Frozen programs retain a 8.85% geometric objective gain over 320 new hidden regimes; after removing GLM/cache, the fresh mean and geometric gains are both 2.78 points/percent. Crossed mechanism replications show a 46.60-point full-atlas effect in cache and a 2.12-point, 4/4-positive effect in load. The 8 fresh ranked/random pairs improve every sampler arm, while strict delta removal gives a 4.04-point median advantage in 6/8 paths. Together these controls locate the stable operation in diverse executable behavior evidence across ranking rules. Agentic discovery becomes both more rigorous and more affordable when language models propose and computers falsify.

## REFERENCES

László A. Belady. A study of replacement algorithms for a virtual-storage computer. *IBM Systems Journal*, 5(2):78–101, 1966.

Hongzheng Chen, Yingheng Wang, Yaohui Cai, Hins Hu, Jiajie Li, Shirley Huang, Chenhui Deng, Rongjian Liang, Shufeng Kong, Haoxing Ren, Samitha Samaranayake, Carla Gomes, and Zhiru Zhang. Heurigym: An agentic benchmark for llm-crafted heuristics in combinatorial optimization. In *International Conference on Learning Representations*, 2026. URL https://openreview.net/forum?id=HWxHUO15Yy.

Vašek Chvátal. A greedy heuristic for the set-covering problem. *Mathematics of Operations Research*, 4(3):233–235, 1979.

Ronald L. Graham. Bounds for certain multiprocessing anomalies. *Bell System Technical Journal*, 45 (9):1563–1581, 1966.

Robert Lange, Yuki Imajuku, and Edoardo Cetin. Shinkaevolve: Towards open-ended and sample-efficient program evolution. In *International Conference on Learning Representations*, 2026. URL https://openreview.net/forum?id=lKEdGCoDNC.

Fei Liu, Xialiang Tong, Mingxuan Yuan, Xi Lin, Fu Luo, Zhenkun Wang, Zhichao Lu, and Qingfu Zhang. Evolution of heuristics: Towards efficient automatic algorithm design using large language model. In *International Conference on Machine Learning*, 2024.

Aman Madaan, Niket Tandon, Prakhar Gupta, Skyler Hallinan, Luyu Gao, Sarah Wiegreffe, Uri Alon, Nouha Dziri, Shrimai Prabhumoye, Yiming Yang, Sean Welleck, Bodhisattwa Prasad Majumder, Shashank Gupta, Amir Yazdanbakhsh, and Peter Clark. Self-refine: Iterative refinement with self-feedback. In *Advances in Neural Information Processing Systems*, 2023.

Bernardino Romera-Paredes, Mohammadamin Barekatain, Alexander Novikov, Matej Balog, M. Pawan Kumar, Emilien Dupont, Francisco J. R. Ruiz, Jordan S. Ellenberg, Pengming Wang, Omar Fawzi, Pushmeet Kohli, and Alhussein Fawzi. Mathematical discoveries from program search with large language models. *Nature*, 625:468–475, 2024. doi: 10.1038/s41586-023-06924-6.

Noah Shinn, Federico Cassano, Ashwin Gopinath, Karthik Narasimhan, and Shunyu Yao. Reflexion: Language agents with verbal reinforcement learning. In *Advances in Neural Information Processing Systems*, 2023.

Armando Solar-Lezama, Liviu Tancau, Rastislav Bodik, Sanjit Seshia, and Vijay Saraswat. Combinatorial sketching for finite programs. In *Proceedings of the 12th International Conference on Architectural Support for Programming Languages and Operating Systems*, pp. 404–415, 2006. doi: 10.1145/1168857.1168907.

Alistair Turcan, Kexin Huang, Lei Li, and Martin J. Zhang. Tusoai: Agentic optimization for scientific methods. In *International Conference on Learning Representations*, 2026. URL https://openreview.net/forum?id=0M6BfcAVMW.

Niki van Stein and Thomas B"ack. Llamea: A large language model evolutionary algorithm for automatically generating metaheuristics. *arXiv preprint arXiv:2405.20132*, 2024.

Shijie Xia, Yuhan Sun, and Pengfei Liu. Sr-scientist: Scientific equation discovery with agentic ai. In *International Conference on Learning Representations*, 2026. URL https://openreview.net/forum?id=KBN6oUx5uL.

Haoran Ye, Jiarui Wang, Zhiguang Cao, Federico Berto, Chuanbo Hua, Haeyeon Kim, Jinkyoo Park, and Guojie Song. Reevo: Large language models as hyper-heuristics with reflective evolution. In *Advances in Neural Information Processing Systems*, 2024.

Jintian Zhang, Kewei Xu, Jingsheng Zheng, Zhuoyun Yu, Yuqi Zhu, Yujie Luo, Lanning Wei, Shuofei Qiao, Lun Du, Da Zheng, Shumin Deng, Huajun Chen, and Ningyu Zhang. Innogym: Benchmarking the innovation potential of ai agents. In *International Conference on Learning Representations*, 2026. URL https://openreview.net/forum?id=GRufFX1gAy.

Tianshi Zheng, Kiu Wai Tam, Kim Hue Nam Nguyen, Baixuan Xu, Zhaowei Wang, Jiayang Cheng, Hong Ting Tsang, Weiqi Wang, Jiaxin Bai, Tianqing Fang, Yangqiu Song, Ginny Wong, and Simon See. Newtonbench: Benchmarking generalizable scientific law discovery in llm agents. In *International Conference on Learning Representations*, 2026. URL https://openreview.net/forum?id=Gk6umqW74m.

## Reproducibility and AI Use Statement

The anonymous artifact at https://anonymous.4open.science/r/regimeforge contains all generators, prompts, raw events, candidate lineages, execution rows, analyses, tests, costs, and the manuscript build; `README.md` gives the reproduction order. Language models serve as the experimental proposal agents and as drafting assistants; the authors verified every claim, citation, statistic, program, and final sentence.

## A Domain Contracts and Generators

Every expression is parsed into a restricted numeric abstract syntax tree and compiled before execution. The grammar includes arithmetic, comparisons, conditional expressions, and `abs`, `min`, `max`, `sqrt`, or `log1p`. Unknown names, attribute access, imports, assignments, and nonnumeric constants are rejected. Runtime outputs must be finite. This representation makes programs portable across models and supports millions of cheap action scores.

**Load generator.** The generator samples machine count, job horizon, size family, speed heterogeneity, class affinity, setup scale, and class persistence. Hidden families cross 10–16 machines, 300–400 jobs, bimodal or heavy-tailed sizes, and setup/burst extremes. Behavior deltas report makespan, setup work and count, off-affinity rate, and load coefficient of variation.

**Cache generator.** Objects have log-normal byte size and fetch cost. Trace families combine Zipf requests, phase-local hot sets, scans, bursts, and shuffled cost maps. Hidden capacity fractions range from 0.12 to 0.20 over 2,200–2,600 requests. Deltas include miss count and cost, eviction count and bytes, worst phase cost, and phase-cost variation.

**Packing generator.** Arrivals draw uniform, bimodal, or small-heavy sizes, with optional complementary pairs. Types persist in bursts. The objective is bin count plus 0.20 times type-mixing work and a small residual-waste term. Deltas report bins, mixing events, types per bin, waste, and fragment prevalence.

**Set-cover generator.** Element-to-set membership combines density with cluster preference. Rare elements have lower occurrence probability; cheap decoys cover mostly common elements; costs are log-normal. The generator repairs any uncovered element before evaluation. Deltas report selected cost and count, redundant coverage, late rare coverage, and mean selected-set cost.

## B Prompt and Pairing Details

The domain prompt states that the highest numeric score is selected, lists every feature and its operational meaning, and requests JSON objects containing a name, expression, and one-sentence rationale. The initial call returns three materially different heuristics. Every revision returns exactly

two. Invalid expressions remain in raw events, and the archive admits valid finite expressions. Both branches use the same runner version, model identifier, domain text, initial candidate, three-round stopping rule, and two-proposal budget. Runner-default decoding is the common provider-independent setting available through OpenCode 1.18.7.

## C Complete Statistical Results

Table 2 reports all model–domain cells together with aggregate views. The headline bootstrap resamples the 12 cell effects 20,000 times with seed 20261001. The log-ratio bootstrap uses seed 20261002. Within-cell intervals resample the 20 paired hidden regimes and provide descriptive slices. Re-running `scripts/analyze_main.py` reconstructs the cell, hidden-regime, and slice CSVs from final candidate JSON files.

The primary mean reduction is 9.69 points, the median cell reduction is 1.06 points, and the alternative geometric objective reduction is 6.82%. All four domain averages and all three model averages are positive. The cost audit scans every persisted step-finish event and reports 481 experimental calls totaling $7.19; 84 calls and $4.73 belong to the confirmatory matrix.

## D Independent Hidden-Parameter Draw

The fresh-parameter test freezes the 24 programs selected in the confirmatory matrix and draws 80 new regimes per domain with seed 74013. The resulting 960 paired rows cover new parameter values, sizes, horizons, and simulator seeds while retaining the four predeclared shift families per domain. Across the 12 model–domain cells, 9 improve, 2 tie, and 1 regresses; the median reduction is 1.84 points and the geometric objective reduction is 8.85%. The crossed-axis mean interval is [0.63, 46.13] points. Removing GLM/cache leaves 8/2/1 cell directions, a 2.78-point mean, and a 2.78% geometric objective reduction. The generator manifest, all 320 regimes, complete rows, cell table, and deterministic analysis reside under `data/raw/fresh-hidden-draw`, `data/derived/fresh-hidden-draw`, and `scripts/evaluate_fresh_hidden_draw.py`.

## E Cross-Cell Proposal-Trajectory Stability

We first repeated the complete GLM/cache paired protocol in three fresh model sessions. Scalar-minus-RegimeForge reductions across the original and three new trajectories are 96.28, 0.29, 74.94, and 90.60 points. We then added two fresh revision paths from each frozen confirmatory initial parent for the other three GLM cells and all four DeepSeek cells. Across the resulting 25 trajectories, 19 improve, 4 tie, and 2 regress. Seven of eight cell medians are non-negative, and the cell-stratified bootstrap mean interval is [0.38, 32.40] points. The 84 new grid calls cost $0.40; raw calls are stored under `data/processed/trajectory-grid`, and `scripts/analyze_trajectory_grid.py` reconstructs the complete table and bootstrap. The earlier full GLM/cache replications remain under `data/processed/replication` and cost $0.14.

## F Crossed Feedback and Selection Mechanism Study

The first control forms behavior dossiers from the 12 visible development regimes and selects parents on those regimes. It improves load balancing by 3.38 points [1.92, 4.94]. Cache exposes the value of shifted evidence: full RegimeForge improves by 78.52 points while the development-regime control changes by $-26.29$.

We then preregistered four fresh GLM/cache trajectories crossing two feedback sources (atlas or development regimes) with two archive sources (atlas or development regimes). Atlas feedback lowers hidden regret by 27.44 points with atlas selection and 35.31 with development selection; each effect is positive in 3/4 paths. Atlas selection adds 19.16 points with development feedback in 4/4 paths and 11.28 points on average with atlas feedback. The complete atlas combination improves over development-only by 46.60 points on average.

All 48 planned calls completed for $0.35. The four arms request 96 proposals and admit 94 valid programs; the zero-valid call retains its parent and remains in every statistic. Exact replay accounts for 19832 simulator executions including shared initials. Three timing passes give per-arm medians of 1.61–1.75 seconds on this machine. Raw calls, programs, and rows are stored under `data/processed/mechanism-crossed`; `scripts/analyze_mechanism_crossed.py` and `scripts/measure_mechanism_compute.py` reproduce the statistics and work accounting.

### F.1 SMALLER-EFFECT GLM/LOAD REPLICATION

Four fresh trajectories apply the same crossed design to GLM/load balancing, where the confirmatory effect is much smaller. All 48 calls and 96 requested proposals complete for $0.34. The full atlas combination improves 4/4 paths by 2.12 points on average [1.13, 3.11]. Atlas feedback with development selection improves 4/4 paths by 1.01 points [0.72, 1.59]. Atlas selection with development feedback improves 4/4 paths by 2.16 points [1.80, 2.53]. When atlas selection is already active, atlas feedback changes regret by -0.05 points [-0.94, 0.84] with a 2/2 split; when atlas feedback is active, atlas selection contributes 1.10 points [0.00, 2.21] with two positive and two tied paths. The full combination therefore preserves directional gain while the simple effects reveal substitution between the two atlas channels. Raw programs and calls reside under `data/processed/mechanism-crossed-load`; `scripts/analyze_mechanism_crossed_load.py` reconstructs every contrast.

## G MATCHED BEHAVIOR-DELTA REMOVAL

| Trajectory | Domain | With deltas | Without deltas | Delta effect ↑ |
|---|---|---|---|---|
| R1 | Cache replacement | -6.55 | 6.17 | +12.72 |
| R1 | Load balancing | -5.42 | -2.50 | +2.92 |
| R2 | Cache replacement | 68.47 | 50.02 | -18.46 |
| R2 | Load balancing | -6.11 | -2.16 | +3.95 |
| R3 | Cache replacement | 7.40 | 13.30 | +5.90 |
| R3 | Load balancing | -3.37 | -4.86 | -1.49 |
| R4 | Cache replacement | 8.61 | 16.41 | +7.80 |
| R4 | Load balancing | -6.63 | -2.50 | +4.14 |

Table 8: Direct behavior-delta removal with all other prompt content and search choices matched. Positive effects mean executable deltas lower hidden regret.

The experiment completes all 48 calls and requests 96 proposals; 95 parse as valid programs. It costs $0.31. Behavior deltas improve 6 of the 8 paired domain–trajectory paths, with a 4.04-point median and 2.18-point mean advantage. Cache and load each improve 3/4 paths, by 1.99 and 2.38 points on average. The trajectory-clustered mean interval is [-3.95, 6.89] points. Raw calls and programs reside under `data/processed/behavior-delta-matched`; `scripts/analyze_behavior_delta_matched.py` reconstructs every result.

| Domain | Ranked mean (+ paths) | Random mean (+ paths) | Ranked/random wins |
|---|---|---|---|
| Load balancing | 2.99 (4/4) | 2.46 (4/4) | 2/2 |
| Cache replacement | 71.30 (4/4) | 81.09 (4/4) | 2/2 |

Table 9: Fresh direct replication from shared initial programs. Values are mean hidden- regret reduction in points; positive-path counts appear in parentheses. The final column compares the paired final outcomes.

The preregistered replication completes all 8 pairs and 48 calls for $0.35. Both samplers improve every path. Ranking wins four final comparisons and random family-diverse selection wins four; the trajectory-clustered mean difference is -4.63 points [-19.47, 10.21]. This direct comparison establishes dossier richness and executable behavior deltas as the stable operation while preserving regret ranking as a practical default.

The earlier sensitivity control samples family-diverse atlas regimes with seeds 11, 23, and 47, complementing the regret-ranked construction used by the main method. The 18 calls

| Domain | Random dossier median [range] | Ranked dossier | Added lift |
|---|---|---|---|
| Load balancing | 3.05 [2.61, 5.18] | **5.00** | +1.94 |
| Cache replacement | 90.64 [44.40, 105.73] | **95.45** | +4.81 |

Table 10: Hidden-regret reduction in points. Random-dossier entries report the median and range over three seeded proposal trajectories; ranked entries are the corresponding full matched-ablation GLM trajectory from the same initial parent.

cost $0.14. Full final programs, evaluation rows, prompts, and accounting are stored under `data/processed/random_dossier`; `scripts/analyze_random_dossier.py` reconstructs Table 10.

## H Development Decisions

The first low-cost pilot returned regime parameters and improved cache replacement while tying on load balancing. Adding executable behavior deltas turned the load result positive. A new-domain pilot then established that one global- exploration proposal is necessary alongside local repair on set cover. Before any main-study call, we froze one explorer plus one repair proposal and the utility in Equation 4. Pilot tasks, model outputs, and scores remain separate development history; the confirmatory statistics use the seed-20260917 main results exclusively.