MOSAIC: DYNAMIC BENCHMARKING AS MECHANISM DESIGN

Anonymous authors
Paper under double-blind review

ABSTRACT

A benchmark scores each model on its own and ranks the scores. Those scores now decide which models get deployed and which outputs train the next round, so the leaderboard also shapes what developers build. Standalone scoring pays the same for a capability whether every other model already has it or none does. Duplicating what already scores well is then each developer's best response, and collapse onto one mode is an equilibrium: a property of the score, not of the developers. We introduce MOSAIC (Mechanism for Output Scoring that Aligns Incentives with Coverage), which pays a valid response the log-determinant volume its features add to what the record already spans: responses that the declared feature map sends to one direction, a *mode*, count as equivalent, and each further one earns less. On a shared orthogonal menu, below an explicit quality–novelty threshold, MOSAIC's equilibria are exactly the no-repeat allocations of the highest-quality modes, all of them globally optimal, where standalone scoring's unique equilibrium puts every developer on the best mode; the threshold is worst-case tight. Quality plus coverage is an exact potential: a developer's gain from replacing its own entry equals the population's gain, so every maximal improvement path ends at an equilibrium. By submodularity every such equilibrium attains at least half the optimum, and below the threshold a globally optimal population. Keeping every past response in the comparison set makes coverage grow linearly in the horizon while fresh directions remain, against standalone scoring's $\Theta(\log T)$. In separate experiments MOSAIC raises pooled effective modes from 15 to 32 over twelve adaptation rounds, and the gain comes from keeping the record, whether each developer keeps its own or they share one. The same mechanism raises the share of held-out tasks some developer solves by 1.4 points at unchanged per-slot accuracy, and closes 15% of the baseline-to-SOTA gap on RSI-Exam's sealed grader.

1 INTRODUCTION

A benchmark is a fixed set of tasks and a rule that turns one model's answers into one number, and the ranked list of those numbers is a leaderboard. LiveCodeBench is a current instance: it scores models on competitive-programming problems and publishes the ordering (Jain et al., 2025). The task set, however, no longer stays fixed. Benchmarks now refresh their tasks against the models they rank, by collecting problems that defeat the current best (Nie et al., 2020; Kiela et al., 2021), by admitting only items published after a model's training cutoff (White et al., 2025; Jain et al., 2025), or by regenerating every item from a template on each run (Zhu et al., 2024; Mirzadeh et al., 2025). Their scores also decide which models get served, which agents keep a slot, and which outputs train the next round (Chiang et al., 2024). Such a leaderboard no longer only describes the set of models on it, its *population*; it decides membership, which is what happens to any measure that becomes a target (Strathern, 1997): developers converge on whatever it rewards (Hays et al., 2026; Chen et al., 2026b; Dominguez-Olmedo et al., 2025), and training on the survivors narrows the distribution again (Shumailov et al., 2024; Jiang et al., 2025), as separate studies measure for alignment tuning and model-assisted writing (Kirk et al., 2024; Padmakumar & He, 2024; Murthy et al., 2025).

Every one of those scores is standalone: it scores one model at a time, never the responses already on record. A model that solves a suite of problems is paid the same whether a hundred deployed models already solve them or nobody does. Duplicating a widespread capability is then every developer's

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.

best response, the individually highest-scoring choice given what the others are doing. The profile in which all of them duplicate is therefore an equilibrium: no developer gains by changing alone, even where each could have submitted a different capability. We call that profile *population collapse*. Economics has the same convergence as minimum differentiation (Hotelling, 1929), and algorithmic decision-making has it as monoculture, where agents deferring to one accurate signal leave the collective worse off than private noisier ones would (Kleinberg & Raghavan, 2021) and homogenize whom they pick (Bommasani et al., 2022); language models already narrow this way (Wu et al., 2025). Collapse here is a property of the score, not of the developers, and in particular it makes two frontier problems hard: a multi-agent system is worth its cost only when its agents cover each other's failures, and a continual learner trains on what it just accepted. What should a benchmark pay for, so that self-interested developers stay diverse?

We put the question on one task in Fig. 1: return the sum $1..n$. It admits three correct programs: an iterative loop, a recursion that repeats the loop's structure, and the closed-form identity. A standalone score orders the three by intrinsic quality alone, so whichever ranks first stays on top whatever the others submit. Against the record, however, their order depends on what the others submit: the recursion's feature vector aligns with the loop's and adds almost no volume, while the identity opens a new direction and earns a full term. We call a direction of the feature map a *mode*: the loop and the identity are two modes, the recursion lies near enough to the loop's direction to add almost nothing, and the same loop with renamed variables is the loop mode submitted twice.

Asked this way, benchmark design is mechanism design. If the evaluator wants the coverage a population reaches, the score has to pay a model for what it adds to that population and for nothing else. We introduce MOSAIC, a Mechanism for Output Scoring that Aligns Incentives with Coverage, which scores a model by the coverage it adds to the record. Each of N developers controls one entry, and the evaluator keeps every admitted response as a record with a declared feature vector ϕ_j and validity weight $w_j \geq 0$. We measure the coverage of a set S of records as a log-determinant volume $V(S) = \log\det(\mathbf{I} + \alpha \sum_{j \in S} w_j \phi_j \phi_j^\top)$, where $\alpha > 0$ fixes how fast a direction is discounted once other records cover it, and we pay a response the increment $\Delta_i(S) = V(S \cup \{i\}) - V(S)$ it brings. That volume is monotone and submodular (Nemhauser et al., 1978; Krause & Golovin, 2014), and paying a candidate for what it adds beyond a set is a familiar criterion (Section 7). What changes here is the set the increment is computed against: other developers' entries, not one system's own candidates. We fix three declarations in Section 2 that determine what the increment measures: the evaluation unit, the validity gate, and the feature map. The volume is the population objective, the quantity the evaluator wants the record to maximize, and the increment is what each entry is paid. We build it in that order: the objective (Section 2), the populations its increment makes stable (Section 3.1), how developers reach them (Section 3.2), the commit rule that keeps concurrent updates raising the objective (Section 3.3), and what happens once the record is kept (Section 4).

The populations stable under this score are the complementary ones. On orthogonal modes with strictly ordered qualities, and below a novelty premium the score itself fixes, MOSAIC's pure Nash equilibria are exactly the no-repeat allocations of the best modes, every one globally optimal, while standalone scoring's unique equilibrium is collapse onto the best mode; the threshold is tight (Theorem 1). MOSAIC scores incumbent and proposal against the same leave-one-out population, so a developer's gain from a unilateral change equals the change in quality plus coverage. That objective is an exact potential (Monderer & Shapley, 1996), so every improvement path halts at a pure Nash profile; submodularity puts each of those at 50% of the optimum or above (Vetta, 2002), and below the threshold at a globally optimal no-repeat allocation (Theorem 2 and Corollary 1). Concurrency breaks that identity and one rule restores it: we rescore each proposal against the population it would enter, a *fresh-state* commit. Under a *persistent* reference the population keeps diversifying, effective modes (Hill, 1973) and accrued coverage growing linearly in the horizon on every δ-rich, γ-rational path, two conditions we measure (Proposition 1), where a standalone score with a unique best action accrues only $\Theta(\log T)$.

Paying for novelty at the margin also invites a developer to present one capability as several directions; no feature map removes that incentive, and under a weight-conserving gate Proposition 6 caps the split's total payment below αw.

Contributions.

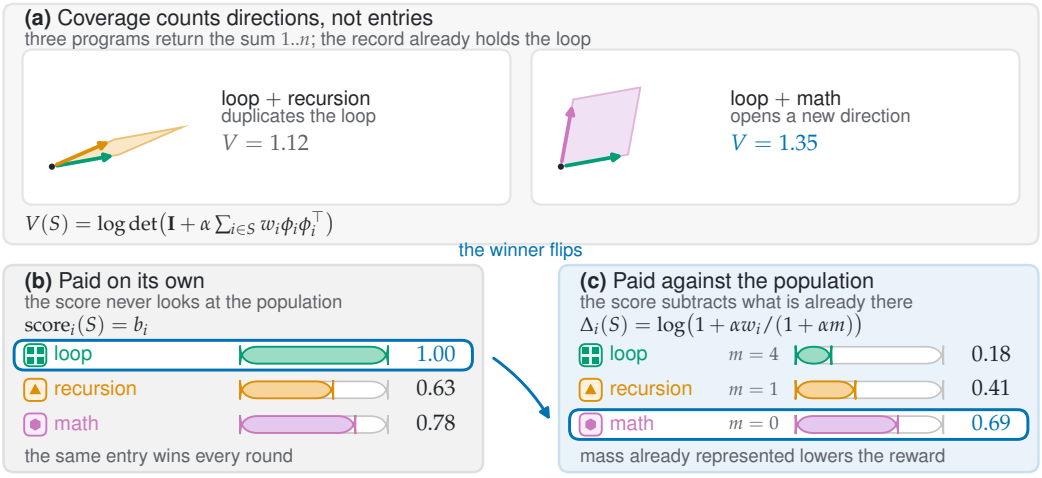


Figure 1: **Scoring against the population changes which output scores highest.** All panels use the same three strategies for *return the sum* $1..n$. *(a)* Coverage is a log determinant, so a near-copy of the loop adds little area and the complementary math direction more. *(b)* A standalone score pays intrinsic quality b_i, so the loop stays on top. *(c)* The MOSAIC score discounts the weight m already on that direction, so the uncovered math direction scores highest.

1. **Formulation.** We cast dynamic benchmark design as mechanism design over the population the benchmark selects and shapes, and place collapse in the scoring rule.
2. **Mechanism.** MOSAIC pays a valid response the log-determinant coverage it adds beyond the record, under a declared evaluation unit, validity gate, and feature map.
3. **Theory.** A sharp quality–novelty threshold below which MOSAIC's equilibria are exactly the complementary allocations of the best modes and standalone's is collapse; an exact potential under which decentralized improvement converges, below the threshold to a globally optimal population; linear accrual under a persistent reference against standalone's $\Theta(\log T)$; and a cap on one capability split into k pieces.
4. **Evidence.** Each row of Table 5 changes one factor to test one claim: twelve rounds of adaptation end at 32 pooled effective modes against 15, a gain that comes from the record's lifetime and not its scope, and fresh-state commits preserve the potential on all 18 concurrent-release schedules, where a stale reference cycles on 8.

2 The MOSAIC Score: Population Coverage and What an Entry Adds

We proposed paying a response for what it adds to the record; here we make "adds" a number. A score that evaluates one response at a time cannot tell a capability no developer has from one every developer already has. Telling them apart means comparing a response against those already on record, so the evaluator declares three things: the evaluation unit, which fixes what one record is; the validity gate, which fixes which records count; and the feature map, which fixes which differences count as different capabilities, the question pluralistic alignment asks of a target distribution (Sorensen et al., 2024). Two conventions hold throughout. *A record* is one admitted response and *the record* is every response admitted so far, the set those three declarations apply to. A *capability* is a direction of the feature map; in Section 4 we call it a *mode* once outputs are collapsed onto it.

2.1 The Population Objective

Each admitted response is one record i in $\mathcal{U}$, the ground set of records, and during a fixed epoch we give it a feature vector $\phi_i \in \mathbb{R}^d$; identical outputs remain distinct records. Any embedding serves as the feature map, since every positive-semidefinite kernel factors into one. We apply the validity gate before computing the geometry (Section G.2): its weight $w_i \geq 0$ fixes which records count,

and $w_i = 0$ excludes an output. For finite $S \subset \mathcal{U}$, coverage is

$$V(S) = \log\det\left(\mathbf{I} + \alpha \sum_{i \in S} w_i \phi_i \phi_i^\top\right) = \sum_r \log(1 + \alpha\lambda_r), \qquad \alpha > 0, \tag{1}$$

over the eigenvalues λ_r of the weighted scatter matrix. We call V the *population objective*: the evaluator wants the record to maximize it, and no developer is paid it directly. Each eigenvalue is one axis of the ellipsoid; the logarithm is concave, so mass on a crowded axis raises V less, and α sets where it saturates. Throughout we set $\alpha = 1$ and use unit-norm features, under which n identical copies of one capability are paid, in total, less than 1.44 times what a single entry on an uncovered direction earns (Corollary 3 and Section D.1).

2.2 What an Entry Is Paid

A developer is paid not the objective but what its entry adds to it.

Definition 1 (Log-determinant MOSAIC score)**.** *For a finite declared reference $S \subset \mathcal{U}$ and a candidate $i \in \mathcal{U} \setminus S$, the MOSAIC score is the coverage the candidate adds, $\Delta_i(S) = V(S \cup \{i\}) - V(S)$.*

We compare candidates at a fixed S, since otherwise a score difference conflates a candidate change with a reference change. We keep three sets apart: the record is everything admitted, the *reference* S is the part a given decision's candidates are scored against, and the *population* is the set of entries developers currently hold (Section 2.4). A reference is *shared* when the alternatives in one decision face the same S and *persistent* when accepted records stay in it for later decisions. Regularized log determinants are classical design objectives (Krause & Golovin, 2014; Kulesza & Taskar, 2012); MOSAIC uses the corresponding contribution as the payment (Section C).

Writing L_S for the reference's regularized scatter $\mathbf{I} + \alpha \sum_{j \in S} w_j \phi_j \phi_j^\top$, the matrix determinant lemma gives the local form

$$\Delta_i(S) = \log\left(1 + \alpha w_i \phi_i^\top L_S^{-1} \phi_i\right). \tag{2}$$

Here L_S^{-1} whitens by the population, shrinking whatever part of ϕ_i the reference already explains. That quadratic form is a squared residual norm, and specifically the ridge leverage score (El Alaoui & Mahoney, 2015) of regression against the standing population. A high score therefore means the standing population predicts the candidate poorly: the score is a joint property of candidate and record, which we test on real programs in Section 5.1.

2.3 Why the Objective Has the Right Shape

That score is well behaved because of one property of the objective it comes from. For positive-semidefinite features and fixed nonnegative weights V is normalized, monotone and submodular, so $\Delta_i(A) \geq \Delta_i(B) \geq 0$ for $A \subseteq B$ (Lemma 1). The equilibrium results below use only three consequences of the geometry: a valid response adds nonnegative coverage, a copy of a represented direction keeps a positive but shrinking increment, and an unspanned direction earns the full $\log(1 + \alpha w_i \|\phi_i\|^2)$. Under a Gaussian quality prior the score is also twice the information one further evaluation adds, which is why coverage is in nats and D-optimal design becomes a developer incentive (Proposition 5; Lindley, 1956; Chaloner & Verdinelli, 1995).

2.4 From Score to Mechanism

Three rules turn the score into a mechanism, and we state them in full in Section D. Each of N developers controls one entry drawn from a finite menu $\mathcal{A}_i$, so entries and not actions are the game's ground set. The evaluator declares once per epoch a weighted prompt suite and a background H of records that no developer owns and no developer can change, so a set C of active entries has represented coverage

$$G_H(C) := V(H \cup C) - V(H), \tag{3}$$

the coverage the population adds above that background, which inherits normalization, monotonicity and submodularity from V. Writing $A(a)$ for the population at profile a and $A_{-i}(a)$ for it with entry i removed, developer i is paid its leave-one-out contribution $s_i(a) = G_H\big(A_{-i}(a) \cup \{a_i\}\big) -$

$G_H\big(A_{-i}(a)\big)$ (Shapley, 1953; Han et al., 2023), so incumbent and proposal face the same A_{-i}, and accepted writes are rescored against the current population before they commit. Three decisions stay separate throughout. The validity gate decides *admission*, whether a response becomes a record at all; s_i is the *score*, the number the evaluator computes for an admitted entry; and that number is also the *payment*, in the mechanism-design sense that the mechanism values a replacement at $b_i(a_i) + \omega\, s_i(a)$, the payoff we analyze in Section 3, where $\omega > 0$ converts a nat of coverage into units of quality. An application may spend that value on slot admission or on selecting training data (Section 1); a reward depending on where it *ranks* is a different game, outside Theorem 2. The published standalone score may stay the capability report while s_i decides which entry fills a slot: where unilateral gains under represented coverage approximate those under a monotone submodular endpoint utility, the half-optimum guarantee transfers to that utility (Theorem 3).

3 Stable Populations and How Developers Reach Them

3.1 Stable Populations: The Quality–Novelty Threshold

We have so far fixed what one developer is paid. What it is paid for, however, depends on choices the other developers are making at the same time, so we now ask which profiles of those choices are stable. We take the menu of Section 1 with strict quality differences: each of N developers chooses one of $d \geq N$ orthogonal unit modes $e_1, \ldots, e_d$ over an empty background at unit weights, and mode k has intrinsic quality $b_k \geq 0$ with $b_1 > \cdots > b_d$. Developer i receives $u_i(a) = b_{a_i} + \omega\, s_i(a)$ under MOSAIC and $\widetilde{u}_i(a) = b_{a_i}$ standalone, $\omega > 0$ weighting coverage. Two constants determine the equilibria: the quality range $R_b := b_1 - b_d$ and the novelty premium $\chi_\alpha := \log\big((1+\alpha)^2/(1+2\alpha)\big)$, the payment gap between a mode's first and second occupant, $\log(4/3) \approx 0.29$ nats at $\alpha = 1$.

Theorem 1 (Sharp quality–novelty threshold)**.** *Under the orthogonal menu and payoffs above, suppose $R_b < \omega\chi_\alpha$. Then the MOSAIC pure Nash profiles are exactly the no-repeat allocations of the N highest-quality modes, and every one of them globally maximizes $\sum_i b_{a_i} + \omega V(a)$. The standalone game has a unique pure Nash profile, every developer on mode 1. The threshold is tight: for $N = d = 2$ and $R_b \geq \omega\chi_\alpha$, full collapse onto mode 1 is also a MOSAIC pure Nash profile (proof in Section B.5).*

One exchange proves all four claims of the theorem. A developer alone on a mode is paid $\log(1+\alpha)$ of coverage and a second occupant only $\log\big((1+2\alpha)/(1+\alpha)\big)$; the difference is χ_α. Leaving a shared mode for an unused one therefore gains at least $\omega\chi_\alpha$ in coverage and loses at most R_b in quality. Below the threshold, therefore, every developer sharing a mode has a profitable move, and every no-repeat profile that skips a top-N mode has a developer who gains by taking it. In a no-repeat allocation of the top N modes nobody does. The same exchange raises $\sum_i b_{a_i} + \omega V(a)$, which makes each of those allocations a global maximizer. At equal qualities the price of anarchy (Roughgarden, 2015) is one, while standalone collapse attains only a $\log(1+\alpha N)/(N\log(1+\alpha))$ fraction of optimal coverage, vanishing as N grows (Proposition 2). Monoculture's own bound is a tight price of anarchy of two (Kleinberg et al., 2026), where what agents give up is accuracy (Section I).

3.2 Decentralized Convergence via an Exact Potential

We next ask whether unilateral improvements, made by developers who see only their own payoff, are guaranteed to reach such a population; holding out the developer's own entry is what makes the answer yes. In the epoch of Section 2.4 give developer i a declared intrinsic quality $b_i(a_i) \geq 0$ per menu item and payoff $u_i(a) = b_i(a_i) + \omega\, s_i(a)$, with $b \equiv 0$ the pure coverage game, and call quality plus coverage the population objective, $\Pi(a) := \sum_i b_i(a_i) + \omega\, G_H(A(a))$. The held-out set is the same on both sides of a replacement $a_i \to a_i'$, so what the rest already covers cancels and developer i's payoff changes by exactly as much as Π does: Π is an *exact potential* (Monderer & Shapley, 1996), and every unilateral improvement raises Π, so improvement paths cannot cycle.

Theorem 2 (Quality-plus-coverage potential and asynchronous improvement)**.** *Under the fixed-epoch conditions of Section 2.4, Π is an exact potential for the quality-plus-coverage game: every unilateral replacement satisfies*

$$u_i(a_i', a_{-i}) - u_i(a_i, a_{-i}) = \Pi(a_i', a_{-i}) - \Pi(a_i, a_{-i}). \tag{4}$$

Consequently every serialized strict-improvement path is finite and, if maximal, ends at a pure Nash profile, so an asynchronous service reaches such a profile after finitely many accepted replacements provided no developer is starved of activation forever; and every pure Nash profile a^{NE} satisfies

$$\Pi(a^{\mathrm{NE}}) \geq \tfrac{1}{2} \max_{b \in \prod_i \mathcal{A}_i} \Pi(b). \tag{5}$$

(proof in Section B.2).

Corollary 1 (Decentralized improvement reaches a complementary optimum)**.** *On the menu of Theorem 1 with $R_b < \omega\chi_\alpha$, every maximal serialized strict-improvement path ends at a no-repeat allocation of the N highest-quality modes, a global maximizer of* Π.

Proof. By Theorem 2 the path ends at a pure Nash profile, and by Theorem 1 every pure Nash profile is such an allocation. □

The corollary is the implementation result: no developer is told where the gaps are, and the population still arrives at the best complementary allocation. Off the orthogonal menu the 50% floor in (5) is the standard bound for submodular competitive games (Vetta, 2002; Nemhauser et al., 1978); any strictly increasing transform of s_i leaves the equilibria unchanged (Proposition 3), and a rank reward, which depends on other developers' scores, is a different game.

3.3 Concurrent Updates: Fresh-State Commits Preserve the Potential

The service handles concurrency at commit time. Identity (4) compares incumbent and proposal against the same current A_{-i}, and a proposal scored against a stale population is credited for capability another commit has already added. For example, two developers preparing the same missing capability against one snapshot both appear to improve, both commit, and the second is paid for coverage the first added. If the service rescores each proposal against the population at commit time and commits it only if it is still strictly better, every commit is a unilateral move and hence a verified increase in Π.

4 Persistent Reference: Linear Diversification Before Saturation

In Section 3 we fixed N replaceable entries and characterized the pure Nash profiles developers reach unilaterally. Convergence, however, is the right guarantee for one epoch and the wrong one for a leaderboard whose job is to keep rewarding modes the population lacks. A *persistent* reference instead keeps every accepted record in the comparison set, so the state only grows, the capabilities already in the reference are paid least next round, and the question becomes how long the score keeps eliciting new modes. This regime differs from that of Section 3 and Theorem 2: the active entries no longer describe the whole state, because every accepted record stays in the reference after the entry that produced it is replaced.

For the persistent reference we fix one prompt, a feature map, and a canonical mode map μ collapsing gate-admitted outputs into the *modes* of Fig. 1. Two paraphrases of one solution are one mode. Each mode s has a unit feature ϕ_s and a fixed validity weight $v_s \in [v_{\min}, v_{\max}]$, $v_{\min} > 0$. Each of N developers makes one gate-admitted submission per round, and we serialize the insertions by a fixed public order, each entering the reference before the next developer acts. Writing $\sigma_k = \mu(z_k)$ for the mode of the kth insertion, the reference evolves as $L_{k+1} = L_k + \alpha v_{\sigma_k} \phi_{\sigma_k} \phi_{\sigma_k}^\top$ from $L_0 = \mathbf{I}$. After T rounds, we write $m_s(T)$ for the gated mass at mode s, $p_s(T) \propto m_s(T)$ for its share, D_T for the number of distinct modes ever played, and $V_T := \log\det L_{NT}$ for the accrued coverage. We compute two *effective mode counts* from the share vector $p(T)$ (Hill, 1973):

$$\text{Hill-2:} \quad N_{\mathrm{eff}}(T) := \Big[\sum\nolimits_s p_s(T)^2\Big]^{-1}, \qquad \text{Hill-1:} \quad \exp\Big(-\sum\nolimits_s p_s(T) \log p_s(T)\Big). \tag{6}$$

Both equal k when mass spreads evenly over k modes and fall toward one as one mode takes over; Hill-2 never exceeds Hill-1, so every bound below is stated for Hill-2 and holds for both. For example, ten records split evenly between the loop and the identity of Fig. 1 yield $N_{\mathrm{eff}} = 2$, and the same ten with nine on the loop yield 1.22.

The mode count grows linearly under two conditions on the realized path, one on what developers can still do and one on what they choose. A path is *pathwise δ-rich* when the acting developer can always produce an admitted output whose whitened residual satisfies $\phi^\top L_k^{-1}\phi \geq \delta$ against the current reference L_k, that is, an output at least a fraction δ of whose feature mass the reference has yet to explain. A path is *pathwise γ-rational* when every chosen output earns at least a γ fraction of the best score then available in its own action set (Definitions 4 and 5). We measure both on the 18,432 insertions of Section 5.3. That replay selects by argmax, so $\gamma = 1$; the measured δ is 0.13 under an own persistent reference against 0.048 under a shared one. The arm with more breadth thus has the smaller δ: a shared history explains more of every direction, so every residual shrinks, while the shared history also keeps developers off the directions the others have taken. In Proposition 1 δ is a worst-case floor on what remains available, not a forecast of breadth.

Beyond orthogonal modes. Orthogonal modes isolate the mechanism in its simplest form (Theorem 1): once a mode is represented, a further copy earns less than an unused orthogonal mode. Under feature overlap, unequal weights, approximate responses and persistence the same principle caps the mass any one mode can accumulate: a fresh option always scores at least a floor c_δ while an insertion at a saturated mode scores less as its mass grows (Lemmas 2 and 3), so past a fixed mass it scores under γ times the floor. We write the floor and the cap as $c_\delta := \log(1 + \alpha v_{\min}\delta)$ and $\bar{m} := v_{\max}/(e^{\gamma c_\delta} - 1) + v_{\max}$; raising either δ or γ shrinks $\bar{m}$, and the second $v_{\max}$ covers the one serialized insertion crossing the threshold.

Proposition 1 (Persistent mode saturation)**.** *Fix a horizon T and suppose the realized path is pathwise δ-rich and pathwise γ-rational through the first NT insertions. Then every mode obeys the uniform mass bound $m_s(T) \leq \bar{m}$. Consequently every fixed finite set F of modes obeys $\sum_{s\in F} p_s(T) \leq |F|\bar{m}/(Nv_{\min}T)$, and*

$$D_T \;\geq\; \frac{Nv_{\min}}{\bar{m}}\,T, \qquad N_{\text{eff}}(T) \;\geq\; \frac{Nv_{\min}}{\bar{m}}\,T, \qquad V_T \;\geq\; \gamma c_\delta NT. \tag{7}$$

The proof is given in Section E.1.

The proposition needs no equilibrium assumption, and in the setting of Theorem 1 ($\delta = \gamma = 1$, unit weights) the cap simplifies to $\bar{m} = 1+1/\alpha$. With the measured constants, (7) gives $D_T \geq 2.7$ modes per prompt against 5.7 realized, a factor of 2.1. In contrast, changing only the reward, to a standalone score with a unique best mode, makes every developer submit that one mode, and $V_T = \log(1 + \alpha v_{s^\star} NT) = \Theta(\log T)$ on every horizon. The linear-against-logarithmic gap is therefore attributable to the reference alone (Corollary 4 and Section I). The feature map's dimension ends the linear phase: with unit features in dimension d no policy accrues more than $V_T \leq d\log(1 + \alpha v_{\max} NT/d)$, so the linear phase of Proposition 1 is the approach to a ceiling that rises with the effective dimension of the declared feature representation (Corollary 5).

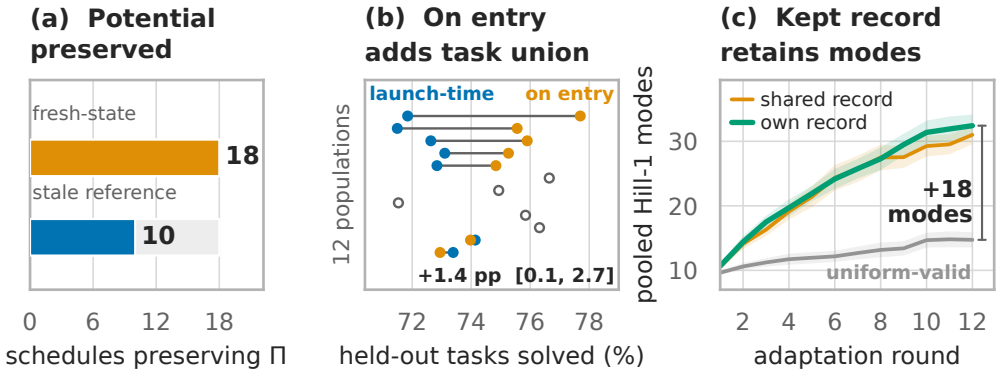
5 Empirical Evaluation

So far we have reasoned about the reference from the definition of the score. In this section and the next we vary the reference and nothing else, one row of Table 5 at a time (Section H), holding the candidates, gate, feature map and update order fixed within each comparison. Refining shared and persistent from Section 2.2, we call a reference's lifetime *reset* when rebuilt from the current round, *persistent* when it keeps every record ever admitted, and *launch-time* when fixed once; its scope is one developer's own history or one shared across all of them. Two controls fall outside this classification: a *standalone* score, which keeps no reference and which we run as uniform-valid selection, and a *stale* one, scored against a population that has since changed. Coverage is G_H from (3) in nats and breadth is Hill-1 effective modes from (6) pooled over developers; Table 3 lists the full statistics beside the protocols in Section H.

5.1 Does the Feature Map Separate Modes?

The feature map separates algorithms and ignores renaming. Because the feature map determines which differences change the score, we test it first. On a paired HumanEval panel the declared abstract-syntax signature ignores identifiers and comments exactly: five rewrites of one solution yield $\ln 6 =$ **1.79** nats on all 164 problems, the value of five coincident unit vectors, while

Figure 2: **MOSAIC preserves the population geometry under both feedback loops.** **(a)** counts the release schedules that preserve the potential; an open marker in **(b)** is a population the reference leaves unchanged; **(c)** pools Hill-1 modes over twelve rounds, with standard-error bands.



five structurally distinct solutions yield **2.60**, **48%** of the way to the $5 \ln 2 = 3.47$ ceiling of five orthogonal directions. Validity comes from executable tests, independent of the map (Section H.2; Chen et al., 2021).

5.2 Do Decentralized and Concurrent Updates Preserve the Potential?

Fresh-state rescoring preserves the potential in every cell. With the representation of Section 5.1, the potential identity of (4) holds on all 36,864 non-incumbent unilateral replacements of the 12 four-slot games we seed, and every maximal improvement path ends at a pure Nash profile whose coverage is 0.99 of optimal. What breaks monotonicity is concurrency, not decentralization: across population sizes 4 to 32 every fresh-state cell leaves the potential non-decreasing, while **88%** of stale controls lower it (Section H.4).

5.3 Does a Shared Persistent Reference Increase Breadth?

A shared persistent reference closes all but 0.2% of the gap to the oracle. Fresh-state commits keep the potential rising; the population's final breadth depends on whether the reference is also kept and shared. On a 2×2 replay of 147,456 frozen Qwen2.5-0.5B-Instruct candidates, crossing scope with lifetime at fixed event order (Qwen Team et al., 2024), the shared persistent arm ends **0.2%** of the uniform-to-oracle range below a greedy oracle that takes the least represented mode at every insertion (Fig. 4a, in Section H). Along the ordered path from uniform selection to an own reset reference, then to persistence, then to sharing, the three steps account for 4%, **77%** and **19%** of that range: a record of which modes the population already contains does almost nothing unless it is kept. Scope and lifetime interact, so this is the decomposition along that path and not the only one. Correctness, which the selection rule never uses, stays at 35% under either persistent reference (Fig. 4b, Section H.7). Live adaptation gives the same ordering: lifetime accounts for the gain, and sharing adds nothing measurable once the record is kept (Section 6.2).

6 Case Studies: Benchmarks That Change Their Own Population

So far we varied the reference on replays we constructed. We now run the same substitution inside two deployed loops that already change their own population: a multi-agent system whose workers commit concurrently, and a learner whose next round trains on what it just accepted.

6.1 Concurrent Multi-Agent Systems

Fresh-state admission prevents concurrent regressions. Recomputing each score against the current population preserves coverage in all 18 release schedules, whereas keeping the score computed when the proposal was prepared lets the potential cycle in 8 (Fig. 2a). We replay solved-task

vectors from 14 official LiveCodeBench releases, each filling one slot, across those 18 provider-order and overlap schedules (Jain et al., 2025). Concurrency is what separates the two: stale acceptance cycles in none of the 6 serialized schedules and in all 6 fully concurrent ones.

The population gains capability at unchanged individual quality. Scoring a proposal against the population it enters raises the held-out task union at unchanged per-slot accuracy, and on the later LiveCodeBench window the two populations separate (Fig. 2b, Table 5). Each unit is a four-slot population of fixed pretrained releases: we train no model and score on a separate 490-item SVAMP+ASDiv portfolio (Section H.6; Patel et al., 2021; Miao et al., 2020).

With the four candidate patches held fixed, the commit rule closes 15% of RSI-Exam's baseline-to-SOTA gap. Four workers each prepare one patch for one deployed system, the same concurrency as in the release replay (RSI-Exam Team, 2026). We replay the same four patches, all of which pass the public gate, under both rules, so the commit rule is the only difference between the two arms: fresh-state admission keeps the public-best worker, while stale acceptance takes all four, makes three commits that lower the potential, and ends on the public-worst. The sequential control follows a different workflow and reaches the stale arm's terminal policy (Section H.8).

6.2 Continual Adaptation

The kept record more than doubles the effective modes retained across training rounds, whether it is the developer's own or the population's. Here the population changes because the next round trains on what this one accepted. Three Qwen2.5-0.5B-Instruct developers trained on fresh 32-prompt batches for twelve rounds end at **32** pooled Hill-1 effective modes when each is scored against its own persistent record and 31 when scored against the population's, against 15 under uniform-valid selection, over 24 matched seeds (Fig. 2c; Qwen Team et al., 2024). We cross scope with lifetime to decompose the gain: scoring against a reference at all adds 7 modes, keeping it across rounds adds 11, and sharing it across the three developers adds -1.4 $[-5.6, 2.8]$, so under adaptation the record's lifetime and not its scope sets the breadth. Held-out accuracy changes by -1.9 points $[-3.7, 0.0]$ and the held-out task union by 0.2 $[-1.6, 2.0]$ (Section H.9).

7 Related Work

Three lines of work are adjacent: one optimizes diversity inside a single model's sampled group, where the set is that model's own samples; a second refreshes the test distribution (Nie et al., 2020; Kiela et al., 2021; White et al., 2025; Jain et al., 2025; Zhu et al., 2024; Mirzadeh et al., 2025; Li et al., 2025b); a third studies what a published objective does to the systems that later face it (Perdomo et al., 2020; Hardt et al., 2022; Dominguez-Olmedo et al., 2025; Chen et al., 2026b). All three grade each answer against a fixed key or a fixed geometry, so one developer's best response never depends on what the others submitted. MOSAIC pays less for a mode the more prevalent it is in the standing population, on the exact-potential foundation of marginal-contribution games (Monderer & Shapley, 1996; Vetta, 2002). We work that comparison out in Section I against peer prediction, market scoring rules, contests, monoculture, clone robustness, evaluation aggregation and minority rewards.

8 Conclusion

Dynamic benchmarking can fundamentally be a mechanism design problem. MOSAIC pays a valid response the log-determinant coverage it adds beyond the record, so quality plus coverage is an exact potential: every improvement path ends at a pure Nash profile, and below a sharp quality–novelty threshold those profiles are exactly the no-repeat allocations of the best modes. Keeping every past response in the comparison set makes coverage grow linearly while fresh directions remain, against $\Theta(\log T)$ for a standalone score; in our experiments twelve rounds of adaptation end at 32 pooled effective modes against 15.

AI USE STATEMENT

We used generative AI tools across tasks that ICLR asks authors to disclose. On the required list: the tools implemented methods and analysis pipelines, proposed and refined hypotheses, gave feedback on experimental design, helped develop the conceptual framing, formulated and drafted parts of the mathematical claims and their proofs, cleaned and reformatted result records, and assisted in interpreting results. The frozen candidate logs, the replayed populations, and the adaptation runs are model outputs produced by the released checkpoints named in Section H; they are real model outputs and the object of measurement, and no generative AI tool produced any reported measurement. We did not use these tools for translation or for qualitative and thematic data analysis. On the recommended list: they drafted and edited prose, wrote and edited software, generated the figures, searched and summarized literature, formatted references, and supported brainstorming.

We have reviewed all AI-assisted work and take responsibility for the final content of this paper, including all text, claims, and artifacts.

ETHICS STATEMENT

This work studies how a benchmark's scoring rule changes the population of models it later evaluates. No human subjects are involved, and every dataset is public and used under its own license. Making a score population-relative admits manipulations that a standalone score does not: an entrant can try to buy coverage by padding the gate or by partitioning one behavior into several represented directions. Section G states that boundary and the deviations the leave-one-out identity does and does not cover, so a deployer can see what the mechanism assumes before running it.

REPRODUCIBILITY STATEMENT

The supplement contains complete proofs, protocol details, hyperparameters, and additional results. The anonymous artifact includes executable mechanism code, frozen analysis scripts, tests, and machine-readable records that bind reported claims to raw outputs.

REFERENCES

Yasin Abbasi-Yadkori, Dávid Pál, and Csaba Szepesvári. Improved algorithms for linear stochastic bandits. In *Advances in Neural Information Processing Systems*, volume 24, 2011.

Jacob Abernethy, Yiling Chen, and Jennifer Wortman Vaughan. Efficient market making via convex optimization, and a connection to online learning. *ACM Transactions on Economics and Computation*, 1(2):1–39, 2013. doi: 10.1145/2465769.2465777.

W. Brian Arthur. Inductive reasoning and bounded rationality. *The American Economic Review*, 84 (2):406–411, 1994.

Avrim Blum and Moritz Hardt. The Ladder: A reliable leaderboard for machine learning competitions. In *Proceedings of the 32nd International Conference on Machine Learning (ICML)*, volume 37 of *Proceedings of Machine Learning Research*, pp. 1006–1014. PMLR, 2015. URL https://proceedings.mlr.press/v37/blum15.html.

Rishi Bommasani, Kathleen A. Creel, Ananya Kumar, Dan Jurafsky, and Percy Liang. Picking on the same person: Does algorithmic monoculture lead to outcome homogenization? In *Advances in Neural Information Processing Systems (NeurIPS)*, volume 35, pp. 3663–3678, 2022. doi: 10.52202/068431-0265. arXiv:2211.13972.

Jaime Carbonell and Jade Goldstein. The use of MMR, diversity-based reranking for reordering documents and producing summaries. In *Proceedings of the 21st Annual International ACM SIGIR Conference on Research and Development in Information Retrieval*, pp. 335–336, 1998. doi: 10.1145/290941.291025.

Kathryn Chaloner and Isabella Verdinelli. Bayesian experimental design: A review. *Statistical Science*, 10(3):273–304, 1995. doi: 10.1214/ss/1177009939.

Shuchi Chawla, Jason D. Hartline, and Balasubramanian Sivan. Optimal crowdsourcing contests. *Games and Economic Behavior*, 113:80–96, 2019. doi: 10.1016/j.geb.2015.09.001. Preliminary version in SODA 2012.

Yeon-Koo Che and Ian Gale. Optimal design of research contests. *American Economic Review*, 93 (3):646–671, 2003. doi: 10.1257/000282803322157025.

Haoxuan Chen, Tianming Liang, Wei-Shi Zheng, and Jian-Fang Hu. Breaking winner-takes-all: Cooperative policy optimization improves diverse LLM reasoning. *arXiv preprint arXiv:2605.11461*, 2026a.

Mark Chen, Jerry Tworek, Heewoo Jun, Qiming Yuan, Henrique Ponde de Oliveira Pinto, Jared Kaplan, Harri Edwards, Yuri Burda, Nicholas Joseph, Greg Brockman, et al. Evaluating large language models trained on code. *arXiv preprint arXiv:2107.03374*, 2021.

Yatong Chen, Guanhua Zhang, and Moritz Hardt. Leaderboard incentives: Model rankings under strategic post-training. In *Proceedings of the 43rd International Conference on Machine Learning*, 2026b. URL https://openreview.net/forum?id=r6wfuAKmVb. arXiv:2603.08371.

Yilei Chen, Souradip Chakraborty, Lorenz Wolf, Ioannis Paschalidis, and Aldo Pacchiano. Post-training large language models for diverse high-quality responses. In *The Fourteenth International Conference on Learning Representations*, 2026c. URL https://proceedings.iclr.cc/paper_files/paper/2026/hash/cfeb1fa78c5fb06be27c0980aa8ee581-Abstract-Conference.html. arXiv:2509.04784.

Wei-Lin Chiang, Lianmin Zheng, Ying Sheng, Anastasios N. Angelopoulos, Tianle Li, Dacheng Li, Banghua Zhu, Hao Zhang, Michael Jordan, Joseph E. Gonzalez, and Ion Stoica. Chatbot Arena: An open platform for evaluating LLMs by human preference. In *Proceedings of the 41st International Conference on Machine Learning*, volume 235 of *Proceedings of Machine Learning Research*, pp. 8359–8388. PMLR, 2024. URL https://proceedings.mlr.press/v235/chiang24b.html. arXiv:2403.04132.

Vincent Conitzer and Makoto Yokoo. Using mechanism design to prevent false-name manipulations. *AI Magazine*, 31(4):65–78, 2010. doi: 10.1609/aimag.v31i4.2315.

Thomas M Cover and Joy A Thomas. *Elements of Information Theory*. Wiley-Interscience, 2nd edition, 2006. ISBN 9780471241959.

Ricardo Dominguez-Olmedo, Florian E. Dorner, and Moritz Hardt. Training on the test task confounds evaluation and emergence. In *The Thirteenth International Conference on Learning Representations*, 2025. URL https://openreview.net/forum?id=jOmk0uS1hl. Oral; arXiv:2407.07890.

Florian E. Dorner, Vivian Y. Nastl, and Moritz Hardt. Limits to scalable evaluation at the frontier: LLM as judge won't beat twice the data. In *The Thirteenth International Conference on Learning Representations*, 2025. URL https://openreview.net/forum?id=N06Tv6QcDs. Oral; arXiv:2410.13341.

Cynthia Dwork, Vitaly Feldman, Moritz Hardt, Toniann Pitassi, Omer Reingold, and Aaron Roth. The reusable holdout: Preserving validity in adaptive data analysis. *Science*, 349(6248):636–638, 2015. doi: 10.1126/science.aaa9375.

Ahmed El Alaoui and Michael W. Mahoney. Fast randomized kernel ridge regression with statistical guarantees. In *Advances in Neural Information Processing Systems (NeurIPS)*, volume 28, pp. 775–783, 2015.

Shi Feng, Hanlin Zhang, Fan Nie, Sham M. Kakade, and Yiling Chen. Peer-predictive self-training for language model reasoning. In *Third Conference on Language Modeling*, 2026. URL https://openreview.net/forum?id=1yQ625JKs0.

Dan Friedman and Adji Bousso Dieng. The Vendi score: A diversity evaluation metric for machine learning. *Transactions on Machine Learning Research*, 2023.

Gemma Team, Morgane Riviere, Shreya Pathak, et al. Gemma 2: Improving open language models at a practical size. *arXiv preprint arXiv:2408.00118*, 2024. URL https://arxiv.org/abs/2408.00118.

Tilmann Gneiting and Adrian E. Raftery. Strictly proper scoring rules, prediction, and estimation. *Journal of the American Statistical Association*, 102(477):359–378, 2007. doi: 10.1198/016214506000001437.

Dongge Han, Michael Wooldridge, Alex Rogers, Olga Ohrimenko, and Sebastian Tschiatschek. Replication robust payoff allocation in submodular cooperative games. *IEEE Transactions on Artificial Intelligence*, 4(5):1114–1128, 2023. doi: 10.1109/TAI.2022.3195686. arXiv:2006.14583.

Robin Hanson. Combinatorial information market design. *Information Systems Frontiers*, 5(1): 107–119, 2003. doi: 10.1023/A:1022058209073.

Moritz Hardt, Meena Jagadeesan, and Celestine Mendler-Dünner. Performative power. In *Advances in Neural Information Processing Systems (NeurIPS)*, volume 35, pp. 22969–22981, 2022. doi: 10.52202/068431-1669. URL https://proceedings.neurips.cc/paper_files/paper/2022/hash/90e73f3cf1a6c84c723a2e8b7fb2b2c1-Abstract-Conference.html. arXiv:2203.17232.

Chris Hays, Rachel Li, Bailey Flanigan, and Manish Raghavan. Strategic candidacy in generative AI arenas. In *Proceedings of the 43rd International Conference on Machine Learning*, 2026. URL https://openreview.net/forum?id=K1mAvyHhKU. arXiv:2603.26891.

M. O. Hill. Diversity and evenness: A unifying notation and its consequences. *Ecology*, 54(2): 427–432, 1973. doi: 10.2307/1934352.

Harold Hotelling. Stability in competition. *The Economic Journal*, 39(153):41–57, 1929. doi: 10.2307/2224214.

Naman Jain, King Han, Alex Gu, Wen-Ding Li, Fanjia Yan, Tianjun Zhang, Sida Wang, Armando Solar-Lezama, Koushik Sen, and Ion Stoica. LiveCodeBench: Holistic and contamination free evaluation of large language models for code. In *The Thirteenth International Conference on Learning Representations*, 2025. URL https://openreview.net/forum?id=chfJJYC3iL.

Liwei Jiang, Yuanjun Chai, Margaret Li, Mickel Liu, Raymond Fok, Nouha Dziri, Yulia Tsvetkov, Maarten Sap, and Yejin Choi. Artificial hivemind: The open-ended homogeneity of language models (and beyond). In *Advances in Neural Information Processing Systems*, volume 38, pp. 90705–90774, 2025. doi: 10.52202/085713-2732. URL https://proceedings.neurips.cc/paper_files/paper/2025/hash/754d5a526a5ee5a47220664a0eb92751-Abstract-Datasets_and_Benchmarks_Track.html.

Douwe Kiela, Max Bartolo, Yixin Nie, Divyansh Kaushik, Atticus Geiger, Zhengxuan Wu, Bertie Vidgen, Grusha Prasad, Amanpreet Singh, Pratik Ringshia, Zhiyi Ma, Tristan Thrush, Sebastian Riedel, Zeerak Waseem, Pontus Stenetorp, Robin Jia, Mohit Bansal, Christopher Potts, and Adina Williams. Dynabench: Rethinking benchmarking in NLP. In *Proceedings of the 2021 Conference of the North American Chapter of the Association for Computational Linguistics: Human Language Technologies*, pp. 4110–4124. Association for Computational Linguistics, 2021. doi: 10.18653/v1/2021.naacl-main.324. URL https://aclanthology.org/2021.naacl-main.324/.

Jung-Hun Kim, Milan Vojnovic, and Se-Young Yun. Rotting infinitely many-armed bandits. In *Proceedings of the 39th International Conference on Machine Learning*, volume 162 of *Proceedings of Machine Learning Research*, pp. 11229–11254. PMLR, 2022. URL https://proceedings.mlr.press/v162/kim22j.html.

Alex Kipnis, Konstantinos Voudouris, Luca M. Schulze Buschoff, and Eric Schulz. metabench: A sparse benchmark of reasoning and knowledge in large language models. In *The Thirteenth International Conference on Learning Representations*, 2025. URL https://openreview.net/forum?id=4T33izzFpK. Poster; arXiv:2407.12844.

Robert Kirk, Ishita Mediratta, Christoforos Nalmpantis, Jelena Luketina, Eric Hambro, Edward Grefenstette, and Roberta Raileanu. Understanding the effects of RLHF on LLM generalisation and diversity. In *International Conference on Learning Representations (ICLR)*, 2024.

Jon Kleinberg and Manish Raghavan. Algorithmic monoculture and social welfare. *Proceedings of the National Academy of Sciences*, 118(22):e2018340118, 2021. doi: 10.1073/pnas.2018340118. arXiv:2101.05853.

Robert Kleinberg, Erald Sinanaj, and Éva Tardos. Price of anarchy of algorithmic monoculture. *arXiv preprint arXiv:2604.00444*, 2026. Earlier version presented at WINE 2025.

Yuqing Kong and Grant Schoenebeck. An information theoretic framework for designing information elicitation mechanisms that reward truth-telling. *ACM Transactions on Economics and Computation*, 7(1):1–33, 2019. doi: 10.1145/3296670.

Andreas Krause and Daniel Golovin. Submodular function maximization. In *Tractability: Practical Approaches to Hard Problems*, pp. 71–104. Cambridge University Press, 2014. doi: 10.1017/CBO9781139177801.004. URL https://www.cambridge.org/core/books/abs/tractability/submodular-function-maximization/C1925D301CBF7D7BBC55E44644A1ABA1.

Andreas Krause, Ajit Singh, and Carlos Guestrin. Near-optimal sensor placements in Gaussian processes: Theory, efficient algorithms and empirical studies. *Journal of Machine Learning Research*, 9(8):235–284, 2008.

Alex Kulesza and Ben Taskar. Determinantal point processes for machine learning. *Foundations and Trends in Machine Learning*, 5(2–3):123–286, 2012. doi: 10.1561/2200000044.

Nathan Lambert, Valentina Pyatkin, Jacob Morrison, LJ Miranda, Bill Yuchen Lin, Khyathi Chandu, Nouha Dziri, Sachin Kumar, Tom Zick, Yejin Choi, Noah A. Smith, and Hannaneh Hajishirzi. RewardBench: Evaluating reward models for language modeling. In *Findings of the Association for Computational Linguistics: NAACL 2025*, pp. 1755–1797. Association for Computational Linguistics, 2025. doi: 10.18653/v1/2025.findings-naacl.96. URL https://aclanthology.org/2025.findings-naacl.96/. arXiv:2403.13787.

Igor Letina and Armin Schmutzler. Inducing variety: A theory of innovation contests. *International Economic Review*, 60(4):1757–1780, 2019. doi: 10.1111/iere.12403.

Nir Levine, Koby Crammer, and Shie Mannor. Rotting bandits. In *Advances in Neural Information Processing Systems*, volume 30, 2017.

Chenyi Li, Yuan Zhang, Bo Wang, Guoqing Ma, Wei Tang, Haoyang Huang, and Nan Duan. SetPO: Set-level policy optimization for diversity-preserving LLM reasoning. In *Proceedings of the 43rd International Conference on Machine Learning*, 2026. URL https://openreview.net/forum?id=tYKGzVE1g7. arXiv:2602.01062.

Tianle Li, Wei-Lin Chiang, Evan Frick, Lisa Dunlap, Tianhao Wu, Banghua Zhu, Joseph E. Gonzalez, and Ion Stoica. From crowdsourced data to high-quality benchmarks: Arena-Hard and BenchBuilder pipeline. In *Proceedings of the 42nd International Conference on Machine Learning*, volume 267 of *Proceedings of Machine Learning Research*, pp. 34209–34231. PMLR, 2025a. URL https://proceedings.mlr.press/v267/li25h.html. arXiv:2406.11939.

Xiang Lisa Li, Farzaan Kaiyom, Evan Zheran Liu, Yifan Mai, Percy Liang, and Tatsunori Hashimoto. AutoBencher: Towards declarative benchmark construction. In *The Thirteenth International Conference on Learning Representations*, 2025b. URL https://arxiv.org/abs/2407.08351. arXiv:2407.08351.

Percy Liang, Rishi Bommasani, Tony Lee, Dimitris Tsipras, Dilara Soylu, Michihiro Yasunaga, et al. Holistic evaluation of language models. *Transactions on Machine Learning Research*, 2023. URL https://openreview.net/forum?id=iO4LZibEqW. arXiv:2211.09110.

D. V. Lindley. On a measure of the information provided by an experiment. *The Annals of Mathematical Statistics*, 27(4):986–1005, 1956. doi: 10.1214/aoms/1177728069.

Zhiyi Ma, Kawin Ethayarajh, Tristan Thrush, Somya Jain, Ledell Wu, Robin Jia, Christopher Potts, Adina Williams, and Douwe Kiela. Dynaboard: An evaluation-as-a-service platform for holistic next-generation benchmarking. In *Advances in Neural Information Processing Systems*, volume 34, pp. 10351–10367, 2021. URL https://proceedings.neurips.cc/paper/2021/hash/55b1927fdafef39c48e5b73b5d61ea60-Abstract.html.

Felipe Maia Polo, Lucas Weber, Leshem Choshen, Yuekai Sun, Gongjun Xu, and Mikhail Yurochkin. tinyBenchmarks: Evaluating LLMs with fewer examples. In *Proceedings of the 41st International Conference on Machine Learning*, volume 235 of *Proceedings of Machine Learning Research*, pp. 34303–34326. PMLR, 2024. URL https://proceedings.mlr.press/v235/maia-polo24a.html. arXiv:2402.14992.

Richard P. Mann and Dirk Helbing. Optimal incentives for collective intelligence. *Proceedings of the National Academy of Sciences*, 114(20):5077–5082, 2017. doi: 10.1073/pnas.1618722114. arXiv:1611.03899.

Mike A. Merrill, Alexander G. Shaw, Nicholas Carlini, et al. Terminal-Bench: Benchmarking agents on hard, realistic tasks in command line interfaces. In *The Fourteenth International Conference on Learning Representations*, 2026. URL https://openreview.net/forum?id=a7Qa4CcHak. arXiv:2601.11868.

Meta AI. Llama 3.2: Revolutionizing edge AI and vision with open, customizable models. Meta AI blog, 2024. URL https://ai.meta.com/blog/llama-3-2-connect-2024-vision-edge-mobile-devices/.

Shen-yun Miao, Chao-Chun Liang, and Keh-Yih Su. A diverse corpus for evaluating and developing English math word problem solvers. In *Proceedings of the 58th Annual Meeting of the Association for Computational Linguistics*, pp. 975–984, 2020. doi: 10.18653/v1/2020.acl-main.92.

Nolan Miller, Paul Resnick, and Richard Zeckhauser. Eliciting informative feedback: The peer-prediction method. *Management Science*, 51(9):1359–1373, 2005. doi: 10.1287/mnsc.1050.0379.

Iman Mirzadeh, Keivan Alizadeh, Hooman Shahrokhi, Oncel Tuzel, Samy Bengio, and Mehrdad Farajtabar. GSM-Symbolic: Understanding the limitations of mathematical reasoning in large language models. In *The Thirteenth International Conference on Learning Representations*, 2025. URL https://proceedings.iclr.cc/paper_files/paper/2025/hash/ec2e7a896f8250986b3907f57621ce94-Abstract-Conference.html. arXiv:2410.05229.

Dov Monderer and Lloyd S. Shapley. Potential games. *Games and Economic Behavior*, 14(1): 124–143, 1996. doi: 10.1006/game.1996.0044.

Sonia Krishna Murthy, Tomer Ullman, and Jennifer Hu. One fish, two fish, but not the whole sea: Alignment reduces language models' conceptual diversity. In *Proceedings of the 2025 Conference of the Nations of the Americas Chapter of the Association for Computational Linguistics (NAACL)*, pp. 11241–11258, 2025. doi: 10.18653/v1/2025.naacl-long.561. arXiv:2411.04427.

George L Nemhauser, Laurence A Wolsey, and Marshall L Fisher. An analysis of approximations for maximizing submodular set functions—I. *Mathematical Programming*, 14(1):265–294, 1978. doi: 10.1007/BF01588971.

Quan Nguyen and Adji Bousso Dieng. Quality-weighted Vendi scores and their application to diverse experimental design. In *Proceedings of the 41st International Conference on Machine Learning*, volume 235 of *Proceedings of Machine Learning Research*, pp. 37667–37682. PMLR, 2024. URL https://proceedings.mlr.press/v235/nguyen24d.html.

Yixin Nie, Adina Williams, Emily Dinan, Mohit Bansal, Jason Weston, and Douwe Kiela. Adversarial NLI: A new benchmark for natural language understanding. In *Proceedings of the 58th Annual Meeting of the Association for Computational Linguistics*, pp. 4885–4901. Association for Computational Linguistics, 2020. doi: 10.18653/v1/2020.acl-main.441. URL https://aclanthology.org/2020.acl-main.441/.

Vishakh Padmakumar and He He. Does writing with language models reduce content diversity? In *International Conference on Learning Representations (ICLR)*, 2024. arXiv:2309.05196.

Arkil Patel, Satwik Bhattamishra, and Navin Goyal. Are NLP models really able to solve simple math word problems? In *Proceedings of NAACL-HLT*, pp. 2080–2094, 2021. doi: 10.18653/v1/2021.naacl-main.168.

Juan Perdomo, Tijana Zrnic, Celestine Mendler-Dünner, and Moritz Hardt. Performative prediction. In *Proceedings of the 37th International Conference on Machine Learning*, volume 119 of *Proceedings of Machine Learning Research*, pp. 7599–7609. PMLR, 2020. URL https://proceedings.mlr.press/v119/perdomo20a.html.

Friedrich Pukelsheim. *Optimal Design of Experiments*, volume 50 of *Classics in Applied Mathematics*. SIAM, 2006. ISBN 9780898716047. doi: 10.1137/1.9780898719109. Originally published by Wiley, 1993.

Qwen Team, An Yang, et al. Qwen2.5 technical report. *arXiv preprint arXiv:2412.15115*, 2024. URL https://arxiv.org/abs/2412.15115.

Manish Raghavan. Competition and diversity in generative AI. *arXiv preprint arXiv:2412.08610*, 2024.

Tim Roughgarden. Intrinsic robustness of the price of anarchy. *Journal of the ACM*, 62(5):1–42, 2015. doi: 10.1145/2806883. Preliminary version in STOC 2009.

RSI-Exam Team. RSI-Exam: Benchmarking recursive self-improvement through executable research. Benchmark release, 2026. URL https://rsi-exam.ai/.

Lloyd S Shapley. A value for n-person games. In *Contributions to the Theory of Games II*, volume 28 of *Annals of Mathematics Studies*, pp. 307–317. Princeton University Press, 1953.

Ali Shirali, Rediet Abebe, and Moritz Hardt. A theory of dynamic benchmarks. In *The Eleventh International Conference on Learning Representations*, 2023. URL https://openreview.net/forum?id=i8L9qoeZOS. arXiv:2210.03165.

Victor Shnayder, Arpit Agarwal, Rafael Frongillo, and David C Parkes. Informed truthfulness in multi-task peer prediction. In *Proceedings of the 2016 ACM Conference on Economics and Computation (EC)*, pp. 179–196, 2016. doi: 10.1145/2940716.2940790.

Ilia Shumailov, Zakhar Shumaylov, Yiren Zhao, Nicolas Papernot, Ross Anderson, and Yarin Gal. AI models collapse when trained on recursively generated data. *Nature*, 631(8022):755–759, 2024. doi: 10.1038/s41586-024-07566-y.

Taylor Sorensen, Jared Moore, Jillian Fisher, Mitchell Gordon, Niloofar Mireshghallah, Christopher Michael Rytting, Andre Ye, Liwei Jiang, Ximing Lu, Nouha Dziri, Tim Althoff, and Yejin Choi. Position: A roadmap to pluralistic alignment. In *Proceedings of the 41st International Conference on Machine Learning*, volume 235 of *Proceedings of Machine Learning Research*, pp. 46280–46302. PMLR, 2024. URL https://proceedings.mlr.press/v235/sorensen24a.html. arXiv:2402.05070.

Niranjan Srinivas, Andreas Krause, Sham M Kakade, and Matthias Seeger. Gaussian process optimization in the bandit setting: No regret and experimental design. In *Proceedings of the 27th International Conference on Machine Learning (ICML)*, pp. 1015–1022, 2010. arXiv:0912.3995.

Marilyn Strathern. 'Improving ratings': audit in the British University system. *European Review*, 5 (3):305–321, 1997.

Christian Terwiesch and Yi Xu. Innovation contests, open innovation, and multiagent problem solving. *Management Science*, 54(9):1529–1543, 2008. doi: 10.1287/mnsc.1080.0884.

Taiki Todo, Atsushi Iwasaki, Makoto Yokoo, and Yuko Sakurai. Characterizing false-name-proof allocation rules in combinatorial auctions. In *Proceedings of the 8th International Conference on Autonomous Agents and Multiagent Systems (AAMAS)*, pp. 265–272, 2009.

Adrian Vetta. Nash equilibria in competitive societies, with applications to facility location, traffic routing and auctions. In *Proceedings of the 43rd Annual IEEE Symposium on Foundations of Computer Science (FOCS)*, pp. 416–425. IEEE, 2002. doi: 10.1109/SFCS.2002.1181966.

Colin White, Samuel Dooley, Manley Roberts, Arka Pal, Benjamin Feuer, Siddhartha Jain, Ravid Shwartz-Ziv, Neel Jain, Khalid Saifullah, Sreemanti Dey, Anjali Shubh-Agrawal, Sandeep Singh Sandha, Siddhartha Venkat Naidu, Chinmay Hegde, Yann LeCun, Tom Goldstein, Willie Neiswanger, and Micah Goldblum. LiveBench: A challenging, contamination-limited LLM benchmark. In *The Thirteenth International Conference on Learning Representations*, 2025. URL https://openreview.net/forum?id=sKYHBTaXVa. Spotlight; arXiv:2406.19314.

Fan Wu, Emily Black, and Varun Chandrasekaran. Generative monoculture in large language models. In *International Conference on Learning Representations (ICLR)*, 2025. arXiv:2407.02209.

Xinyi Xu, Zhaoxuan Wu, Chuan Sheng Foo, and Bryan Kian Hsiang Low. Validation free and replication robust volume-based data valuation. In *Advances in Neural Information Processing Systems*, volume 34, pp. 10837–10848, 2021.

Makoto Yokoo, Yuko Sakurai, and Shigeo Matsubara. The effect of false-name bids in combinatorial auctions: New fraud in internet auctions. *Games and Economic Behavior*, 46(1):174–188, 2004. doi: 10.1016/S0899-8256(03)00045-9.

Guanhua Zhang and Moritz Hardt. Inherent trade-offs between diversity and stability in multi-task benchmarks. In *Proceedings of the 41st International Conference on Machine Learning*, volume 235 of *Proceedings of Machine Learning Research*, pp. 58984–59002. PMLR, 2024. URL https://proceedings.mlr.press/v235/zhang24u.html. arXiv:2405.01719.

Hanlin Zhang, Jikai Jin, Vasilis Syrgkanis, and Sham Kakade. Prescriptive scaling reveals the evolution of language model capabilities. In *Proceedings of the 43rd International Conference on Machine Learning*, 2026. URL https://openreview.net/forum?id=IkjsHRpuYY. arXiv:2602.15327.

Pujun Zheng, Zixin Shang, Shufan Jiang, Wenhui Tian, Dongsheng Zhu, Zerun Ma, Dingbo Yuan, and Qi Zhang. SWE-Bench Pro Verified: A reliable benchmark for software engineering agents. *arXiv preprint arXiv:2609.08149*, 2026. URL https://arxiv.org/abs/2609.08149.

Kaijie Zhu, Jiaao Chen, Jindong Wang, Neil Zhenqiang Gong, Diyi Yang, and Xing Xie. DyVal: Dynamic evaluation of large language models for reasoning tasks. In *The Twelfth International Conference on Learning Representations*, 2024. URL https://openreview.net/forum?id=gjfOL9z5Xr. Spotlight; arXiv:2309.17167.

Appendices

Table of Contents

A Notation **18**

B Geometry and Coverage Proofs **18**
B.1 Diminishing Valid Coverage 19
B.2 Quality-Plus-Coverage Potential and Equilibrium Objective 20
B.3 Ordinal Potential under Monotone Utilities 21
B.4 Symmetric Orthogonal-Menu Separation 21
B.5 Quality-Perturbed Orthogonal Menus 22
B.6 Surrogate-to-Utility Transfer 23

C Axiomatic Characterization of the MOSAIC Score **24**

D Mechanism and Evaluation Protocol **27**
D.1 Population, Validity, and Feature Geometry 27
D.2 Active-Population Service Contract 28
D.3 Outcome Isolation and Statistical Units 29

E Proof of the Persistent-Reference Result **30**
E.1 Diversification under the Persistent Per-Prompt Reference 30
E.2 The Two Conditions, Measured 33

F Information Content and Accrual Under a Fixed Representation **34**
F.1 Valid Coverage Is Information about Model Quality 34

G Security and Manipulation Boundary **36**
G.1 Representation, Validity, and Ownership Boundary 36
G.2 Threat Model by Mechanism Component 38

H Experimental Details and Additional Results **38**
H.1 Every Reported Comparison, in Full 38
H.2 Representation and Gate Diagnostics 39
H.3 The Mechanism on Frozen Outputs 40
H.4 Frozen Asynchronous State-Transition Stress Test 40
H.5 Official-Release Replay and Later-Window Evaluation 40
H.6 Population-Reference Intervention 41
H.7 Shared-versus-Own Frozen-Candidate Factorial 41
H.8 Agent-Workflow Commit Intervention 42
H.9 Persistent-Reference Adaptation 43
H.10 Case-Study Protocols 44

I Extended Related Work **45**

A NOTATION

We fix the recurring symbols of the main results in Table 1 and introduce appendix-local variables where we first use them. A subscript H marks a quantity measured above the declared background.

Symbol	Meaning
Geometry (Section 2)	
$\mathcal{U}$, d	ground set of records; feature dimension
$\phi_i \in \mathbb{R}^d$	declared feature vector of record i; unit norm throughout
$w_i \geq 0$	declared validity/reliability weight (not the intrinsic quality b_i); $w_i = 0$ excludes the record
$\alpha > 0$	volume scale; $\alpha = 1$ in every experiment
$V(S)$	represented coverage of S, the regularized log determinant (1)
L_S	$\mathbf{I} + \alpha \sum_{j \in S} w_j \phi_j \phi_j^\top$, the reference's scatter
$\Delta_i(S)$	MOSAIC score: coverage record i adds to reference S
Active population (Sections 2.4 and 3)	
N, $\mathcal{A}_i$	number of developers; developer i's menu of admissible actions
$\mathcal{X}$, λ_x	prompt suite; prompt weights
H_x	background at prompt x: records in the reference that no developer owns
$A(a)$, $A_{-i}(a)$	population at profile a; population with entry i held out
$G_H(C)$	represented coverage of active set C above the background (Section 2.4)
$b_i(a_i)$, $u_i(a)$	declared intrinsic quality of developer i's menu item; payoff $b_i(a_i) + \omega s_i(a)$
$\Pi(a)$	population objective, quality plus coverage $\sum_i b_i(a_i) + \omega G_H(A(a))$ (Section 3.2); the exact potential of Theorem 2
$s_i(a)$	developer i's leave-one-out score (Section 2.4)
$\widetilde{s}_i(a_i)$, h_i	background-only score $G_H(\{a_i\})$, coverage above the background alone; strictly increasing utility map $u_i = h_i(s_i)$ in Proposition 3
b_k, ω, R_b, χ_α	intrinsic quality of mode k; weight on coverage; quality range $b_1 - b_d$; novelty premium $\log((1+\alpha)^2/(1+2\alpha))$
U_H	external utility the deployer wants; ϵ_i its gap to G_H, η_i the equilibrium slack
Persistent reference (Section 4)	
μ, s	mode map collapsing outputs into behavioral classes; a mode
$v_s \in [v_{\min}, v_{\max}]$	fixed weight of mode s
T, $m_s(T)$, $p_s(T)$	rounds; gated mass at mode s; its share
D_T, V_T	modes ever played; accrued coverage $\log\det L_{NT}$
$N_{\text{eff}}(T)$	Hill-2 effective mode count, the inverse Simpson index of $p_s(T)$ (6)
δ, γ	frontier richness; per-insertion rationality fraction
c_δ, $\bar{m}$	score floor for a fresh direction; saturation mass cap (Section 4)
Measured quantities (Section 5)	
Hill-1, Hill-2	effective mode counts (6): exp of Shannon entropy, and inverse Simpson, of the mode shares $p_s(T)$ (Hill, 1973)
task union	share of held-out tasks solved by at least one model in the population
d_z	paired Cohen's d; all intervals are paired Student-t

Table 1: **Notation for the main results.** Recurring symbols keep the same meaning in the main text and proofs.

B GEOMETRY AND COVERAGE PROOFS

Proposition 2 (Symmetric orthogonal-menu separation)**.** *Let $\alpha > 0$ and $d \geq N \geq 2$. Suppose every developer shares the same public menu $[d]$ and chooses $a_i \in [d]$, with modes represented by orthogonal unit vectors $e_1, \ldots, e_d$. All weights are one, the background is empty, and coverage is*

$$V(a) = \log\det\left(I + \alpha \sum_{i=1}^{N} e_{a_i} e_{a_i}^\top\right).$$

Under shared leave-one-out scores, the pure Nash profiles are exactly those with no repeated mode. Every equilibrium therefore maximizes coverage and attains $N\log(1+\alpha)$, so the shared

game's price of anarchy (the ratio between the worst equilibrium and the optimum) is one. Moreover, from any profile occupying K modes, strict best-response dynamics reach an optimal no-repeat profile in at most $N - K$ accepted duplicate-to-unused updates. Under standalone scoring $\widetilde{s}_i(a_i) = \log\det(I + \alpha e_{a_i} e_{a_i}^\top)$, every action has score $\log(1+\alpha)$; hence every profile is a pure Nash equilibrium, including complete collapse onto one mode. The standalone game's worst-equilibrium coverage ratio is exactly

$$\frac{\min_{a \in \mathrm{PNE}} V(a)}{\max_b V(b)} = \frac{\log(1+\alpha N)}{N \log(1+\alpha)},$$

which tends to zero as N grows for fixed α. The proof is given in Section B.4.

B.1 Diminishing Valid Coverage

Log-determinant coverage is the standard submodular design objective (Krause & Golovin, 2014; Kulesza & Taskar, 2012); we record the statement the body uses.

Lemma 1 (Diminishing returns of log-determinant coverage)**.** *For positive-semidefinite feature geometry and fixed nonnegative validity weights, the function V on finite subsets of $\mathcal{U}$ is normalized, monotone, and submodular. Hence $\Delta_i(A) \geq \Delta_i(B) \geq 0$ for finite $A \subseteq B \subseteq \mathcal{U}$ and $i \in \mathcal{U} \setminus B$.*

Proof. Let

$$L_S = \mathbf{I} + \alpha \sum_{i \in S} w_i \phi_i \phi_i^\top. \tag{8}$$

Each rank-one term is positive semidefinite because $w_i \geq 0$, so $L_S \succeq \mathbf{I} \succ 0$ and every inverse and log determinant below is well defined.

For monotonicity, take $A \subseteq B$ and write

$$L_B = L_A + \alpha \sum_{j \in B \setminus A} w_j \phi_j \phi_j^\top = L_A + C_{B \setminus A}, \qquad C_{B \setminus A} \succeq 0. \tag{9}$$

Congruence by $L_A^{-1/2}$ gives

$$L_A^{-1/2} L_B L_A^{-1/2} = \mathbf{I} + L_A^{-1/2} C_{B \setminus A} L_A^{-1/2} \succeq \mathbf{I}. \tag{10}$$

Every eigenvalue of the matrix on the right of (10) is at least one. Taking determinants and then logarithms therefore yields

$$V(B) - V(A) = \log\det\left(L_A^{-1/2} L_B L_A^{-1/2}\right) \geq 0. \tag{11}$$

Thus V is monotone, and $V(\varnothing) = \log\det \mathbf{I} = 0$ gives normalization.

For diminishing returns fix $i \notin S$. Factoring L_S out and applying the rank-one determinant lemma gives

$$\begin{aligned} V(S \cup \{i\}) - V(S) &= \log\det\left(L_S + \alpha w_i \phi_i \phi_i^\top\right) - \log\det L_S \\ &= \log\det\left(\mathbf{I} + \alpha w_i L_S^{-1/2} \phi_i \phi_i^\top L_S^{-1/2}\right) \\ &= \log\left(1 + \alpha w_i \left\| L_S^{-1/2} \phi_i \right\|_2^2\right) \\ &= \log\left(1 + \alpha w_i \phi_i^\top L_S^{-1} \phi_i\right). \end{aligned} \tag{12}$$

The third line uses the rank-one determinant identity

$$\det(\mathbf{I} + u u^\top) = 1 + u^\top u, \qquad u = \sqrt{\alpha w_i}\, L_S^{-1/2} \phi_i,$$

and (12) shows that each increment is nonnegative.

Now let $A \subseteq B$ and $i \notin B$. Equation (9) implies $L_A \preceq L_B$, and positive-definite order reverses under inversion:

$$L_B^{-1} \preceq L_A^{-1}. \tag{13}$$

Applying the same quadratic form to (13), multiplying by $\alpha w_i \geq 0$, adding one, and using that $\log$ is increasing gives

$$\begin{aligned}\Delta_i(B) &= \log\bigl(1 + \alpha w_i \phi_i^\top L_B^{-1} \phi_i\bigr) \\ &\leq \log\bigl(1 + \alpha w_i \phi_i^\top L_A^{-1} \phi_i\bigr) \\ &= \Delta_i(A).\end{aligned} \tag{14}$$

Equation (14) is the diminishing-returns definition of submodularity. □

Remark 1 is the $w_i = 0$ case of the same algebra: the rank-one term $\alpha w_i \phi_i \phi_i^\top$ is the zero matrix, so

$$L_{S\cup\{i\}} = L_S, \qquad \Delta_i(S) = 0, \qquad \Delta_j(S \cup \{i\}) = \Delta_j(S)$$

for every other candidate j: a gated record adds nothing to coverage and leaves every other score untouched.

B.2 Quality-Plus-Coverage Potential and Equilibrium Objective

Proof of Theorem 2. Fix developer i and a unilateral replacement $a_i \to a_i'$. The leave-one-out set A_{-i} is the same before and after the replacement. Subtracting the two current scores therefore cancels the common term:

$$\begin{aligned}s_i(a_i', a_{-i}) - s_i(a_i, a_{-i}) &= \bigl[G_H(A_{-i} \cup \{a_i'\}) - G_H(A_{-i})\bigr] - \bigl[G_H(A_{-i} \cup \{a_i\}) - G_H(A_{-i})\bigr] \\ &= G_H\bigl(A(a_i', a_{-i})\bigr) - G_H\bigl(A(a_i, a_{-i})\bigr).\end{aligned} \tag{15}$$

Adding $b_i(a_i') - b_i(a_i)$ to both sides and multiplying the coverage terms by ω gives

$$u_i(a_i', a_{-i}) - u_i(a_i, a_{-i}) = \Pi(a_i', a_{-i}) - \Pi(a_i, a_{-i}),$$

which is the exact-potential identity (4). Every accepted strict better response strictly increases $\Pi(a)$. Because the product of the finite menus contains only finitely many profiles, no strict-improvement path can be infinite. A maximal path ends at a profile in which no developer has a profitable unilateral replacement, that is, at a pure Nash profile.

For the half guarantee we extend Π to sets of entries. For a finite set C of entries (i, a) with $a \in \mathcal{A}_i$ put

$$\Pi(C) := \sum_{(i,a)\in C} b_i(a) + \omega\, G_H(C),$$

so that $\Pi(a) = \Pi(A(a))$ and, by the definition of s_i,

$$u_i(a) = \Pi\bigl(A(a)\bigr) - \Pi\bigl(A_{-i}(a)\bigr).$$

The first term is a nonnegative modular function of C and the second is normalized, monotone, and submodular, so Π is normalized, monotone, and submodular. Let $S = A(a)$ be the population of a pure Nash profile and let $O = A(a^\star) = \{o_1, \ldots, o_N\}$ maximize Π over feasible profiles. The comparison set $S \cup O$ may contain an incumbent and an alternative from the same slot; it is used only to bound the objective and need not itself be a feasible population. Monotonicity, submodularity, and then diminishing returns give

$$\begin{aligned}\Pi(O) &\leq \Pi(S \cup O) \\ &\leq \Pi(S) + \sum_{i=1}^{N} \bigl[\Pi(S \cup \{o_i\}) - \Pi(S)\bigr] \\ &\leq \Pi(S) + \sum_{i=1}^{N} \bigl[\Pi((S \setminus \{a_i\}) \cup \{o_i\}) - \Pi(S \setminus \{a_i\})\bigr].\end{aligned} \tag{16}$$

Each term in the final sum is developer i's payoff from deviating to o_i, and the Nash condition bounds it by the incumbent payoff,

$$\Pi((S \setminus \{a_i\}) \cup \{o_i\}) - \Pi(S \setminus \{a_i\}) = u_i(o_i, a_{-i}) \leq u_i(a) = \Pi(S) - \Pi(S \setminus \{a_i\}). \tag{17}$$

Finally, order the elements of S arbitrarily. Each removal gain on the right of (17) is at most the corresponding insertion gain against its prefix, by diminishing returns. The prefix gains telescope, so

$$\sum_{i=1}^{N}\big[\Pi(S) - \Pi(S \setminus \{a_i\})\big] \le \Pi(S). \tag{18}$$

Combining (16)–(18) gives $\Pi(O) \le 2\Pi(S)$, proving (5). At $b \equiv 0$ the objective is $\omega G_H(A(a))$ and the bound is the half-coverage guarantee. □

B.3 Ordinal Potential under Monotone Utilities

Proposition 3 (Ordinal potential under monotone utilities)**.** *In the epoch of Theorem 2, let each developer i have utility $u_i(a) = h_i(s_i(a))$ for a strictly increasing h_i. Then $G_H(A(a))$ is an ordinal potential: every unilateral replacement satisfies* $\operatorname{sign}[u_i(a'_i, a_{-i}) - u_i(a)] = \operatorname{sign}[G_H(A(a'_i, a_{-i})) - G_H(A(a))]$. *The pure Nash profiles are exactly those of the coverage game, every serialized strict-improvement path is finite, and (5) holds at $b \equiv 0$ (proof in Section B.3).*

Proof of Proposition 3. Fix developer i and a unilateral replacement $a_i \to a'_i$, and write

$$\Delta s := s_i(a'_i, a_{-i}) - s_i(a), \qquad \Delta G := G_H\big(A(a'_i, a_{-i})\big) - G_H\big(A(a)\big).$$

By (15), $\Delta s = \Delta G$. Since h_i is strictly increasing,

$$u_i(a'_i, a_{-i}) - u_i(a) = h_i\big(s_i(a) + \Delta s\big) - h_i\big(s_i(a)\big)$$

has the sign of Δs, hence of ΔG, which is the defining property of an ordinal potential (Monderer & Shapley, 1996).

A profile is a pure Nash profile exactly when no unilateral replacement has a positive payoff change; by the sign identity this is the condition that no unilateral replacement raises $G_H(A(a))$, which by (15) is also the pure Nash condition of the coverage game, so the two equilibrium sets coincide. The menus are finite, so along a serialized strict-improvement path $G_H(A(a))$ strictly increases through finitely many values and the path is finite; this is the finite improvement property that every finite ordinal potential game has (Monderer & Shapley, 1996). The bound (5) at $b \equiv 0$ is a statement about the pure Nash profiles of the coverage game, and since the two games have the same pure Nash profiles, it transfers unchanged. The affine payoff $u_i(a) = b_i(a_i) + \omega s_i(a)$ is not of the form $h_i(s_i(a))$, since $b_i(a_i)$ depends on the action and not on the score; Theorem 2 covers it directly. □

B.4 Symmetric Orthogonal-Menu Separation

Proof of Proposition 2. For a profile a, let

$$n_k(a) = \big|\{i : a_i = k\}\big|, \qquad k \in [d].$$

Orthogonality diagonalizes the coverage matrix, giving

$$V(a) = \sum_{k=1}^{d} \log\big(1 + \alpha n_k(a)\big). \tag{19}$$

If developer i occupies mode k, its shared leave-one-out score is therefore

$$s_i(a) = \log\left(\frac{1 + \alpha n_k(a)}{1 + \alpha(n_k(a) - 1)}\right). \tag{20}$$

First consider a profile with a repeated mode. Some k has $n_k(a) \ge 2$, so the profile occupies at most $N - 1$ modes. Since $d \ge N$, there is an unused mode ℓ. Moving any developer from k to ℓ changes its score from (20) to $\log(1 + \alpha)$, and the latter is strictly larger because

$$(1 + \alpha)\big(1 + \alpha(n_k(a) - 1)\big) - (1 + \alpha n_k(a)) = \alpha^2(n_k(a) - 1) > 0.$$

Thus no profile with a repeated mode is a pure Nash equilibrium.

The same argument gives the finite best-response bound. If a profile occupies $K < N$ modes, at least one developer occupies a repeated mode and at least one mode is unused. For any developer on a repeated mode, every unused mode gives score $\log(1+\alpha)$, while every mode occupied in its leave-one-out reference gives a strictly smaller score by the calculation above. A strict best response therefore moves the developer from a repeated mode to an unused one. Each accepted update increases the number of occupied modes by one, so after at most $N-K$ accepted updates the profile occupies N modes and is optimal.

Conversely, suppose all occupied modes are distinct. Every incumbent score equals $\log(1+\alpha)$. A deviation to an unused mode preserves this score. A deviation to a mode occupied by $m \geq 1$ other developers yields

$$\log\left(\frac{1+\alpha(m+1)}{1+\alpha m}\right) < \log(1+\alpha),$$

where strictness follows from

$$(1+\alpha)(1+\alpha m) - (1+\alpha(m+1)) = \alpha^2 m > 0.$$

Hence no unilateral deviation is profitable, and the profile is a pure Nash equilibrium. The shared-game equilibria are therefore exactly the no-repeat profiles.

They are also precisely the coverage maximizers. If $n_k(a) \geq 2$ and $n_\ell(a) = 0$, moving one occupant from k to ℓ changes the two affected terms in (19) by

$$\log\left(\frac{(1+\alpha(n_k(a)-1))(1+\alpha)}{1+\alpha n_k(a)}\right) > 0,$$

again because the numerator exceeds the denominator by $\alpha^2(n_k(a)-1)$. Repeating this operation eliminates every duplicate and yields $V(a) = N\log(1+\alpha)$. Since every shared-game equilibrium attains this optimal value, the shared game's price of anarchy is one.

Under standalone scoring, the score of any action is instead

$$\widetilde{s}_i(k) = \log\det(I + \alpha e_k e_k^\top) = \log(1+\alpha),$$

independently of both k and the other developers' choices. Every profile is therefore a pure Nash equilibrium. In particular, the fully collapsed profile has one occupancy equal to N and all others zero, so (19) gives coverage $\log(1+\alpha N)$.

This collapsed value is the minimum coverage of any profile. Indeed,

$$\exp V(a) = \prod_{k:n_k(a)>0} \big(1+\alpha n_k(a)\big) \geq 1 + \alpha \sum_{k=1}^{d} n_k(a) = 1+\alpha N,$$

because all omitted product terms are nonnegative. Since every standalone profile is an equilibrium and no-repeat profiles attain the common optimum $N\log(1+\alpha)$, the standalone game's worst-equilibrium coverage ratio is exactly

$$\frac{\log(1+\alpha N)}{N\log(1+\alpha)}. \tag{21}$$

For fixed $\alpha > 0$, the numerator is $O(\log N)$ and the denominator is $\Theta(N)$, so this ratio tends to zero as $N \to \infty$. □

B.5 Quality-Perturbed Orthogonal Menus

The payoffs $u_i(a) = b_{a_i} + \omega s_i(a)$ and $\widetilde{u}_i(a) = b_{a_i}$, the quality range $R_b = b_1 - b_d$ and the novelty premium $\chi_\alpha = \log\big((1+\alpha)^2/(1+2\alpha)\big) > 0$ are those of Section 3.1; the orthogonal-menu conditions are those of Proposition 2.

Proof of Theorem 1. Let

$$m_\alpha(n) := \log\frac{1+\alpha n}{1+\alpha(n-1)}$$

be the coverage added by the nth occupant of one mode. This quantity decreases in n, and the gap between the coverage added by the first occupant of a mode and by the second is exactly the novelty premium,

$$m_\alpha(1) - m_\alpha(2) = \chi_\alpha.$$

If a developer shares mode k with $n-1 \geq 1$ others and mode ℓ is unused, moving to ℓ changes its payoff by

$$b_\ell - b_k + \omega\big[m_\alpha(1) - m_\alpha(n)\big] \geq -R_b + \omega\chi_\alpha > 0.$$

No MOSAIC equilibrium can therefore repeat a mode.

A no-repeat profile that omits one of the top N modes assigns some developer to a lower-quality mode. Moving that developer to the omitted higher-quality mode leaves its coverage at $m_\alpha(1)$ and strictly raises intrinsic quality. Conversely, consider a no-repeat allocation of the top N modes. A move to an unused mode lowers intrinsic quality and leaves the coverage unchanged. A move onto an occupied mode changes payoff by at most $R_b - \omega\chi_\alpha < 0$. The no-repeat allocations of the top N modes, and only these, are pure Nash.

The augmented objective $\sum_i b_{a_i} + \omega V(a)$ is the population objective Π of Section 3.2 on this menu, an exact potential by Theorem 2. Any duplicate-to-unused move raises it when $R_b < \omega\chi_\alpha$, and among no-repeat profiles the allocation of the top N modes maximizes it. Under standalone scoring, strict ordering makes mode 1 every developer's unique best response, so full collapse is the unique equilibrium. Finally, for $N = d = 2$, a developer in the collapsed profile gains $-R_b + \omega\chi_\alpha$ by moving from mode 1 to mode 2. The collapsed profile is therefore a MOSAIC equilibrium exactly when $R_b \geq \omega\chi_\alpha$. □

B.6 Surrogate-to-Utility Transfer

Let U_H be a normalized, monotone, submodular external utility over active actions; the argument below extends the valid-utility-game bound of Vetta (2002). For $F \in \{G_H, U_H\}$ define the gain from replacing entry i by a_i' as

$$\Delta_i^F(a_i'; a) := F\big(A_{-i}(a) \cup \{a_i'\}\big) - F\big(A(a)\big). \tag{22}$$

Theorem 3 (Surrogate-to-utility transfer)**.** *Suppose a profile a is a $(\eta_1, \ldots, \eta_N)$-approximate MOSAIC equilibrium,*

$$\Delta_i^{G_H}(a_i'; a) \leq \eta_i \quad \textit{for every } i \textit{ and } a_i' \in \mathcal{A}_i,$$

and the two objectives agree on the menus to within a surrogate gap *ϵ_i:*

$$\left|\Delta_i^{U_H}(a_i'; a) - \Delta_i^{G_H}(a_i'; a)\right| \leq \epsilon_i \quad \textit{for every } i \textit{ and } a_i' \in \mathcal{A}_i. \tag{23}$$

Then

$$U_H(A(a)) \geq \frac{1}{2}\left[\max_{b \in \prod_i \mathcal{A}_i} U_H(A(b)) - \sum_{i=1}^N (\eta_i + \epsilon_i)\right]. \tag{24}$$

In particular, an exact MOSAIC equilibrium with uniform surrogate gap ϵ achieves at least $\frac{1}{2}U_H(\mathrm{OPT}) - \frac{1}{2}N\epsilon$.

Proof. Write $S = A(a)$ and let $O = A(b^\star) = \{o_1, \ldots, o_N\}$ maximize U_H over feasible profiles. The approximate MOSAIC condition and menu alignment imply, for each optimal replacement o_i,

$$U_H\big((S \setminus \{a_i\}) \cup \{o_i\}\big) - U_H(S) = \Delta_i^{U_H}(o_i; a) \leq \eta_i + \epsilon_i. \tag{25}$$

Thus a profile that is approximately stable under represented coverage is also approximately stable under the external utility, with additive error $\eta_i + \epsilon_i$ for slot i.

Normalization, monotonicity, and submodularity of U_H give the same comparison-set bound as (16):

$$U_H(O) \leq U_H(S) + \sum_{i=1}^N \left[U_H\big((S \setminus \{a_i\}) \cup \{o_i\}\big) - U_H(S \setminus \{a_i\})\right]. \tag{26}$$

For every summand, insert and subtract $U_H(S)$ and apply (25):

$$\begin{aligned} U_H\big((S \setminus \{a_i\}) \cup \{o_i\}\big) - U_H(S \setminus \{a_i\}) \\ \leq \eta_i + \epsilon_i + U_H(S) - U_H(S \setminus \{a_i\}). \end{aligned} \tag{27}$$

The removal gains of any normalized submodular function obey

$$\sum_{i=1}^{N} [U_H(S) - U_H(S \setminus \{a_i\})] \leq U_H(S), \tag{28}$$

which follows by ordering the elements of S, bounding each removal gain by its insertion gain against the corresponding prefix, and telescoping. Combining (26)–(28) yields

$$U_H(O) \leq 2U_H(S) + \sum_{i=1}^{N} (\eta_i + \epsilon_i), \tag{29}$$

which rearranges to the stated external-utility bound. □

The transfer bound depends on the surrogate gap: the theorem assumes the gap is bounded on the feasible menus, while in our workflow studies we score the selected endpoint with independent hidden or verified utility.

C Axiomatic Characterization of the MOSAIC Score

Definition 1 belongs to a broader class: every monotone submodular set function on records induces an increment that discounts duplicates (Krause & Golovin, 2014). Three hypotheses on a population-relative score reduce that class to the log determinant up to its two constants (Theorem 4): increment accounting, isotropic additive coverage and an additive congestion law. Dropping the congestion law and asking only that the copies of a replicated record be paid in total a finite, nonzero multiple of one honest entry still forces logarithmic growth (Corollary 2).

Setting. A population-relative score pays $s(i \mid S) \in \mathbb{R}$ to a record i against a finite reference S of admitted records, each with a unit feature $\phi \in \mathbb{R}^d$ and a gate weight $w \geq 0$. Write $\Sigma(S) = \sum_{j \in S} w_j \phi_j \phi_j^\top$ for the reference scatter and $\lambda_1, \ldots, \lambda_d$ for its eigenvalues.

Definition 2 (Payment axioms)**.** *The score satisfies:*

(P1) Anonymity. $s(i \mid S)$ *depends on* S *only through the multiset* $\{(w_j, \phi_j)\}_{j \in S}$.

(P2) Path independence. *Admitting a finite set in any order yields the same total payment; equivalently there is* V *with* $V(\varnothing) = 0$ *and* $s(i \mid S) = V(S \cup \{i\}) - V(S)$. *This is the discrete analogue of the path independence that defines cost-function market makers (Hanson, 2003; Abernethy et al., 2013); for set functions the equivalence follows by telescoping.*

(P3) Gate nullity. *A record with* $w = 0$ *is paid nothing and leaves* V *unchanged.*

(P4) Scatter sufficiency and isotropy. $V(S) = \Phi(\Sigma(S))$ *for some* Φ *with* $\Phi(U\Sigma U^\top) = \Phi(\Sigma)$ *at every orthogonal* U.

(P5) Orthogonal additivity. *If every feature of* S_1 *is orthogonal to every feature of* S_2, *then* $V(S_1 \cup S_2) = V(S_1) + V(S_2)$.

Throughout, V *is monotone and submodular, which is the formal content of paying a duplicate less than the record it duplicates.*

Of the five, (P4) is the only substantive restriction: it identifies weight with feature scale, so a record of weight w at unit ϕ is treated as one of weight 1 at $\sqrt{w}\,\phi$. The other four are bookkeeping requirements on the payment stream. We write $g(\lambda)$ for the value V assigns to a single record of weight λ at a unit feature, and g'_+ for its right derivative.

Theorem 4 (Characterization of the log-determinant score)**.** *Suppose a population-relative score satisfies*

(i) increment accounting*: (P1)–(P3);*

(ii) isotropic additive coverage*: (P4)–(P5), with V monotone and submodular;*

(iii) additive congestion*: the reference acts on a candidate only through a scalar congestion $\tau \geq 0$ that is nondecreasing and nonzero, vanishes on the empty reference, and is additive in admitted mass along a direction, so that the increment at incumbent mass m in a direction is $g(m+w) - g(m) = \Phi_0\big(w/(1+\tau(m))\big)$ for some Φ_0 with a finite positive right derivative at zero.*

Then there is $\alpha > 0$ such that, for every finite reference S,

$$V(S) = \frac{g'_+(0)}{\alpha} \log\det\big(\mathbf{I} + \alpha\Sigma(S)\big). \tag{30}$$

Normalizing $g'_+(0) = \alpha$ gives (1), whose increment is Definition 1.

Proof. Step 1 (spectral form). (P2) yields V and (P4) gives $V(S) = \Phi(\Sigma(S))$ with Φ a function of the spectrum alone. Fix a positive semidefinite Σ with decomposition $\Sigma = \sum_r \lambda_r u_r u_r^\top$ and realize it by the d records (λ_r, u_r), whose features are pairwise orthogonal. Applying (P5) $d-1$ times gives

$$\Phi(\Sigma) = \sum_{r=1}^{d} \Phi\big(\lambda_r u_r u_r^\top\big).$$

A single record has spectrum $\{\lambda, 0, \ldots, 0\}$, so by (P4) its value is a function of λ alone, namely $g(\lambda)$, and

$$V(S) = \sum_{r=1}^{d} g\big(\lambda_r(\Sigma(S))\big). \tag{31}$$

(P3) with $V(\varnothing) = 0$ gives $g(0) = 0$, and monotonicity of V gives g nondecreasing. Restricting submodularity to a single direction gives, for $m \leq m'$ and $w \geq 0$,

$$g(m+w) - g(m) \;\geq\; g(m'+w) - g(m'),$$

which is concavity of g. Every rule obeying (i)–(ii) is therefore indexed by one nondecreasing concave scalar function; rank rewards, fixed prize pools and Elo-type updates are excluded at (P2), and rules privileging a coordinate direction at (P4).

Step 2 (exact form). Additivity and monotonicity of τ give $\tau(m) = \alpha m$ with $\alpha > 0$ by Cauchy's equation, so the increment at incumbent mass m is $g(m+w) - g(m) = \Phi_0\big(w/(1+\alpha m)\big)$. Dividing by w and letting $w \downarrow 0$ yields

$$g'_+(m) = \frac{\Phi'_{0,+}(0)}{1+\alpha m},$$

and evaluating at $m = 0$ identifies $\Phi'_{0,+}(0) = g'_+(0)$. Integrating from 0 with $g(0) = 0$ gives

$$g(\lambda) = \frac{g'_+(0)}{\alpha} \log(1+\alpha\lambda),$$

and substituting into (31) and summing over the spectrum gives (30). □

The congestion law is the hypothesis that selects the exact form. Without it, the class (i)–(ii) is still pinned down asymptotically by the total payment to a record's copies.

Definition 3 (Replication gain)**.** *For w with $g(w) > 0$, an owner submits n copies of one record (w, ϕ) into an empty reference. The owner's total leave-one-out payment, relative to the payment to one record in an unoccupied direction, is*

$$\beta(n) = \frac{n\big[g(nw) - g((n-1)w)\big]}{g(w)}, \qquad \beta(1) = 1. \tag{32}$$

The normalization is the comparison an owner actually faces: splitting one capability across n entries is profitable exactly when the n entries are paid more in total than one entry on an uncovered direction.

Corollary 2 (Growth class without the congestion law)**.** *Under (i)–(ii) alone, let* $L = \lim_{\lambda\to\infty} \lambda g'_+(\lambda) \in [0,\infty]$. *Then* $\lim_{n\to\infty} \beta(n) = L/g(w)$. *Consequently the replication gain is finite and nonzero if and only if* $L \in (0,\infty)$, *in which case* $g(\lambda) = L\log\lambda + o(\log\lambda)$.

Proof. The n copies are exchangeable, so each is paid $g(nw) - g((n-1)w)$ and the owner's total is n times that. Concavity, from Step 1 above, places this finite difference between the one-sided derivatives at its endpoints,

$$w\, g'_+(nw) \;\le\; g(nw) - g\big((n-1)w\big) \;\le\; w\, g'_+\big((n-1)w\big).$$

After multiplication by n, both bounds converge to L because the two endpoints are asymptotically equal; division by $g(w)$ gives the first claim. When $L \in (0,\infty)$, integrate the eventual bounds

$$\frac{L-\epsilon}{t} \;\le\; g'_+(t) \;\le\; \frac{L+\epsilon}{t}$$

and send $\epsilon \downarrow 0$ to get $g(\lambda)/\log\lambda \to L$. □

The three regimes are exhaustive whenever L exists. Superlogarithmic g, such as λ^p with $p \in (0,1)$, has $L = \infty$: replication is unboundedly profitable. Bounded g has $L = 0$ and zero asymptotic replication gain, but it also caps V, so accrued coverage stops growing at a finite horizon. The logarithmic class is the only one with finite nonzero asymptotic replication gain and unbounded accrued coverage.

Corollary 3 (The score and its constant)**.** *For the score of Theorem 4 normalized to* $g'_+(0) = \alpha$, *the replication gain satisfies*

$$\lim_{n\to\infty} \beta(n)\cdot\log(1+\alpha w) = 1. \tag{33}$$

For finite n *the exact gain is*

$$\beta(n) = \frac{n\log\big((1+\alpha nw)/(1+\alpha(n-1)w)\big)}{\log(1+\alpha w)}. \tag{34}$$

At unit weight the asymptotic gain is at most one exactly when $\alpha \ge e-1$. *By contrast, we run the experiments at* $\alpha = 1$, *where* $1 < \beta(n) < 1/\log 2$ *for every* $n \ge 2$. *Exact replication is profitable, and its aggregate gain approaches* $1/\log 2$.

Proof. Here $g(\lambda) = \log(1+\alpha\lambda)$ has

$$\lambda g'(\lambda) = \frac{\alpha\lambda}{1+\alpha\lambda} \longrightarrow 1 \qquad (\lambda\to\infty),$$

so $L = 1$ in Corollary 2 and

$$\lim_{n\to\infty}\beta(n) = \frac{1}{g(w)} = \frac{1}{\log(1+\alpha w)},$$

which is (33). Substituting g into (32) gives (34).

At $w = 1$ the asymptotic gain is at most one exactly when $\log(1+\alpha) \ge 1$, that is when $\alpha \ge e-1$. At $\alpha = w = 1$ the finite gain is

$$\beta(n) = \frac{n\log(1+1/n)}{\log 2} = \frac{\log\big((1+1/n)^n\big)}{\log 2},$$

and for $n \ge 2$ the binomial theorem and the standard exponential bound give

$$2 < \left(1+\frac{1}{n}\right)^n < e.$$

Taking logarithms yields $1 < \beta(n) < 1/\log 2$. □

If we normalize instead to $g'_+(0) = \alpha/2$, V is the D-optimal criterion of classical experiment design (Lindley, 1956; Chaloner & Verdinelli, 1995) and payments are in nats of information about model quality; the characterization holds under either normalization.

Equation (33) shows that the ridge constant controls the replication incentive: the asymptotic aggregate payment to unit-weight exact copies crosses the honest payment at $\alpha = e-1$. Representation-level splits into separated directions obey a different bound, Proposition 6. Sufficiency is the routine direction and follows from Lemma 1 together with Proposition 5.

Object	Role in the protocol	Failure if omitted
Prompt and response	Defines the unit whose contribution is measured	Cross-prompt repetition is invisible to a per-response score
Validity gate w	Removes known-invalid mass before geometry is evaluated	Fluent invalid padding can appear diverse
Feature map ϕ	Determines which response differences count as directions	Paraphrase or semantic splitting can be miscounted
Reference set	Makes a score conditional on the coverage the reference already represents	A static reward keeps awarding full credit to outputs in saturated directions
External endpoint	Evaluates usefulness separately from the selection objective	The mechanism can reward a proxy without evidence of adaptation value

Table 2: **Protocol objects and their roles.** We test each component separately because each admits a different failure.

D Mechanism and Evaluation Protocol

D.1 Population, Validity, and Feature Geometry

The service logs each scoring event so a third party can recheck it: the prompt, sampled response, gate inputs and resulting nonnegative weight $w(x, y)$, feature vector $\phi(x, y)$, epoch, verified slot, population version, incumbent, proposal, computed score, and commit decision. A response can have high absolute quality and still receive a small live contribution score when a nearby direction is already represented. We therefore report absolute quality separately from reference-conditional reward.

Remark 1 (Zero weight)**.** *If $w_i = 0$ then $V(S \cup \{i\}) = V(S)$ and no other candidate's score changes, so an output rejected by the validity signal cannot create coverage. The weight is a protocol input: correctness or other validity criteria come from a task oracle, checker, or calibrated judge, and MOSAIC reweights the outputs they admit.*

Validity weight. We apply the validity weight to each item before computing coverage. On verifiable tasks it comes from a binary task oracle or a calibrated soft checker. On open-ended tasks it may come from an LLM-judge validity gate. MOSAIC neither estimates nor validates that signal. An item assigned zero mass contributes no rank-one scatter term, so it can inflate neither its own margin nor the apparent novelty of other items. However, a false positive still adds its rank-one scatter term as a valid item would, which is why calibration and adversarial probes remain necessary.

Standing population. The standing population has a fixed background H and a current active population $A(a)$. The background changes only at an explicit epoch boundary. The population is live: member i is paid against every other current member,

$$s_i(a) = G_H(A(a)) - G_H(A_{-i}(a)), \tag{35}$$

which is the leave-one-out score s_i of Section 2.4. This is the convention under which the finite game of Theorem 2 has potential $G_H(A(a))$.

Which convention the evaluator adopts determines whether same-round redundancy is credited. Under the alternative convention, in which each member is scored against the fixed background alone, $\widetilde{s}_i(a_i) = G_H(\{a_i\})$, the game remains an exact potential game, but its potential is the separable $\sum_i G_H(\{a_i\})$, so two identical outputs submitted in the same round are both paid in full and same-round redundancy is not internalized. Any deployment reporting a MOSAIC score should say which of (35) and the history-only rule it evaluates, because the population objective the mechanism aligns with differs between them. Table 2 lists the five protocol objects and the failure each one guards against. We specify each object in turn in the rest of this subsection and in the subsections that follow.

Feature and kernel representations. Let $\mathbf{\Phi}$ contain feature vectors as columns and let W be the diagonal matrix of nonnegative weights. The weighted feature scatter and response-space Gram forms are equivalent by Sylvester's determinant identity,

$$\det\left(\mathbf{I} + \alpha \mathbf{\Phi} W \mathbf{\Phi}^\top\right) = \det\left(\mathbf{I} + \alpha W^{1/2} K W^{1/2}\right), \qquad K = \mathbf{\Phi}^\top \mathbf{\Phi}. \tag{36}$$

The conjugation by $W^{1/2}$ keeps the kernel-side matrix symmetric positive semidefinite; the one-sided WK has the same determinant but is not symmetric. A directly specified similarity matrix has this feature-space interpretation only when it is positive semidefinite, which is the L-ensemble condition of determinantal point processes (Kulesza & Taskar, 2012). We retain a positive ridge because it makes every matrix invertible and prevents near-singularity from becoming a numerical claim of infinite novelty.

For the primary arithmetic geometry we use a direct normalized feature map of operator unigrams, operator bigrams, and chain length. Its empirical scatter is positive semidefinite by construction. We treat generic text embeddings only as secondary diagnostics; the fixed comparator is the sentence encoder of Section H.2.

D.2 Active-Population Service Contract

At the beginning of an epoch, the evaluator freezes the prompt suite, prompt weights, background records, gate, representation, slot population, and admissible action menus. Each action is a uniquely identified fixed bundle of prompt-level records. The service stores one current action per verified slot and a monotonically increasing population version.

For a proposal from slot i, the server performs one atomic transition:

1. read the current population version and form the same leave-one-out reference A_{-i} for the incumbent and proposal;
2. compute both live contributions against that reference;
3. accept only if the proposal is a strict improvement under the published numeric tolerance $\varepsilon_{\mathrm{adm}}$ and deterministic tie rule;
4. compare-and-swap the slot against the version read in step 1; if the version is stale, return to step 1 and rescore before committing;
5. after a successful commit, increment the population version and recompute affected displayed contribution scores.

Theorem 2 assumes exact comparison, equivalently $\varepsilon_{\mathrm{adm}} = 0$. A positive tolerance generally yields only a tolerance equilibrium unless no gain lies in $(0, \varepsilon_{\mathrm{adm}}]$; in the primary finite games we checked that none does.

This contract permits independent proposal preparation but not simultaneous accepted writes. It also prevents a proposal from being accepted on a stale high margin after another slot has committed an action in the same direction. Every accepted update increases the exact potential, but convergence of an operating service additionally needs fair opportunities and responsive developers as stated in Theorem 2. Figure 3 shows the resulting state graph under fresh and stale admission.

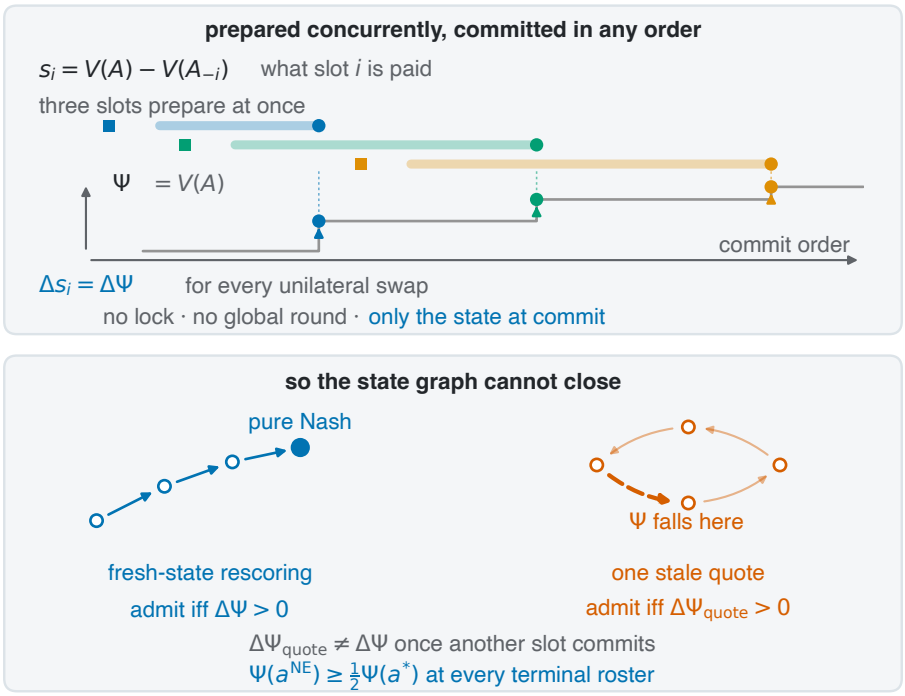


Figure 3: **Fresh-state admission implements ascent in the geometric potential.** Concurrent proposals may be prepared independently, but each accepted replacement is rescored against the population it enters. Every fresh-state edge then increases the potential; a stale quote can decrease it and return the process to a population it already left. Counts and replay details appear in Section H.5.

D.3 Outcome Isolation and Statistical Units

The external SVAMP/ASDiv questions are public inputs needed for response generation, but we withhold their answer keys and every derived correctness field from the mechanism and profile-lock stages. We lock the final action identifiers before an isolated scorer joins them to response artifacts and private answers. The MOSAIC representation, potential, and internal coverage never use the external endpoint. Here "private" means execution-isolated from the selector until locking, not that the benchmark sources or gold labels are unavailable publicly.

Within an experimental cell we match prompts, generation budgets, and training budgets across all controlled comparisons. We pair effects within the unit each result states, report paired Student-t intervals at 95% except where a paired task bootstrap is named, and generate seeds deterministically from a base seed. For adaptation results we report the per-round curves, so the reader can distinguish a short-lived quality shift from a persistent population shift.

An independently seeded four-slot population is the scientific unit within each panel. Developers, prompts, update opportunities, and schedule replays are not additional replicates. The primary effect is the population-level current-minus-launch difference in source-balanced capability-union exact match; we give each panel its own paired Student-t interval and decision. For mean per-slot exact match we test noninferiority against the frozen -0.02 margin. No secondary representation, schedule, batch size, or source-specific endpoint can rescue a failed primary endpoint or integrity gate.

In the two frozen-output panels we sample generation seeds conditional on one cached model, the fixed item portfolio, and fixed action recipes; in the heterogeneous panel we condition on four declared checkpoints. In neither design do we sample developers or model families.

E Proof of the Persistent-Reference Result

The theorem needs one condition on what developers can still do and one on what they choose. The first says that every developer still has a fresh direction available: at every history, each developer can still produce an admitted output whose feature mass lies at least δ outside the span of the current reference. For developer a and a finite pre-action reference history h, let $\mathcal{M}_a(h)$ be the gate-admitted modes that developer can produce at h.

Definition 4 (Developer-specific rich frontier)**.** *For $\delta > 0$, the action correspondence has a* uniformly δ-rich frontier *if, for every developer a and every finite protocol-reachable history h immediately before a acts, there exists $u \in \mathcal{M}_a(h)$ such that $\phi_u^\top L_h^{-1} \phi_u \geq \delta$, where L_h is the reference at h: at least a fraction δ of the output's feature mass is left unexplained after whitening by the reference. Mass of at least δ outside the span suffices, since with $W(h)$ the span of the reference, $\phi_u^\top L_h^{-1} \phi_u \geq \|P_{W(h)^\perp} \phi_u\|^2$. A realized path is* pathwise δ-rich *when this requirement is imposed only at the pre-action histories that occur on that path, which is the weaker property the theorem uses.*

Definition 5 (Pathwise γ-rationality)**.** *A realized developer profile is* pathwise γ-rational*, $\gamma \in (0, 1]$, if at every insertion the chosen output earns at least γ times the supremum score available in the acting developer's own action set against the current reference. This is a per-insertion approximate best response, not an expectation or population-equilibrium condition.*

Corollary 4 (Static-score comparison)**.** *Specialize the theorem's setting to a fixed shared action set $\mathcal{M}$ whose induced correspondence is uniformly δ-rich. In both arms keep the same prompt, gate, N developers, fixed public choice-and-insertion order, initially empty accumulating reference, mode features, validity weights, and the same coverage functional V_T. In the comparison arm, change only the reward to a standalone score $q_s > 0$. The reference is still updated after every insertion, but it does not enter q and is used only to compute V_T. If q has a unique best mode $s^\star$ and the realized profile is pathwise γ-rational with respect to q for $\gamma > \sup_{s \neq s^\star} q_s / q_{s^\star}$, then every developer submits $s^\star$. For every T,*

$$V_T = \log(1 + \alpha v_{s^\star} NT), \tag{37}$$

and hence $V_T = \Theta(\log T)$ as $T \to \infty$.

The corollary isolates what the reference is doing. Under the identical insertion process, a score independent of the population concentrates every submission on its best mode $s^\star$, and coverage grows only logarithmically. The horizon enters only inside the logarithm, so each added round contributes less coverage than the one before it. The linear-against-logarithmic gap over the same insertions is therefore attributable to the reference alone.

E.1 Diversification under the Persistent Per-Prompt Reference

We prove Proposition 1 and Corollary 4 for the persistent sequential protocol stated with those results. Fix one prompt and suppress its superscript. We represent outputs in a real Hilbert space and start the analysis after the validity gate: at a pre-action history h, developer a has its own gate-admitted action set $\mathcal{M}_a(h)$ of unit modes ϕ_s with validity $v_s \in [v_{\min}, v_{\max}]$, $v_{\min} > 0$. In each round, each of N developers submits once. In the fixed public order, the mechanism processes one submission, assigns its score, updates that prompt's reference, and only then lets the next developer act.

For each serialized insertion index k, let L_k be the reference after k insertions (Section 4), let $m_s(k)$ be the gated mass accumulated at mode s, and let W_k be the span of the admitted directions. After T rounds, we use the shorthand $L_T := L_{NT}$, $m_s(T) := m_s(NT)$, and $W_T := W_{NT}$. Each history is finite, so $L_T = \mathbf{I} + K_T$ with finite-rank, trace-class K_T. Its Fredholm determinant equals the ordinary determinant on the finite-dimensional history span W_T; throughout this appendix $\log\det L_T$ denotes that quantity and we write $V(L_T) := \log\det L_T$. The insertion payment for mode s against the current operator L is

$$\Delta(s \mid L) = \log\big(1 + \alpha v_s \, \phi_s^\top L^{-1} \phi_s\big) . \tag{38}$$

The fixed order and immediate update make the realized payments telescope,

$$\sum_{k \leq NT} \Delta_k = \log\det L_T - \log\det L_0 = V(L_T). \tag{39}$$

Along the realized path, the action correspondences and choices satisfy Definitions 4 and 5. When the payment supremum is attained, exact best response is the case $\gamma = 1$; without attainment, that endpoint condition may be infeasible. The proposition uses no additional behavioral assumption.

Lemma 2 (Payment decay on covered directions)**.** *For every mode s and every serialized insertion index k,*

$$\phi_s^\top L_k^{-1}\phi_s \;\le\; \min\left\{1,\; \frac{1}{\alpha m_s(k)}\right\}, \tag{40}$$

where $1/0 := +\infty$, and the map $k \mapsto \phi_s^\top L_k^{-1}\phi_s$ is nonincreasing.

Proof. Split the reference into mode s and everything else,

$$L_k = B + \alpha m_s(k)\phi_s\phi_s^\top, \qquad B = \mathbf{I} + \alpha \sum_{u\neq s} m_u(k)\phi_u\phi_u^\top \succeq \mathbf{I}. \tag{41}$$

The Sherman–Morrison identity applied to (41) gives

$$\phi_s^\top L_k^{-1}\phi_s = \frac{b}{1+\alpha m_s(k)\, b}, \qquad b = \phi_s^\top B^{-1}\phi_s \in (0,1]. \tag{42}$$

The right-hand side of (42) is increasing in b, so bounding it once by $b \le 1$ and once by dropping the 1 in the denominator yields

$$\phi_s^\top L_k^{-1}\phi_s \;\le\; \min\left\{b,\; \frac{1}{\alpha m_s(k)}\right\} \;\le\; \min\left\{1,\; \frac{1}{\alpha m_s(k)}\right\}, \tag{43}$$

which is (40). Monotonicity in k follows from the Loewner order:

$$L_{k+1} \succeq L_k \qquad \text{implies} \qquad L_{k+1}^{-1} \preceq L_k^{-1}. \tag{44}$$

□

Lemma 3 (Frontier floor)**.** *For any serialized insertion index k, if $\phi_u^\top L_k^{-1}\phi_u \ge \delta$, then*

$$\Delta(u \mid L_k) \ge c_\delta := \log(1+\alpha v_{\min}\delta). \tag{45}$$

Mass outside the span suffices: $\|P_{W_k^\perp}\phi_u\|^2 \ge \delta$ implies the hypothesis.

Proof. The payment floor (45) follows from (38) with $v_u \ge v_{\min}$. For the second claim, the reference leaves W_k and $W_k^\perp$ invariant and acts as the identity on $W_k^\perp$, so its inverse splits as

$$L_k^{-1} = (L_k|_{W_k})^{-1} \oplus \mathbf{I}|_{W_k^\perp}. \tag{46}$$

By (46) the quadratic form equals its $W_k^\perp$ component in full plus a nonnegative contribution from W_k, hence

$$\phi_u^\top L_k^{-1}\phi_u \;\ge\; \|P_{W_k^\perp}\phi_u\|^2, \tag{47}$$

which is at least δ whenever the projected mass is. □

Proof of Proposition 1. Fix any γ-rational profile and let c_δ be as in Lemma 3.

Step 1 (bounded dwell mass). Suppose an output is submitted at mode s at an instant when the accumulated mass has reached the saturation threshold

$$m_s \;\ge\; m^\star := \frac{v_{\max}}{e^{\gamma c_\delta} - 1}. \tag{48}$$

By Lemma 2, the payment of that submission is then at most

$$\log\left(1 + \frac{\alpha v_{\max}}{\alpha m_s}\right) \;\le\; \log\left(1 + \frac{v_{\max}}{m^\star}\right) \;=\; \gamma c_\delta, \tag{49}$$

with the inequality strict when $m_s > m^\star$, while the developer-specific frontier and Lemma 3 guarantee the current developer an available payment of at least c_δ. Pathwise γ-rationality therefore rules out any submission once m_s exceeds $m^\star$, so the lifetime mass of every mode is capped at

$$\bar{m} := m^\star + v_{\max}, \tag{50}$$

where the second term allows for the single serialized insertion that crosses the threshold. Later submissions see the updated reference.

Step 2 (linear innovation). Every serialized output is gate-admitted and has validity at least $v_{\min}$, so the total admitted mass after T rounds is at least $Nv_{\min}T$. Step 1 caps each mode at $\bar{m}$, so the number of distinct modes played satisfies

$$D_T \;\geq\; \frac{Nv_{\min}}{\bar{m}}\,T, \tag{51}$$

and trivially $D_T \leq NT$.

Step 3 (effective mode count). Let $p_s(T) = m_s(T)/\sum_u m_u(T)$ be the cumulative composition. Steps 1 and 2 bound the numerator and the denominator separately, so every coordinate satisfies

$$p_s(T) \;\leq\; \frac{\bar{m}}{Nv_{\min}T} \qquad \text{for every } s. \tag{52}$$

Hence every finite mode set F satisfies

$$p_F(T) \leq |F|\bar{m}/(Nv_{\min}T), \tag{53}$$

and the participation ratio diverges,

$$N_{\text{eff}}(T) = \frac{1}{\sum_s p_s(T)^2} \;\geq\; \frac{1}{\max_s p_s(T)} \;\geq\; \frac{Nv_{\min}T}{\bar{m}}. \tag{54}$$

If the conditions hold over unbounded horizons, the displayed bounds imply that every fixed finite mode set has vanishing share and no finite-support limit composition exists.

Step 4 (linear coverage accrual). The rich frontier offers the acting developer a payment of at least c_δ at every insertion, so pathwise γ-rationality makes every realized payment at least γc_δ. The telescoping identity (39) therefore gives

$$V(L_T) \;=\; \sum_{k\leq NT} \Delta_k \;\geq\; \gamma c_\delta NT. \tag{55}$$

□

Proof of Corollary 4. Let $s^\star$ be the unique argmax of q. The payment available at every instant is then $\max_s q_s = q_{s^\star}$, so a γ-rational output at mode s requires $q_s \geq \gamma q_{s^\star}$, and any

$$\gamma > \sup_{s\neq s^\star} \frac{q_s}{q_{s^\star}} \tag{56}$$

forces every output to $s^\star$. Coverage is still accrued with the validity weights, so the reference stays rank one and its regularized log-determinant grows logarithmically:

$$L_T = \mathbf{I} + \alpha v_{s^\star} NT\, \phi_{s^\star}\phi_{s^\star}^\top, \qquad V(L_T) = \log(1+\alpha v_{s^\star} NT) = \Theta(\log T). \tag{57}$$

□

The threshold (56) is a ratio of *static payoffs*, because γ-rationality compares an output against the best payment on offer and under a static score that payment is q. Validity affects only the accrual rate, through the mode that score selects.

Proposition 4 (Finite orthogonal frontier)**.** *Suppose all developers share M orthonormal valid modes and use exact best responses under the persistent reference protocol. Then every mode is played infinitely often and, uniformly in s,*

$$n_s(T) = \frac{NT}{M} + O(1), \qquad m_s(T) = \frac{v_s NT}{M} + O(1). \tag{58}$$

Thus play-count frequencies converge to $1/M$, while gated-mass shares converge to $v_s/\sum_u v_u$.

Proof. Orthonormality makes the payment at mode s

$$g_s(n_s) = \log\!\left(1 + \frac{1}{n_s + (\alpha v_s)^{-1}}\right). \tag{59}$$

Because g_s decreases in the adjusted count $a_s = n_s + (\alpha v_s)^{-1}$, an exact best response increments a smallest a_s. Least-load balancing brings all adjusted counts within a bounded interval and keeps them there; no coordinate can remain fixed while the others diverge, because it would eventually be the unique minimum. Hence $a_s - a_t = O(1)$ for every s, t. Together with $\sum_s n_s(T) = NT$, this gives (58); the two normalized limits follow by division. □

E.2 The Two Conditions, Measured

We measure both conditions of Proposition 1 in the factorial of Section H.7, whose log records the MOSAIC score of every eligible candidate at each of its 18,432 insertions. With $\alpha = 1$ and unit weights that score is $\log(1 + \phi^\top L^{-1}\phi)$, so we recover the quadratic form that bounds the payment in Lemma 3 as $e^{\text{score}} - 1$ exactly, with no refitting. We validate the measurement two ways: the recorded payment equals the menu maximum at every insertion, to 0, since the replay rule is an argmax; and no measured value exceeds the ceiling Lemma 2 proves for it.

The frontier floor equals $1/(1+\alpha m)$ at the heaviest mode, exactly. We take δ to be the quadratic form of the best candidate then available; its floor over the horizon is 0.048 with a persistent shared reference and 1 of 8 with a persistent own reference; medians are 0.20 and 0.50. Round by round, that floor equals $1/(1 + \alpha m)$ at the mass m the heaviest mode has by then accumulated. The observed-to-predicted ratio is 1.000 in every one of the eight rounds under an own persistent reference and never below 0.999 under a shared one. The own-reference floor is $1/(t + 1)$ at round t, counting rounds from $t = 0$, to every digit recorded, one developer adding one unit of mass per round to the mode it has already filled. Lemma 2 proves only the inequality. On both persistent arms it is an identity, so on real candidates the declared map produces the orthogonal-mode reference that Proposition 1 assumes.

Resetting the reference each round breaks the identity, so the identity is a consequence of accumulated mass: the same formula underestimates the observed floor by a factor that grows from 1.0 in the first round to 5.0 by the sixth under either reset reference, median 3.5 for both. We run all four arms on one frozen panel, and all four realize the same 5.54 to 5.66 distinct modes per prompt, so the reference alone accounts for the separation in floors.

So δ is a function of the horizon rather than a free constant: a fixed δ holds as a floor only while the heaviest mode's mass stays below $1/\delta - 1$; at three developers and $\delta = 0.048$ that mass is reached after 6.7 rounds, and we ran eight. This decay is the measured form of the vanishing residuals described at the end of this subsection. Sharing the reference divides δ by 2.6, since three developers saturate a mode three times faster.

A longer horizon does not make the count bound linear. With $\bar{m} = 1/\delta_T + 1$ and the measured $\delta_T = 1/(1 + m_{\max}(T))$, the own-reference floor is $\delta_T = 1/T$ to every recorded digit, so the count bound $Nv_{\min}T/\bar{m}$ simplifies to $NT/(T + 1)$, which climbs only to a ceiling of N, three here. For the bound to reach the count the panel realizes, each prompt needs a wider mode vocabulary, so only a wider panel puts the count bound in the linear phase Proposition 1 describes.

Rationality does not bind here. Under the replay's own rule γ is one by construction, so the question is what a developer that ignores the score would collect. A uniformly random pick from the acting developer's own menu earns the full maximum in 98% of insertions, because the opcode map of Section H.7 sends many candidates in a menu to one mode and the argmax set is large. We therefore report no median for γ and quote the worst case instead: the lowest-scoring eligible candidate is paid 0.075 of the best available payment under a shared persistent reference and 0.19 under an own one.

What the constants then predict. Substituting $\gamma = 1$ and the own-reference $\delta = 1/8$ into the constants of Section 4 gives $c_\delta = 0.118$ and a mass cap $\bar{m} = 9.0$, hence $D_T \geq \mathbf{2.67}$ distinct modes per prompt against $\mathbf{5.65}$ realized, a factor of 2.1. The shared reference, whose δ is 2.6 times smaller, gives $c_\delta = 0.046$, $\bar{m} = 22$ and $D_T \geq 1.1$ against 5.7 realized. The cap is the size of the whole horizon: a prompt stream receives 24 insertions, so the mass at which γ-rationality pushes a developer off a saturated mode is reached only on the final round, and the counting argument completes a single cycle. For (7) to reach the realized count, δ would have to be 0.31. The horizon lies inside the linear phase, where we measure the diversification directly in Section 5.3.

Where the two conditions have to be measured. The mass cap in Proposition 1 needs Definition 5 at every insertion, so γ has to be measured per insertion: a population rational only in conditional expectation can still put unbounded mass on one mode. A soft gate that admits additional bounded mass adds positive-semidefinite terms to L_T, so covered-direction payments fall and Lemma 3 still holds once the rich frontier is evaluated against the full admitted history. A dimension

condition would be weaker than the δ condition: a kernel with decaying spectrum has unboundedly many fresh directions whose residuals vanish, so coverage still accrues sublinearly, which is the failure the δ condition excludes and dimension does not.

Relation to prior work. The ingredients of Lemmas 2 and 3 are standard in linear bandits and experimental design. The decay and telescoping arguments connect to the elliptical-potential lemma (Abbasi-Yadkori et al., 2011); threshold abandonment connects to rotting bandits (Levine et al., 2017; Kim et al., 2022); for the surrounding innovation-contest, generative-market, dynamic-benchmark, data-valuation, and complexity-economics literatures see (Letina & Schmutzler, 2019; Raghavan, 2024; Shirali et al., 2023; Xu et al., 2021; Arthur, 1994). Unlike an algorithm-specific bound, Proposition 1 covers every pathwise γ-rational submission process under its stated assumptions.

F Information Content and Accrual Under a Fixed Representation

The local form is also what makes the score cheap enough to compute as submissions arrive. Scoring a candidate costs one matrix-vector product against a cached L_S^{-1}; admitting it costs one Sherman–Morrison update of that cache. Both are $O(d^2)$, and no determinant is ever recomputed. Weight and representation stay factored in that update: w_i scales the rank-one term, while ϕ_i sets its direction, hence which admitted distinctions it overlaps.

The same local form also has an exact information-theoretic interpretation. Identify model quality with a vector θ on the feature space, so quality along direction ϕ is $\phi^\top\theta$, and let evaluating output i return $y_i = \phi_i^\top\theta + \varepsilon_i$.

Proposition 5 (The MOSAIC score is an information gain)**.** *Let $\theta \sim \mathcal{N}(0, \alpha\mathbf{I})$, let the evaluation noise be independent Gaussian of precision w_i, and measure mutual information I in nats. Then, writing $y_S = (y_j)_{j\in S}$ for the evaluations of the reference,*

$$V(S) = 2I\big(\theta; y_S\big), \qquad \Delta_i(S) = 2I\big(\theta; y_i \mid y_S\big). \tag{60}$$

The evidence separating two quality vectors is a quadratic form in $L_S - \mathbf{I}$. It vanishes along directions orthogonal to the span of the population's features, where the posterior variance of a quality functional remains at its prior value.

Coverage is therefore information, in nats, and the score is the information one further evaluation adds given what the population has already revealed. This is the D-optimal criterion of classical experiment design, and it explains why an uncovered direction receives more credit: evaluating it resolves quality along an axis on which the population has provided no evidence. MOSAIC turns a curator's sampling objective into a developer incentive. The proof and the precise separation identity are in Section F.1.

F.1 Valid Coverage Is Information about Model Quality

The long-run statements of Section 4 concern accumulated V, which is also the objective the mechanism optimizes; if V were only that objective, the accrual claim would be circular. The information identity of Proposition 5 removes the circularity for the objective, though not for the representation: the declared feature map determines the design matrix, and every payment is computed inside that representation.

Proof of Proposition 5. The model is a Gaussian linear model with prior precision $\alpha^{-1}\mathbf{I}$ and observation precisions w_i, so the posterior precision is the prior precision plus the weighted second moment of the observed features,

$$\mathrm{Cov}\big(\theta \mid y_S\big)^{-1} = \alpha^{-1}\mathbf{I} + \sum_{i\in S} w_i\phi_i\phi_i^\top = \alpha^{-1}L_S. \tag{61}$$

The posterior covariance is therefore αL_S^{-1}, and evaluating it on ψ gives the posterior variance of a quality functional,

$$\mathrm{Var}\big(\psi^\top\theta \mid y_S\big) = \alpha\,\psi^\top L_S^{-1}\psi, \tag{62}$$

which is at its prior value $\alpha\|\psi\|^2$ exactly when ψ is orthogonal to every admitted direction of positive weight. For jointly Gaussian variables the mutual information is half the log ratio of prior to posterior covariance determinants (Cover & Thomas, 2006), so

$$2I\big(\theta; y_S\big) = \log \frac{\det(\alpha \mathbf{I})}{\det(\alpha L_S^{-1})} = \log\det L_S = V(S), \tag{63}$$

and applying (63) to $S \cup \{i\}$ and subtracting gives the conditional form,

$$2I\big(\theta; y_i \mid y_S\big) = V(S \cup \{i\}) - V(S) = \Delta_i(S). \tag{64}$$

In the Hilbert-space setting of Section E.1, where $\mathcal{N}(0, \alpha\mathbf{I})$ is not a measure, read $I(\theta; y_S)$ as $I(P_W\theta; y_S)$ for the finite-dimensional span W of the admitted features together with whichever ψ or $\theta - \theta'$ is under discussion, on which the prior is $\mathcal{N}(0, \alpha\mathbf{I}_W)$: the observations depend on θ only through $P_W\theta$ and $L_S - \mathbf{I}$ vanishes off W, so the identities are unchanged. Finally, given θ the evaluation record is Gaussian with mean linear in θ and covariance free of θ, so

$$\mathrm{KL}\big(P_{y_S|\theta} \,\|\, P_{y_S|\theta'}\big) = \frac{1}{2}\sum_{i\in S} w_i\big(\phi_i^\top(\theta - \theta')\big)^2 = \frac{1}{2\alpha}\,\big\|\theta - \theta'\big\|^2_{L_S - \mathbf{I}}, \tag{65}$$

which is the separation statement of Proposition 5 and vanishes exactly when $\theta - \theta'$ is orthogonal to every admitted direction of positive weight. □ □

The identities hold exactly at the ridge and the weights the implementation uses, and they need only that the design be fixed before outcomes are realized. Interpreting V as information does commit us to two modelling choices: we identify the gate with an evaluation's reliability, not a response's merit, and we treat a repeated submission as an independent measurement.

V is twice Lindley's expected Shannon information for the normal linear model (Lindley, 1956; Chaloner & Verdinelli, 1995), so maximizing coverage is Bayesian D-optimal design, and the accrued volume is twice the log shrinkage of the posterior credible ellipsoid, $\mathrm{vol}(E_S)/\,\mathrm{vol}(E_0) = e^{-V(S)/2}$. The classical D-criterion $\log\det\big(\sum_i w_i\phi_i\phi_i^\top\big)$ is the $\alpha \to \infty$ limit of $V(S) - d\log\alpha$ and is $-\infty$ whenever the admitted features fail to span $\mathbb{R}^d$ (Pukelsheim, 2006), so the ridge is what makes the criterion finite on a population that covers only part of the space. Two further consequences of the identity are immediate. Remark 1 becomes an information statement: a zero-weight output is an infinitely noisy measurement, so invalid padding carries no information about θ and is unpaid for that reason. And the lifetime budget of Proposition 1 is the repeated-measurement law: the mechanism pays r repeated submissions of one feature a total of $\sum_{k\le r}\log\frac{1+\alpha w k}{1+\alpha w(k-1)} = \log(1+\alpha w r)$, so one feature yields $O(\log r)$ nats however often it is resubmitted. A last consequence cuts the other way. Because Gaussian information gain does not depend on the realized outcome, Δ_i is computable before any evaluation is run, which is what makes the mechanism implementable, and by the same token the mechanism can reward a submission for being informative but never for being correct. We leave correctness to the validity gate, which is why the gate is applied before coverage is computed.

None of these identities is new. They are the maximum information gain of Gaussian process optimization specialized to a linear kernel (Srinivas et al., 2010) and the log-determinant information objective of near-optimal sensor placement (Krause et al., 2008). What is new here is that a payment rule implements the criterion, so the informativeness of the evaluation is produced by the incentives of the developers rather than by the sampling policy of a curator.

Proposition 1 needs a uniformly rich frontier, which no finite-dimensional representation can sustain indefinitely. Corollary 5 caps accrual in a fixed representation at $O(d\log T)$ and, beneath that cap, gives the factor of M gained by spreading over the M directions the representation can resolve.

Corollary 5 (Accrual in a fixed finite-dimensional representation)**.** *Let the features be unit vectors in dimension d with gated weights at most $v_{\max}$. Then for every policy and every realization,*

$$V(L_T) \;\le\; d\log\!\left(1 + \frac{\alpha v_{\max} NT}{d}\right) \;=\; O(d\log T), \tag{66}$$

with equality whenever d divides NT and the population cycles through an orthonormal basis at weight $v_{\max}$. If all developers share the same $M \le d$ orthonormal valid modes, their mode weights

satisfy $v_s \in [v_{\min}, v_{\max}]$ with $v_{\min} > 0$, and they play exact best responses to the MOSAIC payment under the accumulating reference of Section 4, then

$$V(L_T) = M \log T + O(1), \tag{67}$$

a factor of M over the single-mode rate of Corollary 4.

Proof. Every admitted output contributes a rank-one term of trace at most $\alpha v_{\max}$, so $\operatorname{tr} L_T \le d + \alpha v_{\max} NT$. Among positive definite matrices of a given trace the determinant is largest when the spectrum is flat, hence

$$V(L_T) = \log\det L_T \;\le\; d \log \frac{\operatorname{tr} L_T}{d} \;\le\; d \log\left(1 + \frac{\alpha v_{\max} NT}{d}\right), \tag{68}$$

which is (66); this determinant–trace inequality is the bound behind the elliptical potential lemma (Abbasi-Yadkori et al., 2011). For (67) the reference is diagonal in the mode basis, so $V(L_T) = \sum_{s \le M} \log(1 + \alpha m_s(T))$, and Proposition 4 gives $m_s(T) = v_s NT/M + O(1)$ for every mode; summing the M logarithms gives $M \log T + O(1)$, against the $\log T + O(1)$ of (57). □

G Security and Manipulation Boundary

G.1 Representation, Validity, and Ownership Boundary

Under exact zero weight, an invalid response affects no live score (Remark 1). A soft empirical weight is generally nonzero, so this exact exclusion result does not describe soft-gate error.

The attack surface specific to the mechanism is splitting at the representation level. Suppose one capability is written in several paraphrased or partitioned forms and the kernel maps those forms to separated directions.

Proposition 6 (Splitting). *Fix a standing reference S and a submitter holding one valid capability.*

(i) Presented as k outputs that the representation maps to orthonormal directions orthogonal to the span of S, each admitted at weight w, the capability collects $k \log(1 + \alpha w)$ under either pricing convention. That is exactly k times the honest single submission's payment when the honest direction is itself orthogonal to that span, and more when it is not. No split geometry does better: for any unit directions $\Psi = [\psi_1, \dots, \psi_k]$, Hadamard's inequality gives $\log\det(\mathbf{I} + \alpha w \Psi^\top L_S^{-1} \Psi) \le k \log(1 + \alpha w)$.

(ii) A rule that sees only the admitted feature–weight pairs and the identities claimed for them assigns identical scores in two histories that present the same object: k developers with k distinct capabilities, and one submitter presenting one capability under k claimed identities. Any rule that pays $c > 0$ for a fresh orthogonal direction therefore pays the splitter at least kc.

(iii) If the gate conserves gated mass across a capability, so that the k fragments split one weight w between them, the split collects at most $k \log(1 + \alpha w/k) < \alpha w$ for every k. Measured against the honest submission, paid $\log(1 + \alpha w\, \phi^\top L_S^{-1} \phi)$, the gain is at most $\alpha w / \log(1 + \alpha w\, \phi^\top L_S^{-1} \phi)$: on a fresh honest direction, where $\phi^\top L_S^{-1} \phi = 1$, that is $\alpha w / \log(1 + \alpha w)$, at most $1/\log 2$ whenever $\alpha w \le 1$, the asymptotic replication gain of Corollary 3; on a direction the reference already covers it grows inversely with the residual. Conserving mass caps the absolute payment to a split.

Proof. In (i) each split direction is orthogonal to S and to its predecessors, so its quadratic form in (2) is exactly one and it is paid $\log(1 + \alpha w)$, while the honest submission is paid

$$\log\bigl(1 + \alpha w\, \phi^\top L_S^{-1} \phi\bigr) \;\le\; \log(1 + \alpha w).$$

The Hadamard bound follows from

$$\bigl(\Psi^\top L_S^{-1} \Psi\bigr)_{jj} \;\le\; \|\psi_j\|^2 = 1.$$

Part (ii) is immediate, since a rule cannot condition on a distinction its inputs do not record.

In (iii) the fragments carry weights $w_j \geq 0$ with $\sum_j w_j = w$. Hadamard's inequality bounds their joint gain by $\sum_j \log(1 + \alpha w_j)$, and Jensen's inequality bounds that in turn by $k \log(1 + \alpha w/k)$; under live leave-one-out pricing, submodularity bounds the sum of the removal gains by the same joint gain, so either convention pays at most $k \log(1 + \alpha w/k)$. The map

$$k \longmapsto k \log(1 + \alpha w/k)$$

increases to the supremum αw, while the honest submission is paid $\log(1 + \alpha w\, \phi^\top L_S^{-1} \phi)$ by (2), which gives the stated ratio; it is $\alpha w / \log(1 + \alpha w)$ at unit residual and diverges as the residual falls. At $\alpha = w = 1$ against a reference already carrying mass 100 on the honest direction, $\phi^\top L_S^{-1} \phi = 1/101$: the honest submission is paid $\log(102/101) = 0.0099$ while two half-weight fragments in fresh directions collect $2 \log(3/2) = 0.81$, a factor of 82. □ □

Part (ii) is an information statement and not a strategic one: MOSAIC cannot condition on an ownership distinction that the representation does not record, so any bound on the gain must come from canonicalization, from a gate that conserves mass across a capability, or from identity that is costly or verified. Claimed identities are free, and part (ii) already covers them. The false-name literature draws the same division of labor (Yokoo et al., 2004; Todo et al., 2009; Conitzer & Yokoo, 2010), and Hays et al. (2026) is its leaderboard instance: their You-Rank-We-Rank mechanism achieves approximate clone robustness by making producers rank their own submissions, that is, by collecting exactly the ownership report that feature–weight pairs alone do not determine. Part (iii) is the least costly remedy available to a deployment, and it is implemented in the validity weight: with $\alpha v_{\max}$ of order one and gated mass conserved across a capability, what splitting collects is bounded by a constant, uniformly in the number of fragments. The bound is on the absolute payment, not on the ratio to an honest entry: fragments of a capability the record already covers are still paid as fresh directions, and only canonicalization stops that.

Robustness to splitting therefore rests on kernel and partition quality alone.

Unreachable code changes the signature on 99% of problems and adds 10% of the coverage range. On the same 164-problem HumanEval panel we pad the base solution with a dead branch (`if False:` around a copy of its first loop), a no-op loop (`for _ in range(0): pass`), a guarded early return (`if 0: return None`), and all three together. The four pads lower the cosine between padded and base abstract-syntax signatures below 0.999 on 163, 164, 163, and 164 problems, with mean cosine to the base of 0.945, 0.952, 0.967, and 0.887. The five-set of base plus the four pads scores 1.96 nats, 10% of the range from the 1.79 floor to the 3.47 ceiling (per-problem median 7.6%, maximum 34%), against 48% for five distinct verified solutions, and the padded set scores below the distinct set on all 164 problems. The signature counts control-flow constructs, so unreachable constructs are a splittability exposure of the same form as Proposition 6 and invisible to an execution gate, since the pads never run; a canonicaliser that drops unreachable code before hashing, or a coverage-weighted signature over executed statements, removes the exposure.

A second manipulation lowers a rival's payment. Because payments are leave-one-out, a coalition can suppress that payment by placing gated mass on the direction the rival is paid for.

Proposition 7 (Blocking)**.** *Let developer i hold the unit direction ϕ_i at weight w_i.*

(i) If other submissions place gated mass m on features collinear with ϕ_i, then $\Delta_i \leq \log(1 + w_i/m)$ independently of α, so mass $m = \Theta(1/\varepsilon)$ cuts i's payment to an ε fraction of what it would be unsuppressed.

(ii) Suppression requires covering ϕ_i. Against any reference whose admitted directions span W, $\Delta_i \geq \log(1 + \alpha w_i \|P_{W^\perp} \phi_i\|^2)$, so a positive lower bound remains unless the admitted directions span ϕ_i, which the coalition must arrange. Collinear mass is the cheapest way to do so, and decomposing ϕ_i into directions the coalition can produce also works. Blocking a capability means covering it.

(iii) For $k \geq 2$ blockers each placing weight w on ϕ_i, their leave-one-out payments total at most $2 \log 2$ nats however large k is, whereas under a δ-rich frontier the same k submissions could have earned at least $k c_\delta$.

(iv) *Lemma 3 holds after every admissible history, the coalition's own flood included, so no coalition can push the payment* available *to a developer below* c_δ.

Proof. Part (i) is Lemma 2 substituted into the local form (2) and part (ii) is Lemma 3 applied to ϕ_i. For (iii), let B be the reference without the flood and $x = \alpha w \phi_i^\top B^{-1} \phi_i$; each of the k blockers is paid $\log((1+kx)/(1+(k-1)x))$, so the coalition collects

$$k \log\left(1 + \frac{x}{1+(k-1)x}\right) \;\le\; k \log \frac{k}{k-1} \;\le\; 2 \log 2, \tag{69}$$

the first step because $x/(1+(k-1)x) \le 1/(k-1)$ and the second because $k \log \frac{k}{k-1}$ decreases in k; and by Lemma 3 each forgone submission would have been paid at least c_δ. Part (iv) is Lemma 3. □ □

A blocking coalition therefore forgoes $\Omega(k)$ in payment and collects $O(1)$, and its suppression hits every developer on the targeted direction. Three cases escape that accounting. An attacker optimizing rank can trade a bounded absolute loss for a decisive relative gain. Under the accumulating reference of Section 4 a single flood suppresses a direction in every later round, so the coalition inflicts harm of order T on the rival for a one-shot cost of order k; the flood itself is paid by insertion there, $\log(1+\alpha m)$, which grows with m but leaves the coalition's net cost at order k. Renormalizing the reference restores per-round symmetry. And once the frontier is exhausted c_δ vanishes with it, so blocking is cheapest exactly when the benchmark has least left to offer. Propositions 6 and 7 both concern deviations by several identities at once, which is exactly the class that the potential identity of Theorem 2, stated for unilateral replacements, does not cover.

The rule therefore gives exact alignment, budget feasibility, and a factor-two welfare guarantee that holds under learning, and it does not give robustness to splitting or to the targeted blocking of a committed direction. Both of the latter need identity or provenance: no score computed from features establishes either, so the evaluator has to enforce them administratively.

G.2 Threat Model by Mechanism Component

The guarantees are component-specific. Diminishing returns address paraphrases only when the feature map keeps them nearby; zero weight blocks invalid padding only when the gate is correct; and coverage scoring is not false-name-proof when one capability can be split across features or identities. Disjoint judges and external endpoints reduce shared-evaluator dependence but do not establish adversarial robustness.

The replaceable active population avoids vested first-arrival scores by rescoring every current entry. It instead requires verified, rate-limited slots and fresh-state commits. A submitter who controls several nominal developers can still occupy several directions or several copies of one direction, and a coordinated coalition can make profitable joint deviations even at a pure Nash profile. Theorem 2 covers unilateral changes by verified slot owners; it does not provide Sybil resistance, coalition-proofness, or a budget for registry subsidies.

Within that scope, a pre-outcome-frozen sweep over alias counts $1, 2, 4, 8, 16$, run post hoc on already-generated model outputs, verifies the containment the registry is meant to give: clone weight, coverage, owner reward, and reward share are invariant under pure relabeling and under input permutation, and a conflicting orthogonal feature submitted into an already-occupied slot is rejected. The sweep confirms linked-alias containment given verified, unique principals; it does not test real-world identity verification.

H Experimental Details and Additional Results

H.1 Every Reported Comparison, in Full

The subsections below give the protocol behind each row, in the order the main text reports them.

Table 3: **Interval, test statistic, and paired effect size for every comparison the main text reports.** Intervals are paired Student-t, except the later-window row, which is a paired task bootstrap. n counts matched seeds, except for HumanEval problems in the first row, later-window tasks in the fifth, and four-slot populations in the sixth and seventh. A dash marks a contrast reported as an interval.

Comparison	Reading	n	Effect and 95% CI	t	d_z
Distinct vs. rewritten programs	coverage G_H	164	$+0.804$ $[0.771, 0.838]$	—	3.70
Shared vs. own persistent	Hill-2 modes	24	$+0.148$ $[0.085, 0.210]$	4.88	0.996
Shared vs. own persistent	Hill-1 modes	24	$+0.155$ $[0.073, 0.236]$	—	—
Shared vs. own persistent	log-det. coverage	24	$+0.118$ $[0.047, 0.188]$	—	—
Fresh-state vs. launch-time reference	tasks solved (pp)	131	$+3.05$ $[0.76, 6.11]$	—	—
Fresh-state vs. launch-time reference	task union (pp)	12	$+1.394$ $[0.081, 2.707]$	—	—
Fresh-state vs. launch-time reference	per-slot match (pp)	12	$+0.485$ $[-0.647, 1.617]$	—	—
Own persistent vs. uniform, adaptation	Hill-1 modes	24	$+17.69$ $[13.43, 21.95]$	8.59	1.75
Shared persistent vs. uniform, adaptation	Hill-1 modes	24	$+16.27$ $[12.11, 20.44]$	8.09	1.65
Own reset vs. uniform, adaptation	Hill-1 modes	24	$+7.10$ $[2.60, 11.61]$	3.26	0.67
Own persistent vs. own reset, adaptation	Hill-1 modes	24	$+10.59$ $[6.40, 14.77]$	5.23	1.07
Shared vs. own persistent, adaptation	Hill-1 modes	24	-1.42 $[-5.59, 2.75]$	-0.70	-0.14
Shared vs. own reset, adaptation	Hill-1 modes	24	$+2.92$ $[-1.32, 7.16]$	1.43	0.29
Own persistent vs. uniform, adaptation	held-out match (pp)	24	-1.87 $[-3.70, -0.03]$	-2.10	-0.43
Shared vs. own persistent, adaptation	held-out match (pp)	24	$+0.81$ $[-0.87, 2.50]$	1.00	0.20
Shared persistent vs. uniform, 8 developers	Hill-1 modes	6	$+21.41$ $[12.91, 29.90]$	6.47	2.64
Shared persistent vs. uniform, 16 developers	Hill-1 modes	6	$+22.15$ $[16.56, 27.74]$	10.18	4.16
Shared persistent vs. uniform, frozen weights	specialization NMI	6	$+0.103$ $[0.043, 0.162]$	4.42	1.81

H.2 Representation and Gate Diagnostics

Represented coverage separates distinct programs from renamed copies in every paired problem. With $\alpha = 1$ and unit weights, mean represented coverage is $\mathbf{2.60}$ for distinct sets and $\mathbf{1.79}$ for rewrite sets, a paired gap of 0.804 whose smallest within-problem value is 0.130; distinct programs score higher in every paired comparison. The rewrite figure is the minimum of the scale: the signature sends all five rewrites of a problem to one unit vector, whose coverage is $\ln 6 = 1.79$, and the measured mean equals it on all 164 problems. Five orthogonal directions would yield $5 \ln 2 = 3.466$, so the distinct-set mean lies at $\mathbf{48}\%$ of the range the map can express, with the smallest problem at 7.7% and the largest at 74%. The gap is not a validity effect, because all 820 rewrite-set programs pass the executable tests. For each of 164 HumanEval problems the distinct set contains five verified-correct solutions, while the rewrite set contains one of those solutions and four deterministic identifier-and-comment rewrites. Unit tests define validity, and a normalized fixed-width abstract-syntax-tree signature represents control flow, operators, calls, recursion, nesting depth, and statement count. We omit identifiers and comments from the signature by design, so invariance to those rewrites is part of the map's definition; the measured sensitivity is to alternative syntactic constructions.

We measure every representation the paper reports on the same paired panel. We rerun the construction above under three feature maps with $\alpha = 1$, unit weights, and unit-normalized features. For the abstract-syntax signature the gap above has 95% CI $[0.771, 0.838]$ and $d_z = 3.70$. Under an operator-only projection of that signature, which retains the operator block, nesting depth, recursion flag, and statement count, distinct sets score 2.80 against 1.79 for rewrites (gap 1.01, $[0.97, 1.05]$, all 164 problems). Under raw `all-MiniLM-L6-v2` embeddings, distinct sets score 2.29 against 2.04 for rewrites (gap 0.26, $[0.23, 0.28]$, $d_z = 1.59$, distinct higher on 98% of the 164), so a renamed copy adds most of the coverage a structurally distinct program adds before any merge is applied.

The deployed merge threshold removes the paraphrase channel and also merges genuinely distinct programs. Under the same embeddings, rewrite pairs have mean cosine 0.907 and 92.5% of them lie at or above the 0.8 merge threshold, while distinct pairs have mean cosine 0.849 and only 23.5% fall below it. Greedy clustering at that threshold leaves the five-rewrite set at 1.14 modes, exactly one mode in 86.6% of problems, and the five-distinct set at 1.88. This mode count therefore

separates the two sets by $1.65\times$ while discarding genuine distinctions, so the persistent-reference mode gains it reports are lower bounds.

H.3 The Mechanism on Frozen Outputs

Exhaustive enumeration returns zero exact-potential residual and no non-terminating path. In the finite-game replay we measure exact-potential residuals, convergence, and equilibrium efficiency over the complete state space. Each of 12 seeded games has four slots and four frozen actions per slot; a deterministic client accepts the highest-scoring strict improvement under a fixed tie break. Enumeration covers every profile, unilateral replacement, start state, and six update schedules (Table 4): the maximum identity residual is **0** across 36,864 non-incumbent replacements, and **all** 18,432 strict-improvement paths terminate at a pure Nash profile.

Quantity	Exhaustive result
Finite state space	$12 \times 4^4 = 3{,}072$ profiles
Exact-potential identity	36,864 non-incumbent replacements (49,152 comparisons including incumbents); maximum $\lvert\Delta s_i - \Delta G_H\rvert = 0$
Strict-improvement dynamics	18,432 paths; all increase G_H, avoid cycles, and terminate at pure Nash profiles
Terminal strictness	all 12 non-incumbent alternatives at each equilibrium are negative; closest gain -0.026
Equilibrium efficiency	minimum worst-equilibrium/optimum ratio 0.992; 18,072/18,432 paths reach an optimum
Common-launch comparison	fresh-state scoring reaches an optimum in all 12 games; frozen-reference scoring in 9

Table 4: **Complete results for all four-slot games.** Twelve four-slot games with four frozen actions per slot yield 3,072 profiles, 36,864 non-incumbent replacements, and 18,432 all-start paths under six schedules. The score game satisfies the exact-potential identity with zero residual; every maximal strict-improvement path terminates at a pure Nash profile; and the minimum worst-equilibrium/optimum represented-coverage ratio is 0.992.

Accepted gains exceed the stopping threshold by two orders of magnitude. Across the 36,864 non-incumbent replacements, the smallest positive gain is 2.10×10^{-8}, or $\mathbf{209.6}\times$ the 10^{-10} threshold, so no acceptance in the enumeration rides on floating-point noise.

H.4 Frozen Asynchronous State-Transition Stress Test

Stale quotes decrease the potential in 87.5% of controls; fresh-state rescoring decreases it in none. All 480 fresh-state cells have zero identity residual, zero potential decrease, no repeated population, and no profitable terminal deviation. By contrast, among 352 stale-proposal controls, **308** decrease the potential ($\mathbf{87.5}\%$) and 195 repeat a population. We also ran the 128 serialized control cells, where a snapshot is the current state and its rule is therefore the fresh-state rule: those cells reproduce the fresh-reference cells exactly, and we report them separately, since a control that cannot fail cannot measure how often the rule fails. Over all 480 control cells the decrease rate is 64.2%. We fix 32 synthetic menus, $N \in \{4, 8, 16, 32\}$ slots, six actions per slot, 32 features, and batch sizes $\{1, \lceil N/4\rceil, \lceil N/2\rceil, N\}$. Because the two arms share menus, snapshots, and arrival schedules, commit-time rescoring is the only changed component.

H.5 Official-Release Replay and Later-Window Evaluation

The later window is held out of selection by construction. We replay the 14 official LiveCodeBench submission bundles released before the pinned snapshot date: five Anthropic, five Google, and four OpenAI releases. The common intersection contains 1,055 problems. The first

924 dated problems define population features and quality weights; the disjoint **131**-problem later window enters **neither** selection nor trace choice.

Fresh-state commits never decrease the potential; stale acceptance decreases it in 8 of 18 cells. Fresh-state CAS has zero potential decrease and no repeated population in all **18** cells, and reaches a pure Nash profile. Stale acceptance, however, decreases the potential and repeats a population in **8** of 18 cells, including all 6 full-batch orders. We cross fresh-state CAS against stale acceptance, batch sizes $\{1, 2, 3\}$, and all six provider orders under matched proposals. In the three-proposal cell in particular, the second accepted proposal was credited $+0.00161$ nats against the population it was scored on and -0.00045 nats against the population it was committed to.

Live-reference selection solves 3.05 points more of the held-out later window. Live- and frozen-reference populations tie on 92% of the 924 selection-window tasks. On the later window the fresh-state population solves 85% of 131 tasks and the frozen-reference population 82%, with four live-only solves and no frozen-only solve: a difference of **3.05** points, paired task-bootstrap 95% CI $[0.76, 6.11]$. The populations share Claude Sonnet 4 Thinking and O4-Mini High; the one differing slot holds Gemini 2.5 Pro 06-05 in place of Gemini 2.5 Pro 05-06, and the newer release solves four more problems and loses none. We compute every temporal statistic on this later-window, selection-isolated comparison.

The artifact includes the complete event streams, the selected populations, and the provider roster the replay binds by hash.

H.6 Population-Reference Intervention

We fixed the decision rule before joining any external answer. In this intervention we measure the population-level effect of updating the leave-one-out population reference on a source-balanced external task union. Twelve populations each contain four slots and the same four fixed pretrained releases: Qwen2.5-0.5B, Qwen2.5-1.5B, Llama-3.2-1B, and Gemma-2-2B (Qwen Team et al., 2024; Meta AI, 2024; Gemma Team et al., 2024). Both arms share the 64-prompt selection panel, normalized 21-dimensional arithmetic-operator representation, numeric-answer gate, launch action, serialized update opportunities, tie break, and 490-item SVAMP+ASDiv evaluation portfolio. We lock the population before joining external answers. The rule requires a positive paired mean with a two-sided 95% interval above zero and a lower confidence bound above -0.02 for mean per-slot exact match.

Updating the reference changes the selected population in 7 of 12 cases and adds 1.4 points on the external union. Live- and frozen-reference scoring select different populations in **7** of 12 paired cases. The source-balanced task-union effect is $+\mathbf{1.394}$ percentage points (paired Student-t 95% CI $[0.081, 2.707]$), and mean per-slot exact match changes by $+0.485$ points (paired Student-t 95% CI $[-0.647, 1.617]$). That interval's lower bound exceeds the fixed -2.0-point quality margin, so the panel satisfies both the external-union criterion and the quality floor. An independent standard-library verifier reproduces all twelve cells and summary statistics.

H.7 Shared-versus-Own Frozen-Candidate Factorial

All four arms replay one treatment-independent candidate log. We pin one public Qwen2.5-0.5B-Instruct checkpoint, and all twenty-four seeds share a fixed schedule of eight GSM8K prompts repeated over eight rounds and three developer streams; each menu contains 32 independently sampled candidates. We generate all 4,608 menus and 147,456 candidates before replaying any arm. Every candidate passes the finite-numeric gate, and no menu is empty. The opcode feature map takes unigram and bigram counts over the ordered chain of binary arithmetic operators a solution performs, adds a capped chain-length coordinate, and unit normalizes the result, giving 21 dimensions. Correctness is exact numeric match, and we use it only for the outcome check.

Sharing a persistent reference adds 0.148 Hill-2 effective modes and does not reduce correctness. The primary contrast we chose before the first run is final Hill-2 effective modes for a shared persistent minus an own persistent reference, and its paired mean is $+\mathbf{0.148}$ (Student-t

95% CI $[0.085, 0.210]$, $t_{23} = 4.88$, $d_z = 0.996$). Hill-1 increases by 0.155 ($[0.073, 0.236]$) and log-determinant coverage by 0.118 ($[0.047, 0.188]$). Coverage is the quantity the replay rule maximizes and the two Hill numbers count the modes it is spread over, so the three statistics summarize the same selected population and are not three independent checks of it. The correctness contrast is $+\mathbf{0.0004}$, with a one-sided 97.5% lower bound of -0.0015 that lies above the -0.01 margin (Fig. 4b); absolute correctness is 34.92% under a shared persistent and 34.87% under an own persistent reference, and correctness is the one quantity the rule never observes. We cross own versus shared reference scope with round-reset versus persistent lifetime; the four arms' absolute final Hill-2 means are 19.39 (own reset), 19.79 (shared reset), 19.99 (own persistent), and 20.14 (shared persistent). Sharing therefore raises Hill-2 under both reference lifetimes, by 0.402 ($[0.241, 0.564]$) under a round-reset reference against the 0.148 above, and making a shared reference persistent raises it by 0.347 ($[0.203, 0.491]$). The scope-by-lifetime interaction is -0.255 ($[-0.406, -0.104]$, $t_{23} = -3.49$): sharing and persistence separate the same developers, so their gains overlap. Seven of the twenty-four seeds end on an identical selected population under both persistent arms, and one further seed changes population without changing any of the three breadth statistics. For each seed the arms consume the same candidate log in the same public event order and maximize the prompt-keyed log-determinant increment with the same lexical tie break.

Two further arms bound the breadth the candidate log allows. Replaying the same panel under a reference-free rule that picks uniformly among eligible candidates, averaged over 32 draws per seed, gives 19.36 final Hill-2 modes; replaying it under an oracle that selects a candidate from the least represented mode at every insertion, breaking ties by coverage gain, gives $\mathbf{20.14}$. This greedy oracle is a strong reference point rather than a certified optimum: we do not solve for the best sequence of commitments under these menus and this order. A shared persistent reference ends at 20.14, $\mathbf{0.002}$ modes below the oracle (0.2% of the uniform-to-oracle range), and its 5.66 distinct modes per prompt match the oracle's. Of that range, a reset reference accounts for the first 4.2% above the reference-free floor, persistence for a further 76.8%, and sharing for the remaining 18.8%. Before reporting either bound, we replay the four factorial arms in the same harness and reproduce their Hill-2 means exactly.

H.8 Agent-Workflow Commit Intervention

Fresh-state commit preserves the intended final population in all three workflow fixtures. Across three fixed candidate frontiers, comprising nine arm-by-task cells, fresh-state commit passes the official Terminal-Bench and SWE-bench Pro Verified tasks while stale exact replay and the sequential control fail both; on RSI-Exam, it raises hidden reward from 0.26 to 0.35 (RSI-Exam Team, 2026; Merrill et al., 2026; Zheng et al., 2026).

The three benchmarks share one workflow. In each cell we run Qwen2.5-Coder-32B-Instruct with four workers, one integrator, temperature zero, at most twenty model requests per invocation, and $1{,}024$ output tokens per request. Fresh and stale receive the same four candidate patches byte for byte; stale acceptance replays the fresh candidates exactly. The hidden verifier runs only after we fix the candidate frontier and the terminal patch. The sequential arm uses the same model and request caps without a population-relative commit rule.

The software frontiers are executable fixtures for the commit rule. For the Terminal-Bench `mvcc-lsm-compaction` task, all four snapshot-boundary policies pass the public regression and stress reproducer, and each has unit public weight. For the SWE-bench Pro Verified Element task, the frozen frontier comprises the official solution and three predeclared perturbations of the public `FixedRollingArray` requirements; their public weights are 7, 6, 5, and 4. No hidden evaluation enters either frontier. Fresh retains the first candidate and passes both verifiers. Stale accepts all four and terminates on the fourth; its last three SWE commits lower the potential, while the three later Terminal commits have zero coverage gain against the population current at commit time. Each benchmark contributes one deterministic task, so these cells establish only the state-transition behavior of fresh versus stale commit.

Fresh commit raises hidden RSI-Exam reward from 0.26 to 0.35. The task asks an agent to replace a post-hoc chest-X-ray review policy; the sealed grader combines cross-site sensitivity, Brier

score, and the gap between mean and worst-site sensitivity under one fixed review budget, then reports a normalized reward on the benchmark's own scale, anchored so that its published baseline solver scores 0 and its SOTA point 0.6. The public gate admits all four fresh candidates, which the visible studies rank strictly.

The commit policy alone accounts for the gap: both arms replay the same four patches. The stale arm replays those four patches exactly. Fresh-state selection accepts the highest-scoring candidate and records no potential decrease. Stale acceptance admits **all four** candidates, records three potential decreases, and ends on the **lowest-scoring** one, at hidden reward $\mathbf{0.26}$; MOSAIC ends at hidden reward $\mathbf{0.35}$. The fixed-budget sequential workflow arrives at the stale arm's terminal policy by a different route and obtains the same hidden utility, so the two controls are one policy scored twice. The gain is 0.09 on a scale whose baseline-to-SOTA span is 0.6, and the commit policy alone accounts for it.

H.9 Persistent-Reference Adaptation

An own persistent reference ends at 32.42 effective modes, a shared one at 31.01, and uniform selection at 14.73. We run five arms on the same 24 matched seeds: uniform-valid selection, and a MOSAIC reference crossed as reset or persistent with a scope of the acting developer's own admits or of all three developers' admits. An own-scope arm runs the same selector once per developer on that developer's candidates and history alone; a shared arm runs it once per round on everyone's. Final pooled valid Hill-1 means are 14.73 (uniform), 21.83 (own reset), 24.76 (shared reset), $\mathbf{32.42}$ (own persistent), and 31.01 (shared persistent). Own persistent gains $\mathbf{17.69}$ modes over uniform (paired Student-t 95% CI $[13.43, 21.95]$, $t_{23} = 8.59$, $d_z = 1.75$) and shared persistent versus uniform 16.27 ($[12.11, 20.44]$, $t_{23} = 8.09$, $d_z = 1.65$). Along scope and lifetime, an own reset reference adds 7.10 modes over uniform ($[2.60, 11.61]$), making it persistent adds 10.59 ($[6.40, 14.77]$), and sharing the persistent record across the three developers adds -1.42 ($[-5.59, 2.75]$, $t_{23} = -0.70$, $d_z = -0.14$); sharing a reset record adds 2.92 ($[-1.32, 7.16]$), and the scope-by-lifetime interaction is -4.34 ($[-9.89, 1.21]$). The primary contrast we fixed before the run was shared minus own persistent, with a 3.5-mode threshold on the interval; the interval's upper bound, 2.75, falls below that threshold. Per-developer Hill-1 is 23.3 under an own and 22.6 under a shared persistent record, and the mean pairwise Jaccard overlap of the developers' mode sets is 0.29 and 0.27, against 0.38 under uniform selection.

Three Qwen2.5-0.5B-Instruct developers run for twelve rounds under each arm. Each round uses a disjoint 32-prompt GSM8K batch; each developer proposes 16 candidates and admits four; and we evaluate every round from fresh generations of the updated checkpoints. Here "valid" means that the response's final answer matches the gold answer, so modes are counted over correct responses only. Final held-out exact-match means are 36.84%, 34.90%, 35.67%, 34.97%, and 35.78% for uniform, own reset, shared reset, own persistent, and shared persistent selection; own persistent minus uniform is -1.87 points ($[-3.70, -0.03]$, $t_{23} = -2.10$, $d_z = -0.43$), a quarter of the mode gain's effect size, and shared minus own persistent is $+0.81$ ($[-0.87, 2.50]$). The held-out task union, the share of the 64 evaluation problems some developer solves at least once, is 83–85% under every arm. The uniform and two shared arms replicate an earlier family run on the same seeds with the same instrument, which ended at 17.10, 24.12, and 31.44 modes; the same-seed absolute difference between the two families has median 5.4 to 8.5 modes, so sampling at temperature 1.0 over twelve rounds, not the seed, sets the noise floor, and we compare every three-developer contrast above against that floor.

Each developer takes 24 full-model updates. Each developer fine-tunes the full bf16 model on exactly four admitted prompt–completion pairs per round. Training uses completion-only cross-entropy with equal example weight, AdamW at 10^{-5}, zero weight decay, batch size four, two epochs, a 1,024-token maximum sequence length, and gradient clipping at 1.0. This gives two optimizer steps per developer per round, or 24 over the experiment, and the next round reloads the saved checkpoint. Candidate generation uses temperature 1.0, top-p 0.95, and at most 320 new tokens. Evaluation draws eight new responses for each of the same 64 held-out GSM8K problems after every round.

The gain grows with the population: 21.41 **modes at eight developers,** 22.15 **at sixteen.** These runs belong to the earlier three-arm family and use a shared persistent reference. With eight developers, twelve rounds, and six seeds per arm, shared-persistent MOSAIC gains **21.41** modes over uniform-valid selection across matched seeds (paired Student-t 95% CI $[12.91, 29.90]$, $t_5 = 6.47$, $d_z = 2.64$). With sixteen developers, eight rounds, and six matched seeds, the gain is **22.15** modes ($[16.56, 27.74]$, $t_5 = 10.18$, $d_z = 4.16$).

Specialization rises with a persistent reference, from 0.520 **to** 0.623 **NMI.** Across six seeds per arm, final prompt-conditional normalized mutual information between developer identity and semantic cluster is 0.520 for uniform-valid selection, 0.622 for a reset reference, and **0.623** for a persistent reference: a matched-seed gain of **0.103** (paired Student-t 95% CI $[0.043, 0.162]$, $t_5 = 4.42$, $d_z = 1.81$), with a per-seed temporal slope 0.0042 NMI per round steeper than under uniform-valid selection ($[0.0017, 0.0066]$, $t_5 = 4.41$). Three frozen Qwen2.5-1.5B-Instruct developers answer the same 30 InfinityChat prompts for 16 rounds. Each prompt has its own admission slot and reference, a developer's only continuity signal is its own previously admitted response, and semantic modes come from greedy clustering of MiniLM embeddings at cosine threshold 0.8.

Table 5: **Every row tests one claim by changing one factor.** In each row we swap the reference and nothing else, except the first, where we swap the candidates, and the second, where we enumerate every improvement path at one reference. Paired 95% CIs; the RSI-Exam row replays four fixed patches and so has none.

Claim	Intervention	Result
§2.3	rewrites → distinct solutions	1.79 → 2.60 nats, +0.80 [0.77, 0.84]
Thm. 2	every path in 12 four-slot games	18,432 acyclic, worst at 0.99 of optimum
§3.3	stale → fresh-state, 18 releases; async. stress	8 → 0 cycles; potential kept in 44 of 352 stale cells → all 480 fresh
Prop. 1	standalone → own persistent, 12 rounds	15 → 32 modes, +18 [13, 22]
§6.2	own → shared persistent, 12 rounds	32 → 31 modes, −1.4 [−5.6, 2.8]
Thm. 3	launch-time → on entry; stale → fresh on RSI-Exam	82% → 85% of 131; task union +1.4 pp; reward 0.26 → 0.35

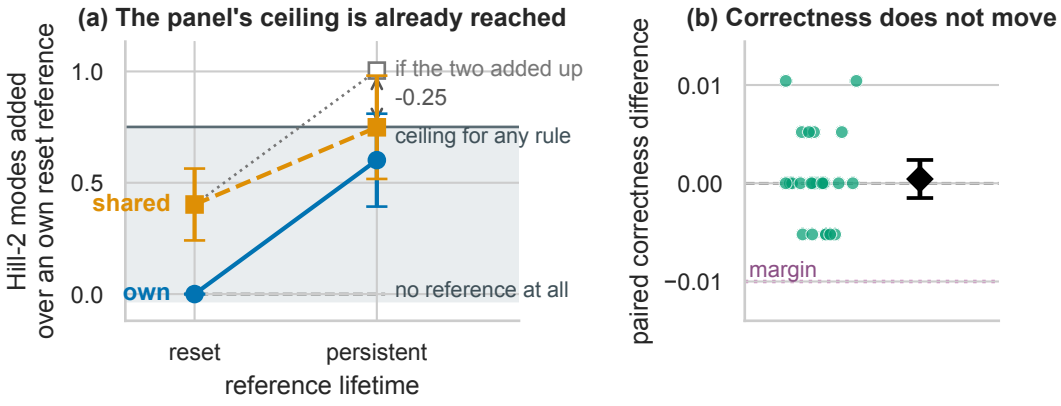


Figure 4: **A shared persistent reference comes within** 0.002 **Hill-2 modes of a greedy oracle on the same candidate log.** (a) Hill-2 effective modes each arm adds over an own reset reference, with paired 95% Student-t intervals across 24 seeds. The band spans the range from a reference-free uniform pick to a greedy oracle that takes the least represented mode at every insertion. The open marker is the additive prediction, the sharing gain plus the persistence gain, and it lies above the oracle. (b) Paired correctness difference, shared persistent minus own persistent, one point per seed, against the margin we fixed before the run.

H.10 Case-Study Protocols

Continual adaptation. Three Qwen2.5-0.5B-Instruct developers (Qwen Team et al., 2024) train on fresh 32-prompt batches for twelve rounds under uniform-valid selection and under a reference we cross as reset or persistent and as own or shared. Across 24 matched seeds the population ends with 15 pooled Hill-1 effective modes under uniform selection, 22 and 25 under an own and a shared

reset reference, and 32 and 31 under an own and a shared persistent one (Fig. 2c), the numbers from which Section 6.2 attributes the gain to the record's lifetime. We use evolving checkpoints and independently seeded candidate generations in every round; every interval, the replication of the uniform and two shared arms, frozen-weight specialization, and the eight- and sixteen-developer runs are in Section H.9.

Concurrent patch admission. In each RSI-Exam arm (RSI-Exam Team, 2026) we run Qwen2.5-Coder-32B-Instruct with four workers and one integrator at temperature zero, and in the stale arm we replay the exact four patches the fresh arm generated. A sequential workflow converges on the stale arm's terminal policy and attains the same hidden reward, 0.26, so concurrency accounts for the gap in Section 6.1 (Section H.8).

I Extended Related Work

Determinant-based LLM diversity. The within-model objectives of Section 7 are a determinant term in an RL objective (Chen et al., 2026c), leave-one-out advantages over a set-level functional (Li et al., 2026), and verifier-gated features with Shapley credit (Chen et al., 2026a); the volume they optimize is a quality-weighted kernel diversity (Friedman & Dieng, 2023; Nguyen & Dieng, 2024), and scoring a candidate by what it adds beyond a set is the criterion behind diversity-based reranking (Carbonell & Goldstein, 1998). What MOSAIC adds is a validity-gated representation, a linearizable commit protocol, persistent-reference dynamics, and interventions evaluated on the population that score optimization itself selects.

Monoculture. Monoculture traces the same collapse to a shared predictor: agents deferring to one accurate signal can leave the collective worse off (Kleinberg & Raghavan, 2021), homogenize whom they select (Bommasani et al., 2022), and narrow what a language model will generate (Wu et al., 2025); the welfare loss there is bounded by a constant (Kleinberg et al., 2026). Those results fix the signal and quantify the harm; we make developers strategic and change what a response is paid.

Dynamic and performative evaluation. A fixed test distribution has known limits, and the dynamic benchmarks answer them by refreshing the test set: adversarial human-in-the-loop rounds (Nie et al., 2020; Kiela et al., 2021), a hosted leaderboard that reruns every model rather than reading a submitted prediction file (Ma et al., 2021), post-cutoff windows that admit only items published after a model froze (White et al., 2025; Jain et al., 2025), items regenerated per run from a complexity-controlled graph (Zhu et al., 2024) or a symbolic template (Mirzadeh et al., 2025), and items a search optimizes against the models it will grade (Li et al., 2025b). Related evaluation suites and preference leaderboards are scored against a fixed key (Liang et al., 2023; Li et al., 2025a; Lambert et al., 2025). Performative prediction formalizes how a published objective changes the systems that later face it (Perdomo et al., 2020; Hardt et al., 2022); adaptation to the test task distorts both rankings and apparent emergence (Dominguez-Olmedo et al., 2025); and once post-training is strategic, the ranking becomes an equilibrium object (Chen et al., 2026b). Analyses of dynamic benchmarks without a population-relative reward isolate the refresh itself (Shirali et al., 2023). Dynamic evaluation thus expands from refreshing the test data to designing the feedback law, which is why the endpoint is a population rather than a winner.

Mechanism Design. Peer prediction pays for agreement with peers about a private signal (Miller et al., 2005; Shnayder et al., 2016; Kong & Schoenebeck, 2019) and has since been turned into a training signal for language model reasoning (Feng et al., 2026). With no belief reported and no outcome realized, properness (Gneiting & Raftery, 2007) is not the axiom at issue. The axioms at issue, marginal contribution and robustness to replication, are the ones Proposition 6 addresses (Vetta, 2002; Xu et al., 2021; Han et al., 2023). Innovation and crowdsourcing contests fix a prize schedule over a one-shot entry pool, and that schedule can promote variety (Che & Gale, 2003; Terwiesch & Xu, 2008; Chawla et al., 2019; Letina & Schmutzler, 2019); a contest fixes its prize vector before entry, while MOSAIC prices a dynamic externality whose value depends on the standing population and persists across rounds. Closest on the anti-herding objective is the minority reward of Mann & Helbing (2017), who pay accurate predictions that the majority misses and recover a stable allocation of attention with near-optimal collective accuracy. Their reward turns on a realized outcome and a binary minority event. Ours prices the coverage a model output adds once produced, then

preserves that price under asynchronous admission and repeated model updates. Against a market scoring rule, which pays each trader the increment of a potential over beliefs (Hanson, 2003; Abernethy et al., 2013), MOSAIC applies the same path-independent accounting to a concave coverage function, so the increment declines as a represented direction fills. How much a benchmark learns is itself an established question, pursued with item response theory over designer-chosen items and a scalar latent ability (Maia Polo et al., 2024; Kipnis et al., 2025). Proposition 5 is that same accounting for a vector-valued quality functional over the population's own responses; in particular, a coverage score hands the choice of items to the developers.

Leaderboard validity under strategic adaptation. A parallel line treats the score itself as the object under attack: adaptive querying of a public leaderboard overfits the holdout, and the remedies bound how much each answer reveals (Blum & Hardt, 2015; Dwork et al., 2015). Closest on manipulation is Hays et al. (2026), whose producer enters many near-identical models and whose You-Rank-We-Rank mechanism restores approximate clone robustness by having producers rank their own entries; that remedy is complementary to ours, since by Proposition 6 marginal contribution alone does not reveal common ownership of the submissions. Zhang & Hardt (2024) study diversity one level up, as an Arrow-style trade-off in aggregating an ordinal multi-task benchmark. The three methods operate at different levels: ownership of the submissions, combination of evaluation units, and MOSAIC's coverage of the response population inside one fixed unit. Scaling an evaluation improves reliability only up to the judge's own error rate (Dorner et al., 2025). We therefore treat the gate as a designer-fixed, independently checked primitive (Section 5.1). Forecasts of where the capability frontier will lie (Zhang et al., 2026) sharpen the question Proposition 1 answers: whether a score still separates models once the frontier arrives there.