SE-EVAL: VERIFIABLE BENCHMARK EVOLUTION FOR ADJACENT MODEL RELEASES

Anonymous authors
Paper under double-blind review

ABSTRACT

Static benchmarks can lose resolution precisely where rapid model development needs it most: between adjacent releases near ceiling. We introduce SE-Eval, a programmatically grounded loop that generates candidate items, verifies unique solutions, measures an ordered model-version sequence, and selects a bounded item pool for the next round. On three pre-specified Qwen seeds, SE-Eval increases adjacent-release disagreement over a frozen pool by $+0.0451$ $[+0.0251, +0.0660]$, a repeatable but sub-minimum-detectable-effect gain. A fresh-seed replication analyzed under a protocol fixed before accepted outcomes reveals a family-dependent boundary: $+0.0396$ on Qwen, -0.0893 on DeepSeek, and $+0.1673$ on Kimi with a family-disjoint generator. In a saturation diagnostic, the three newest Qwen releases score 100% on frozen items but only 83–93% on generated items, showing retained frontier headroom. A scoring-fault diagnostic shows why resolution alone is insufficient: placeholder labels reduce true-gold boundary disagreement from 0.56 to 0.28 while still producing smooth trajectories. Together, these results establish self-evolving benchmarks as longitudinal measurement instruments whose value depends on transition structure, verifiable grounding, and validation across model families.

1 INTRODUCTION

Benchmarks do not become obsolete only when their answers are wrong; they also become obsolete when they can no longer distinguish the systems they are meant to measure. This problem is acute for rapidly updated language-model families: once adjacent releases approach ceiling on a frozen item pool, the benchmark collapses several frontier versions onto the same score. Periodic refresh can restore difficulty and reduce exposure to training data (White et al., 2025; Jain et al., 2025; Chen et al., 2025a), but a system that selects new items using model behavior introduces a second problem. The benchmark is now optimizing its own measurement instrument, moving Goodhart pressure inside the evaluation loop.

We study this as a longitudinal measurement problem: how can an evaluation adapt to an ordered stream of model releases while retaining verifiable answers and interpretable comparisons across rounds? Existing dynamic benchmarks refresh items from recent sources, and automatic benchmark construction generates evaluations on demand (LI et al., 2025; Chen et al., 2025b; Shashidhar et al., 2025). SE-Eval instead uses observed adjacent-release disagreement to decide which verified items persist in a bounded pool. This feedback loop makes resolution an optimization target, so validity and cross-round linking become first-class design requirements rather than downstream checks.

SE-Eval implements a generate–verify–measure–select loop. A failure-driven generator proposes candidates near current transition bands; an executable verifier admits only unique-solution items; a meta-evaluator measures item difficulty and adjacent-version disagreement; and an evolution controller updates the pool under a fixed budget while retaining a small fixed reference subset. This design lets us test both sides of adaptive evaluation: whether evolution preserves resolution after a static pool saturates, and whether the resulting trajectory remains grounded in valid answers.

This study makes four contributions:

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.

- **A longitudinal formulation of self-evolving evaluation.** We frame benchmark evolution as a resolution–validity problem over an ordered model-version sequence, rather than as one-shot benchmark generation or periodic item refresh.
- **A programmatically grounded evolution loop.** SE-Eval connects failure-driven generation, executable unique-solution verification, item-level measurement, and bounded pool selection in a reproducible closed-loop system (§3).
- **A protocol-fixed characterization of where evolution helps.** With the analysis protocol fixed before accepted outcomes were examined, fresh-seed replications show family-dependent resolution gains: Qwen exhibits a repeatable but sub-MDE positive gain, DeepSeek is negative, and Kimi reaches $+0.1673$ with a family-disjoint generator. A post-hoc matched-retirement analysis further finds that all ten realized selection trajectories exceed their corresponding random-retirement null means (§4.4).
- **An empirical Goodhart failure for adaptive benchmarks.** When generated items are scored against unverified placeholder labels, the loop produces smooth trajectories while reducing true-gold boundary disagreement from 0.56 to 0.28. This exposes verifiable grounding as a design requirement for closed-loop evaluation (§4.5).

2 Related Work

Prior work differs in which component changes over time and how validity is assessed. We group it into benchmarks that refresh or transform their items, systems that construct benchmarks on demand, and systems that update components of the evaluation pipeline. SE-Eval integrates generation, measurement, meta-evaluation, and evolution with programmatic admission gating and a retained external reference set.

Benchmark refresh and task variation. LiveBench (White et al., 2025) and LiveCodeBench (Jain et al., 2025) periodically incorporate newly released items from recent or time-indexed sources to reduce contamination risk. EvoEval (Xia et al., 2024) derives coding tasks from existing ones through LLM-based semantic transformations, and the survey by Chen et al. (2025a) describes the broader transition from static to dynamic evaluation. We study whether adjacent-version discrimination can guide pool evolution under an explicit external reference; the *rand* baseline isolates refresh without the selection signal (§4.3).

On-demand automatic benchmark construction. AutoBencher (LI et al., 2025) constructs a benchmark declaratively from a topic specification. DyCodeEval (Chen et al., 2025b) dynamically constructs code-reasoning evaluations under data contamination, while YourBench (Shashidhar et al., 2025) generates document-grounded evaluation sets on demand. SE-Eval repeats the construction and evaluation cycle, using measured item behavior to inform subsequent selection while retaining a fixed reference set.

Evolving the evaluation pipeline itself. Benchmark Self-Evolving (Wang et al., 2025) uses multiple agents to expand a benchmark autonomously. DynaBench (Kiela et al., 2021) uses model failures to guide iterative human-and-model-in-the-loop data collection. Other systems evolve evaluators from observed blind spots (Zhang et al., 2026), evaluate world-model rollouts through hierarchical tool-assisted analysis and verifiable evidence in visual worlds (Chen et al., 2026), synthesize agent trajectories through test-time exploration (Guo et al., 2026), or jointly evolve a deep-research evaluator and solver (Zhu et al., 2026). SE-Eval examines self-referential drift in a persistent item-pool feedback loop, where generated items are selected using signals from the evolving measurement system, and uses a retained verifiable reference set for external grounding. Relative to Benchmark Self-Evolving (Wang et al., 2025), SE-Eval applies repeated evolution to a public model-version sequence and includes a program-decidable validity constraint, retirement and controller dynamics, and a version-adjacent resolution objective.

Psychometrics and adaptive testing. IRT and computerized adaptive testing select calibrated items for efficient latent-ability estimation, often using information-based criteria (Van der Linden et al., 2000; Lord, 2012); ATLAS applies a psychometric adaptive-testing perspective to LLM evaluation (Li et al., 2025). Test equating and scale linking can use common or anchor items to support

comparability across forms (Kolen & Brennan, 2014; Lord & Novick, 2008), while measurement invariance (Millsap, 2012) characterizes the conditions under which such comparisons are valid. Active learning provides general acquisition strategies for allocating labeling effort (Settles, 2009), whereas active testing adapts related ideas to sample-efficient model evaluation (Kossen et al., 2021). SE-Eval complements psychometric calibration by targeting a different object: observed adjacent-release resolution in a changing, programmatically verified item pool. Psychometric linking and invariance theory provide natural tools for extending this empirical objective toward formal cross-round comparability (Appendix B.1).

Positioning. SE-Eval combines continuous generation, selection over pool persistence, a retained verifiable reference subset, and longitudinal comparison across a model-version sequence (Appendix B).

3 METHOD: SE-EVAL

3.1 PROBLEM SETUP AND NOTATION

The evaluated object is a *version sequence* of models $\mathcal{M} = (M_1, \ldots, M_T)$ from one family (e.g., consecutive public releases). An evaluation system maintains an item pool I_t and scores a model by executable-verification accuracy $s_t(M) = \mathrm{acc}(M; I_t)$. The design target is the pool's *adjacent-version resolution*: how reliably it separates neighboring releases behaviorally. We instantiate it as mean adjacent-version disagreement over the pool and summarize its evolution as mean trajectory height over a common time span (the "AUC gap"). Per-item difficulty and discrimination are descriptive observed-response statistics rather than item-response-theory parameters; Appendix B.1 connects this objective to formal linking and calibration. We compare it against two reference systems: a *static* pool ($I_t \equiv I_0$) and a *rand* pool that replaces items without using quality signals. The adaptive conditions share a generation-and-evaluation budget; the static condition requires no generation and is reported with its lower, pre-specified cap.

Let $x_i(M_j) \in \{0, 1\}$ denote executable correctness for item i and version M_j. The item's observed adjacent-release resolution is

$$r_i = \frac{1}{T-1} \sum_{j=1}^{T-1} |x_i(M_{j+1}) - x_i(M_j)|, \qquad R(I_t) = \frac{1}{|I_t|} \sum_{i \in I_t} r_i.$$

SE-Eval seeks pools with high $R(I_t)$ subject to three constraints: every admitted item has a programmatically verified answer, pool size remains bounded, and retirement remains stratified by observed difficulty rather than collapsing onto a single high-disagreement pattern. The reported trajectory statistic averages $R(I_t)$ over the common observed time span. This objective rewards items near behavioral transition bands; it does not assume that later releases are always better, so we separately inspect the signed direction of disagreements.

3.2 THE GENERATE–MEASURE–META-EVALUATE–EVOLVE LOOP

SE-Eval is a closed loop of four modules (Figure 1).

Failure-driven generator g. Given the failure-mode set F_t of the current frontier pair (item categories where the newer version still errs or the two versions disagree), the generator proposes candidate items I^+, stratified by target difficulty to sample both separating and weakly separating regimes. In the controlled runs, the generator belongs to a different model family from the evaluated sequence. Empty or truncated generations are discarded and retried.

Executor (measurement). Every candidate is *programmatically verified*: only items with a decidable unique solution enter the pool, and every model answer is scored by execution, not by judge opinion. This makes s_t exact and reproducible, and removes judge drift from the measurement layer.

Meta-evaluator m. Each item receives difficulty d_i and discrimination a_i (derived from adjacent-version disagreement and solve rates over the current pool). These signals drive retirement through

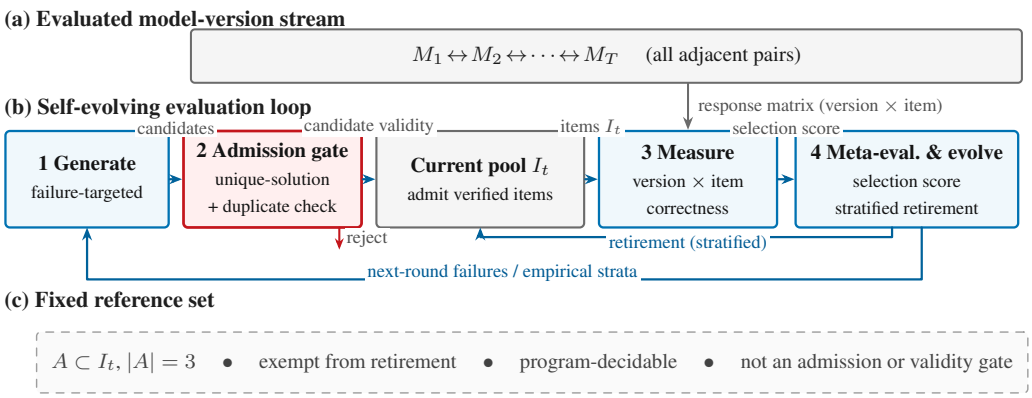


Figure 1: **The implemented SE-Eval loop.** The ordered model stream supplies all adjacent version pairs to measurement on the current pool I_t. Failure-driven generation proposes candidates; the programmatic verifier is the admission gate, accepting only unique-solution items without duplicates, and rejected candidates leave the loop. Measurement produces item-level outcomes (the response matrix), whose selection score drives stratified retirement and the next round's targets. A small fixed reference subset $A \subseteq I_t$ is retained unchanged as a cross-round pool invariant and is exempt from retirement; candidate admission is controlled by the verifier.

the controller; they are the loop's feedback channel. The implemented controller does not use a per-item anchor-consistency score c_i as an admission criterion.

Evolution controller u**.** Under a fixed token budget for each adaptive condition, the controller adds verified candidates, retires items by stratified sampling intended to preserve difficulty coverage (avoiding collapse onto the hardest stratum), and keeps anchors exempt from retirement. The stopping criteria are budget exhaustion, a conclusive pre-specified confidence interval, or two consecutive rounds without significant gain.

Versioned measurement state. An evolving benchmark is reproducible only if the instrument itself is versioned. Each round therefore records candidate provenance, verifier outcomes, pool membership before and after retirement, the item-by-model response matrix, and the statistics used by the controller. This record makes the feedback loop auditable at item level: a reported trajectory can be traced to the answers that entered it, and alternative retirement rules can be replayed on the same realized candidate supply. The fixed references provide a persistent within-study check, while the complete state history preserves the changing part of the instrument.

3.3 A SMALL FIXED REFERENCE SET

The anchor set $A \subseteq I$ consists of fixed reference items whose correctness is decidable by program execution and does not depend on generator decisions. In the runs analyzed here, the anchors' implemented runtime roles are: (i) they are exempt from retirement, so the pool always contains a fixed, externally verified reference, and (ii) the controller's difficulty strata are computed over this anchored pool; candidate admission itself is gated by the programmatic unique-solution verifier and the duplicate gate (§4.1). After the runs, two raters independently assessed a randomized sample for clarity, perceived solvability, and duplication. This human-readability audit is separate from the online controller and complements executable answer verification.

3.4 ANALYSIS DESIGN

The primary Qwen analysis was specified before confirmatory data were examined. It includes four conditions (the full loop, *static*, *rand*, and the anchor-free ablation), matched rounds and item-generation schedules for the adaptive conditions, multiple generation seeds, a pre-specified primary metric (AUC-gap difference) and minimum detectable effect, item-level bootstrap CIs, Holm correction, and explicit treatment of implementation deviations. The three-version DeepSeek sequence

constitutes a robustness replication because unavailable intermediate versions prevented the protocol's planned minimum of four versions. Section 4.1 describes the protocol details and implementation deviations.

4 EXPERIMENTS

4.1 SETUP

Task domains and evaluated sequences. The controlled study uses executable knights-and-knaves puzzles, where unique-solution validity and reference answers are decidable by program; rendering clarity and human interpretability are assessed separately. We evaluate ordered public releases from **Qwen** (5 versions) and **DeepSeek** (3 versions), with kimi-k3 as a generator and meta-evaluator disjoint from both families. An exploratory Kimi sequence initially shares its final model with the generator and is therefore treated only as a coupled stress test (Appendix E.8). The fresh-seed replication assigns a generator outside each evaluated family and is analyzed under a protocol fixed before accepted outcomes were examined. A one-seed exploratory transfer replaces the puzzle verifier with executable Python unit tests while retaining the Qwen sequence and controller.

Confirmatory protocol. The confirmatory Qwen design and analysis plan were specified before the confirmatory data were examined. Four conditions—SE-Eval (closed-loop self-evolution with verifiable anchors), *static* (fixed pool), *rand* (random item replacement), and *noanchor* (anchor-free ablation: no anchors, no admission verifier, naive retirement)—run symmetrically at 16 generated items per round, 3 rounds, with 3 pre-specified generation seeds. Two further seeds were collected after the primary three-seed results had been examined and are therefore post-hoc augmentation: the confirmatory estimate rests on the three pre-specified seeds, and the five-seed pooled estimate is a robustness analysis. The DeepSeek sequence runs 3 pre-specified seeds and two post-hoc exploratory augmentation seeds collected after inspection of the initial results to probe its vs.-rand null boundary. A gateway-forced generation-temperature change ($0.9 \rightarrow 1.0$) applies to this sequence; the augmentation and temperature change are protocol deviations. Worker-level token caps were condition-specific: 0.8M for static, which performs no generation or meta-evaluation, and 1.2M for each adaptive condition. These unequal caps preclude compute-efficiency comparisons. Budget exhaustion is an experiment-level stopping criterion; the seed-1 noanchor worker stopped during round 3 and contributes its two complete rounds, as specified in the analysis plan. All model access goes through a single versioned gateway; item deduplication, leakage checks, and caching are applied identically across conditions. Because unavailable intermediate releases reduced DeepSeek to three versions, it is a robustness replication rather than a second confirmatory sequence.

4.2 STATIC POOLS LOSE FRONTIER RESOLUTION

The frozen pool maps the three newest Qwen releases to the same 100% score, compressing their behavior into a single ceiling. Generated items retain 7–17 percentage points of headroom: the same releases score 83–93% on the generated stream. The signed final-pool staircase makes the contrast explicit: static yields $[0.29, 0.31, 1.00, 1.00, 1.00]$, whereas the evolved pool yields $[0.07, 0.07, 0.86, 0.93, 0.95]$. Evolution does not create the version ordering; it retains measurable density around transitions that the frozen pool no longer resolves. Because item difficulty differs between pools, this is evidence of saturation and headroom rather than a direct estimate of contamination.

The retained signal is localized at transition bands. The aggregate gain is not produced by uniformly harder items. In the final Qwen pools, 76% of generated items distinguish the 2026-01$\rightarrow$3.5-plus boundary, 15% distinguish 3.5-plus$\rightarrow$3.7-max, and 2% distinguish the newest pair. At the dominant boundary, 98.6% of those disagreements favor the newer release; at the next boundary, 87.5% do. Item-level composition tells the same story: items retired by the controller have mean disagreement 0.157, whereas 79 content-addressed unique-within-run additions average 0.228; counting all 89 added pool rows in the five-run diagnostic gives 0.230. Thus the loop primarily reallocates a fixed-capacity pool toward regions where adjacent releases actually change behavior. The concentration also explains why a family can show little or negative aggregate gain: if its available candidates do not populate useful transition bands, selection has no resolution to retain. These mechanism

Table 1: **SE-Eval versus the static pool across evidence tiers.** Δ is the difference in mean adjacent-version-gap trajectory height. Intervals are conditional item-bootstrap 95% intervals and do not include seed- or family-level uncertainty. Original Qwen augmentation seeds and original DeepSeek seeds 4–5 are post hoc; replication generators are disjoint from the family being evaluated. The pre-specified MDE is 0.05.

Stage	Family	Evidence tier	n	Δ vs. static	Conditional 95% interval	Seed signs / MDE
Original	Qwen	pre-specified	3	$+0.0451$	$[+0.0251, +0.0660]$	3+/0−; below
Original	Qwen	pooled robustness	5	$+0.0462$	$[+0.0307, +0.0625]$	5+/0−; below
Original	DeepSeek	robustness	5	-0.0425	$[-0.1021, +0.0172]$	1+/4−; below
Fresh-seed replication	Qwen	fixed pre-analysis	3	$+0.0396$	$[+0.0193, +0.0610]$	3+/0−; below
Fresh-seed replication	DeepSeek	fixed pre-analysis	3	-0.0893	$[-0.1671, -0.0136]$	0+/3−; below
Fresh-seed replication	Kimi	fixed pre-analysis	3	$+0.1673$	$[+0.1139, +0.2198]$	3+/0−; above

statistics are post-hoc descriptive analyses; the matched-retirement replay below provides the direct controlled comparison of the persistence rule.

4.3 Resolution gains are family-dependent

Our primary quantity is the *adjacent-version gap*: the fraction of items on which two adjacent versions disagree on correctness. Its trajectory AUC measures behavioral resolution, not signed capability growth; a post-hoc direction battery confirms that 98.6% of disagreements at the dominant Qwen boundary favor the newer version, with stable version ordering (Appendix E.2). We compare mean trajectory heights on a common time span, using the generation seed as the highest exchangeable unit. Table 1 separates the original evidence tiers from the protocol-fixed fresh-seed replication; estimates from the two stages are never pooled.

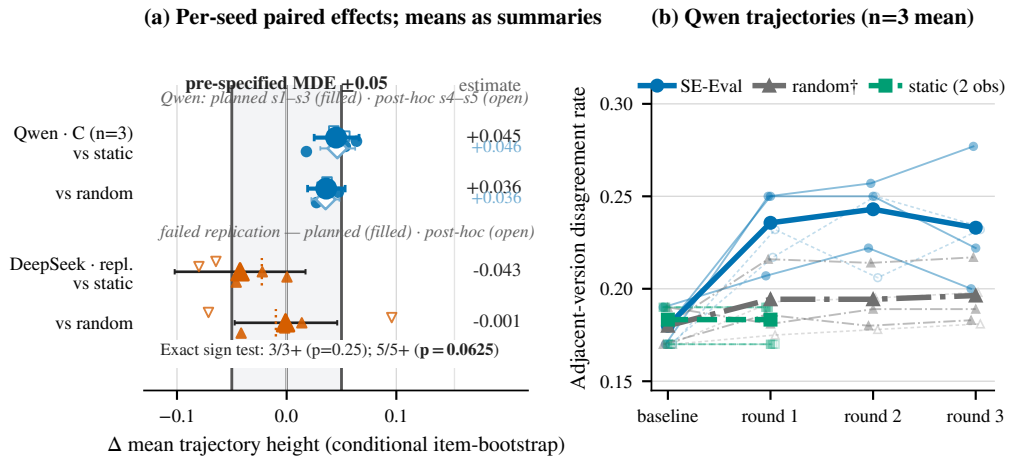
Original study. On Qwen, all three pre-specified seeds are positive: SE-Eval raises the disagreement trajectory by $+0.0451$ $[+0.0251, +0.0660]$ over static and $+0.0361$ $[+0.0191, +0.0533]$ over undirected refresh. Both point estimates are below the 0.05 MDE. Two post-hoc augmentation seeds preserve the direction, giving pooled n=5 estimates of $+0.0462$ $[+0.0307, +0.0625]$ versus static and $+0.0355$ $[+0.0225, +0.0481]$ versus refresh. Exact two-sided seed sign tests are p=0.25 for the pre-specified seeds and p=0.0625 for the pooled robustness set. The item-bootstrap intervals condition on the realized seeds and therefore summarize item uncertainty rather than replacing seed-level inference.

DeepSeek marks the first family boundary: the original estimate is -0.0226 on its three planned seeds and -0.0425 $[-0.1021, +0.0172]$ after the two post-hoc seeds, while its comparison with undirected refresh is essentially null (-0.0010 pooled). The coupled exploratory Kimi run is reported in Appendix E.8; because its final evaluated version also generated the items, it is not used as independent validation.

Fresh-seed replication under a fixed protocol. The protocol and prospectively logged amendments were fixed before accepted replication outcomes were analyzed (Appendix E.9). Three fresh generation seeds were run per family, with a generator disjoint from the family being evaluated. Qwen again shows a positive but sub-MDE gain ($+0.0396$ $[+0.0193, +0.0610]$), close to its original pre-specified estimate. DeepSeek is negative across all three fresh seeds (-0.0893 $[-0.1671, -0.0136]$). Kimi shows the largest gain ($+0.1673$ $[+0.1139, +0.2198]$), above the MDE, while retaining its temperature-1.0 sampling caveat. Holm-adjusted one-sided conditional item-bootstrap tests reject the positive null for Qwen and Kimi but not DeepSeek. These tests condition on the realized seeds; exact one-sided seed sign tests remain underpowered at n=3 (p=0.125 for each all-same-sign family). The replication therefore sharpens a family-dependent boundary rather than establishing a universal gain.

4.4 Selection beats matched random retirement

The original *rand* baseline measures undirected churn under a larger pool and omits quality gating, so it does not isolate the controller's retirement policy. We therefore perform a post-hoc content-addressed replay that fixes each realized candidate stream, accepted set, capacity, retirement count,



CIs: conditional item-bootstrap (10k draws; item within generation seed, aggregated across seeds) — not seed- or family-level CIs.
Markers: filled = pre-specified s1–s3; open = post-hoc s4–s5. Means are summaries only. † Random not capacity matched.

Figure 2: **Original-study effects and Qwen trajectories. (a)** Markers show paired seed effects for Qwen and DeepSeek; intervals summarize conditional item-bootstrap uncertainty. Filled markers are pre-specified seeds and open markers are post-hoc augmentation seeds. The larger Qwen summaries distinguish the pre-specified n=3 estimate from the pooled n=5 robustness estimate. The shaded band marks the MDE ± 0.05. **(b)** Thin lines are individual Qwen seeds; thick lines average the three pre-specified seeds. Static is sampled at two time points, and the original random-refresh baseline is not capacity matched. Fresh-seed replication results are reported in Table 1.

and evaluation budget, randomizing only which eligible items retire (Appendix E.7). All ten realized SE-Eval trajectories exceed their matched-null means (p=0.00195, exact sign test). On the observed streams, behavior-guided persistence therefore retains more adjacent-release resolution than random retirement.

This replay answers a narrower and more informative question than the original random-refresh baseline. It does not ask whether the complete closed loop beats every alternative generator; it asks whether, after candidate generation and verification have produced a particular supply, the controller keeps a more informative bounded subset than matched random retirement. Holding the realized supply fixed removes generation success, admission rate, and pool capacity as explanations for this comparison. It does not isolate the separate contributions of failure-driven generation, difficulty stratification, and the retained reference subset.

4.5 Invalid labels create plausible but wrong trajectories

The bundled ablation exposes a consequential Goodhart failure. It admits generated items without programmatic answer verification and scores them against an all-knights placeholder label. The resulting runtime curves remain smooth even though they no longer measure disagreement with the true answers. On the seed-1 final pool, re-evaluating the same boundary models against solver-derived gold changes generated-item disagreement from 0.28 to 0.56; the verified seed pool gives 0.68. A closed-loop benchmark can therefore optimize a stable-looking trajectory while tracking the scoring mechanism rather than the intended construct.

Program validity and human interpretability. The offline unique-solution audit remains near-perfect in the ablated condition, showing that structural solvability alone does not detect the label failure. In the full system, all 32 sampled generated items pass the executable unique-solution check, while two raters judge 26/32 and 30/32 human-solvable (Table 2). The gap separates programmatic answer validity from human interpretability and motivates reporting both.

Scope of the diagnostic. The bundled condition jointly removes the fixed references and admission verifier and replaces stratified retirement with a naive rule. It therefore diagnoses a scoring

Table 2: **Item-quality evaluation by two raters (sequence A; exploratory).** Per-row n and raw counts; program validity and human interpretability are separate constructs. On generated-item solvability the raters agree on 28/32: four disagreements, all on the lowest-clarity items (operator-precedence ambiguity), Cohen's κ=0.448 (item-bootstrap 95% CI $[0.0, 0.871]$). True near-duplicates: $0/46$ reference, $4/32$ generated (polarity-flip/logically equivalent variants; same-item cross-round persistence excluded, by design). The programmatic check was re-run on generated items only (reference items are the pre-verified frozen pool). Ablation-side numbers are reported in the text (its trajectories are scoring artifacts).

	Reference (seed/anchor) items	**Generated items**
Program validity: unique-solution verifier pass	— (pre-verified pool)	32/32 (100%)
Human interpretability: clarity, 1–5 (rater 1/2)	4.28/4.86 (n=46/44)	3.41/3.47 (n=32)
Human-perceived solvability (rater 1/2)	46/46 (100%) / 44/44 (100%)	26/32 (81%) / 30/32 (94%)
Solvability disagreements (pairs); Cohen's κ [bootstrap 95% CI]	0/44; 1.0	4/32; 0.448 $[0.0, 0.871]$
True near-duplicates	0/46, 0/44 (0%)	4/32 (12.5%)

failure, not the causal effect of any single component. A clean $2 \times 2 \times 2$ factorial must vary references, verification, and retirement independently while holding candidate supply and capacity fixed; the first exploratory batch is summarized in Appendix E.11.

The scoring-independent unique-solution and duplicate assessments are unchanged under content-level deduplication based on statements and clues. Template-hash deduplication overcounts duplicates and is excluded from the reported analysis. The ablated runtime trajectory is excluded from both normalizations because it uses placeholder-label scoring.

Difficulty coverage evolution (exploratory). Appendix E.6 (Figure 3) shows seed-resolved pool composition under three fixed descriptive solve-rate bins, as change from each seed's baseline share on a shared scale—a descriptive diagnostic, not causal evidence about stratified retirement; generated items are exclusively 2–3 person puzzles using mixed connectives (full distribution in the appendix).

Exploratory code-domain check. A one-seed transfer to executable Python-function tasks uses the same Qwen sequence and controller with unit-test verification. SE-Eval increases trajectory height over static by $+0.0503$ $[+0.0005, +0.0998]$ (item bootstrap), with the gap rising from 0.070 to 0.130 while static remains at 0.040. This directionally consistent result shows that the implementation transfers to a second verifier, but one seed provides no cross-domain inference.

Cross-family transfer (exploratory). To probe whether evolved pools exploit family-specific artifacts (selection on the very metric reported), we score pools evolved against one family with the other family, against the static control. Raw discrimination does not transfer: the Qwen-evolved pools score 0.190 mean adjacent-version gap on the DeepSeek family vs. 0.280 for the static pool (Δ=-0.089), and DeepSeek-evolved pools on Qwen are at parity (0.210 vs. 0.200, Δ=$+0.010$). The static control, however, is degenerate in both directions: the three newest Qwen versions and both newer DeepSeek versions sit at 100% on the frozen pool, so its apparent gap collapses onto a single saturation cliff with zero separation between the newest pairs, whereas evolved pools retain nonzero separation on those pairs (0.096 flash→pro; 0.097 for 3.5-plus→3.7-max). These results indicate that evolution trades the saturated cliff for continued headroom on recent versions within-family, without conferring a generalizable cross-family advantage.

5 Discussion

Benchmark evolution is most useful at transition bands. The results locate the value of self-evolving evaluation more precisely than a universal improvement claim. Qwen shows a repeatable but small positive gain, DeepSeek is negative, and Kimi shows a substantially larger gain in the family-disjoint replication. Together with the pairwise trajectories, this pattern suggests that evolution helps when the candidate stream can populate behavioral transition bands that a frozen pool no longer covers. It cannot create resolution when the evaluated family supplies no useful transition structure.

Selection contributes beyond undirected turnover. The post-hoc matched-retirement analysis holds the realized candidate stream, accepted items, capacity, retirement counts, and evaluation budget fixed while randomizing only which items retire. All ten realized SE-Eval trajectories exceed their matched-null means (p=0.00195). This isolates a concrete role for behavior-guided persistence: on these realized streams, the controller retains more resolution than random retirement.

Resolution is meaningful only under valid scoring. The placeholder-label diagnostic exposes the central Goodhart risk of a closed-loop benchmark. A system can produce smooth, apparently stable trajectories while optimizing against labels that do not represent the intended answers. Re-evaluation against true gold changes boundary disagreement from 0.28 to 0.56. Programmatic verification is therefore not merely an implementation convenience; it makes the loop's feedback signal auditable.

Three layers of evidence are needed. The experiments separate three questions that are easily conflated in adaptive evaluation. First, *answer validity*: does each score correspond to the intended task? The placeholder-label failure shows that plausible aggregate curves do not answer this question; executable verification does. Second, *within-study continuity*: can a changing pool be reconstructed and compared across rounds? Retained references and versioned response matrices provide an auditable invariant, although they are not yet a formal psychometric linking design. Third, *transport*: does a gain survive new generation seeds, model families, and task verifiers? The fresh-seed replication and executable-code transfer address progressively broader parts of this question while preserving their distinct evidence tiers. Reporting these layers separately prevents local resolution gains from being mistaken for universal benchmark validity.

Implication for adaptive benchmark design. A useful self-evolving benchmark should therefore report retained resolution near the current frontier together with answer-verification evidence, cross-round provenance, and uncertainty across independent generation seeds and model families. It should also preserve enough state to replay selection against matched alternatives. These requirements turn benchmark evolution from an opaque content-generation pipeline into a measurement procedure whose gains and failures can be inspected directly. Resolution without grounding and transport checks can instead produce a precise instrument for the wrong construct.

6 Conclusion

SE-Eval reframes dynamic benchmarking as longitudinal measurement over an ordered stream of model releases. Its programmatically grounded generate–verify–measure–select loop retains headroom where a frozen pool maps recent releases to the same ceiling score. Across the original study and fresh-seed replication, the gain is sharply family-dependent: repeatable but sub-MDE on Qwen, negative on DeepSeek, and above the pre-specified margin on Kimi with a family-disjoint generator. A post-hoc matched-retirement analysis shows that behavior-guided selection retains more resolution than random retirement on the same realized streams, while the placeholder-label diagnostic shows that smooth adaptive trajectories can still be wrong. The resulting design principle is simple: benchmark evolution is useful when it targets genuine transition bands, but its feedback loop must remain tied to verifiable answers and be evaluated across independent model families.

Limitations

SE-Eval measures adjacent-version disagreement, not signed capability growth. Estimates use few generation seeds: the fresh-seed replication has n=3 per family, item-bootstrap intervals omit seed- and family-level uncertainty, and exact seed-level tests remain underpowered. The results therefore do not support universal improvement. Controlled evidence comes from one executable puzzle domain; a one-seed code-domain transfer is directionally consistent, not evidence of cross-domain generality. Kimi evaluation retains a temperature-1.0 sampling caveat. The bundled ablation jointly changes references, admission verification, and retirement and uses placeholder labels; it diagnoses a scoring failure, not component causality. Unequal token caps preclude compute-efficiency claims, and agreement on generated-item solvability is moderate (κ=0.448).

Ethics Statement

The benchmark instances are synthetic logic puzzles with programmatically generated labels; the study uses no private, sensitive, or personally identifying data. Two raters (an author and a collaborator) independently evaluated puzzle clarity, solvability, and duplication. Only aggregate judgments are reported, and the activity did not involve a behavioral intervention or collection of sensitive attributes. The principal downstream risk is that automatically generated benchmarks can amplify generator artifacts, ambiguity, or bias while appearing objective. SE-Eval mitigates this risk through executable validation, a fixed programmatically verified reference set, a post-hoc human content evaluation, and retained versioned records. Domain-specific human oversight remains necessary when the method is applied outside programmatically verifiable tasks.

Reproducibility Statement

We specify the evaluated model sequences, conditions, budgets, stopping rules, primary metric, and uncertainty procedure in Section 4.1. Appendix C and Appendix E describe the implementation and all changes to or departures from the specified procedures. Versioned configurations, generation and evaluation scripts, raw worker-level outputs, pool snapshots, and analysis summaries are retained with manifests and SHA-256 hashes. Reported confidence intervals are regenerated from the raw item-level results using a fixed bootstrap seed and 10,000 resamples. Post hoc analyses of the exemption replay, the true-gold recheck of the ablated condition, and the matched-capacity baseline switch `SEEVAL_FIXED_POOL` are implemented as deterministic scripts over the retained raw outputs. The content-addressed matched random-retirement replay (Appendix E.7) is reproducible from the archived pool snapshots, evaluation caches, and re-evaluation outcomes; the scripts terminate without producing statistics when an identity cannot be resolved. Together, the appendices and the retained inputs provide a complete path from the raw data to the tables and reported statistics.

AI Use Statement

Generative AI tools contributed to synthetic benchmark-item generation, conceptual and experimental-design development, method implementation, data reformatting, qualitative-assessment synthesis, interpretation, literature discovery, code editing, artifact preparation, drafting, translation, and language polishing. Synthetic items underwent executable verification and the reported human evaluation. AI-assisted code was executed and checked against versioned outputs, and citation metadata, numerical claims, and manuscript text were checked against source records and experiment artifacts. The authors reviewed and take responsibility for the final text, claims, code, data, and artifacts.

References

Simin Chen, Yiming Chen, Zexin Li, Yifan Jiang, Zhongwei Wan, Yixin He, Dezhi Ran, Tianle Gu, Haizhou Li, Tao Xie, et al. Benchmarking large language models under data contamination: A survey from static to dynamic evaluation. In *Proceedings of the 2025 Conference on Empirical Methods in Natural Language Processing*, pp. 10091–10109, 2025a.

Simin Chen, Pranav Pusarla, and Baishakhi Ray. Dynamic benchmarking of reasoning capabilities in code large language models under data contamination. *arXiv preprint arXiv:2503.04149*, 2025b.

Weiliang Chen, Haowen Sun, Jun Gao, Jiawei Chi, Hanyang Wang, Qiyu Dai, Yihao Li, Hao Li, Jingnan Gao, Yi-Hsin Hung, et al. Harnesseval-w: Agentifying the evaluation of visual worlds. *arXiv preprint arXiv:2608.16859*, 2026.

Yixiong Fang, Tianran Sun, Yuling Shi, Min Wang, and Xiaodong Gu. Lastingbench: Defend benchmarks against knowledge leakage. In *EMNLP (Findings)*, pp. 18304–18317, 2025.

Dadi Guo, Tianyi Zhou, Dongrui Liu, Chen Qian, Qihan Ren, Shuai Shao, Zhiyuan Fan, Yi R Fung, Kun Wang, Linfeng Zhang, et al. Towards self-evolving agent benchmarks: Validatable agent trajectory via test-time exploration. In *The Fourteenth International Conference on Learning Representations*, 2026.

Naman Jain, 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 *International Conference on Learning Representations*, volume 2025, pp. 58791–58831, 2025.

Douwe Kiela, Max Bartolo, Yixin Nie, Divyansh Kaushik, Atticus Geiger, Zhengxuan Wu, Bertie Vidgen, Grusha Prasad, Amanpreet Singh, Pratik Ringshia, et al. 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, 2021.

Michael J. Kolen and Robert L. Brennan. *Test Equating, Scaling, and Linking: Methods and Practices*. Springer, 3 edition, 2014. doi: 10.1007/978-1-4939-0317-7.

Jannik Kossen, Sebastian Farquhar, Yarin Gal, and Tom Rainforth. Active testing: Sample-efficient model evaluation. In *International Conference on Machine Learning*, pp. 5753–5763. PMLR, 2021.

Peiyu Li, Xiuxiu Tang, Si Chen, Ying Cheng, Ronald Metoyer, Ting Hua, and Nitesh V Chawla. Adaptive testing for llm evaluation: A psychometric alternative to static benchmarks. *arXiv preprint arXiv:2511.04689*, 2025.

XIANG LI, Farzaan Kaiyom, Evan Liu, Yifan Mai, Percy Liang, and Tatsunori Hashimoto. Autobencher: Towards declarative benchmark construction. In *International Conference on Learning Representations*, volume 2025, pp. 94357–94386, 2025.

Frederic M Lord. *Applications of item response theory to practical testing problems*. Routledge, 2012.

Frederic M Lord and Melvin R Novick. *Statistical theories of mental test scores*. IAP, 2008.

Roger E Millsap. *Statistical approaches to measurement invariance*. Routledge, 2012.

Burr Settles. Active learning literature survey. Computer Sciences Technical Report 1648, University of Wisconsin–Madison, 2009.

Sumuk Shashidhar, Clémentine Fourrier, Alina Lozovskia, Thomas Wolf, Gokhan Tur, and Dilek Hakkani-Tür. Yourbench: Easy custom evaluation sets for everyone. *arXiv preprint arXiv:2504.01833*, 2025.

Wim J Van der Linden, Cees AW Glas, et al. *Computerized adaptive testing: Theory and practice*, volume 13. Springer, 2000.

Siyuan Wang, Zhuohan Long, Zhihao Fan, Xuan-Jing Huang, and Zhongyu Wei. Benchmark self-evolving: A multi-agent framework for dynamic llm evaluation. In *Proceedings of the 31st international conference on computational linguistics*, pp. 3310–3328, 2025.

Colin White, Samuel Dooley, Manley Roberts, Arka Pal, Benjamin Feuer, Siddhartha Jain, Ravid Shwartz-Ziv, Neel Jain, Khalid Saifullah, Sreemanti Dey, et al. Livebench: A challenging, contamination-limited llm benchmark. In *International Conference on Learning Representations*, volume 2025, pp. 91595–91631, 2025.

Chunqiu Steven Xia, Yinlin Deng, and Lingming Zhang. Top leaderboard ranking= top coding proficiency, always? evoeval: Evolving coding benchmarks via llm. *arXiv preprint arXiv:2403.19114*, 2024.

Xing Zhang, Yanwei Cui, Guanghui Wang, Zhihao Lin, and Peiyang He. Metrics that write themselves: Evolving an evaluator from its own blind spots. *arXiv preprint arXiv:2608.18744*, 2026.

Han Zhu, Chengkun Cai, Yuanfeng Song, Xing Chen, Sirui Han, and Yike Guo. Self-evolving deep research via joint generation and evaluation. *arXiv preprint arXiv:2606.04507*, 2026.

Table 3: **Index of analyses in this appendix, grouped by role in the evidence.**

Role	Analysis	Key finding
Pre-specified	Main comparison (Table 1)	Positive but sub-MDE on Qwen (n=3, p=0.25); DeepSeek does not replicate (Table 1)
Pre-specified	Bundled ablation	Offline unique-solution rate ≈1.0; runtime trajectories use placeholder labels and do not estimate true-answer disagreement
Pre-specified	Saturation/headroom comparison (Table 7)	2026-era: 100% on the frozen pool vs. 83–93% on the generated stream (headroom 0.93)
Pre-specified	Human evaluation (Table 2)	Anchors near-perfect (clarity 4.28/4.86); generated items 81.2%/93.8% solvable (κ=0.448)
Statistical analysis	Normalization and bootstrap	Pre-specified item bootstrap; negative DeepSeek estimate; seed-level Wilcoxon p=0.043 excluded from inference
Robustness	Structural deduplication	Structural signature; duplicates 0/0/24/0/6 vs. 0/0/3/1/2; causal interpretation remains unidentified
Evidence scope	Seed-set definition	Seeds 4–5 post-hoc; confirmatory estimate is n=3
Scoring diagnostic	True-gold re-evaluation	Placeholder scoring reduced boundary disagreement from 0.56 to 0.28
Measurement analysis	Rendering sensitivity	Corrected renderer flips 15.9% vs. 4.3%; gap 0.235 → 0.223
Robustness	Matched random-retirement null	Content-addressed rerun: 10/10 seeds above matched-null mean (p=0.00195)
Robustness	Kimi coupled sequence	CIs span 0; generator-coupled, not independent evidence
Robustness	Rand capacity diagnostics	Fixed-capacity AUC 0.430: capacity explains much of rand's gap (+0.046)
Robustness	Exemption replay	Anchors self-protect; random exemption −0.006
Mechanism	Item-level decomposition	Anchors 0.250 / added 0.230 / retired 0.157: compositional shift
Mechanism	Anchor set composition	One shared solve pattern; retention invariant, not a linking set
Mechanism	Signed metric battery	Newer-wins 98.6%/87.5%; τ=0.757; ρ=0.922/0.877
Mechanism	Signed staircases	Static: one +0.69 step to ceiling; evolved: 0.86–0.95 headroom
Mechanism	Frontier-pair targeting	76% at the 3.5p boundary; almost none the newest pair
Mechanism	Pool composition (Fig. 3)	Seed-unique pools (Jaccard 0.10–0.19) vs. 0.39–0.41
Scoring limitation	Placeholder-label fault	Ablated trajectories are not interpretable; the +0.069–+0.144 contrast does not estimate a true-answer effect
Mechanism	Component ablation, first batch (Exp. 2)	Each leave-one-out cell retains the full-system advantage (+0.005/+0.008/+0.019 vs. full; all within its seed spread, sd 0.015); verification-off with true gold interpretable; no single component identified
Design limitation	Bundled-ablation identifiability	No component attribution; 2×2×2 factorial specified as follow-up

A Appendix

A.1 Overview of Analyses

The index highlights analyses that determine or qualify reported conclusions or are required to reproduce reported values. Subsequent subsections provide the supporting diagnostics and implementation details.

B Positioning Relative to Related Work

Table 4 situates SE-Eval relative to its nearest neighbors along seven design dimensions. Each cell summarizes a design feature described or implemented in the cited work; the comparison characterizes distinct objectives rather than ranking the systems. SE-Eval's limitations in score continuity and causal identification appear in the same table. The substantive account remains §2 and Appendix B.1.

B.1 Positioning Relative to Psychometric Evaluation Frameworks

The main text positions SE-Eval against psychometric practice along four distinctions; we state them here together with the boundary cases our data illuminate.

Table 4: **Nearest-neighbor positioning of SE-Eval along seven design dimensions.**

Dimension	Benchmark refresh / task variation (White et al., 2025; Jain et al., 2025; Xia et al., 2024)	On-demand construction (LI et al., 2025; Chen et al., 2025b; Shashidhar et al., 2025)	Pipeline evolution (Wang et al., 2025; Guo et al., 2026; Zhu et al., 2026)	Psychometric / adaptive (Van der Linden et al., 2000; Li et al., 2025)	SE-Eval (this work)
Evolving object	Recent or time-indexed task collections (White et al., 2025; Jain et al., 2025); semantic variants of existing tasks (Xia et al., 2024)	Construction from topic specifications, seed code problems, or source documents (LI et al., 2025; Chen et al., 2025b; Shashidhar et al., 2025)	Benchmark grown by multiple agents (Wang et al., 2025), tasks escalated by test-time exploration (Guo et al., 2026), evaluator co-evolved with its solver (Zhu et al., 2026)	The administered sequence, not the pool: calibrated items selected adaptively (Van der Linden et al., 2000; Li et al., 2025)	Items and the controller: generated items plus admission and retirement decisions across rounds
Feedback signal	Publication time and source availability drive refresh (White et al., 2025; Jain et al., 2025); semantic transformation drives variant construction (Xia et al., 2024)	Topic specifications, code tasks, or source documents drive construction (LI et al., 2025; Chen et al., 2025b; Shashidhar et al., 2025)	Multi-agent reframing and validation drive benchmark expansion (Wang et al., 2025); proposal-driven test-time exploration and multi-level validation drive task evolution (Guo et al., 2026); generation and evaluation are optimized jointly (Zhu et al., 2026)	Latent ability estimates support information-based sequential item allocation (Van der Linden et al., 2000; Li et al., 2025)	Adjacent-version disagreement, including newer- vs. older-wins, under programmatic verification (Appendix E.2)
Validity mechanism	Recent or time-indexed items reduce exposure risk (White et al., 2025; Jain et al., 2025); leakage defense (Fang et al., 2025); semantic variants undergo iterative refinement, consistency checks, and manual correctness review (Xia et al., 2024)	Retrieval or privileged-information grounding with human quality control (LI et al., 2025); contamination-aware construction and evaluation (Chen et al., 2025b); source-document grounding (Shashidhar et al., 2025)	Instance verification and option-based double checking (Wang et al., 2025); validate-by-reproduce (Guo et al., 2026); meta-harness control (Zhu et al., 2026)	Calibrated item parameters with invariance conditions (Millsap, 2012) and equating designs (Kolen & Brennan, 2014)	Retained verifiable reference subset plus programmatic verifier; placeholder-gold labels invalidate the ablated runtime scores (Appendix B.1)
Score continuity	Performance is organized by refreshed release, time segment, or benchmark variant (White et al., 2025; Jain et al., 2025; Xia et al., 2024)	Scores are defined on each generated evaluation set (LI et al., 2025; Chen et al., 2025b; Shashidhar et al., 2025)	Performance is reported on evolved benchmark instances in Benchmark Self-Evolving and TRACE (Wang et al., 2025; Guo et al., 2026), whereas SCORE evaluates solver quality under a co-evolving evaluator (Zhu et al., 2026)	Equating and invariance designs support comparability across forms and time (Kolen & Brennan, 2014; Lord & Novick, 2008; Millsap, 2012)	Retained references provide a within-study invariant but not a linking set; equating, DIF, and drift are not modeled (Appendix B.1)
Generation / selection / retirement	Scheduled acquisition from external sources (White et al., 2025; Jain et al., 2025); LLM-derived semantic variants (Xia et al., 2024)	LLM generation with automated quality controls and, where applicable, human review (LI et al., 2025; Chen et al., 2025b; Shashidhar et al., 2025)	LLM agents generate and validate benchmark instances (Wang et al., 2025; Guo et al., 2026), while SCORE jointly trains an evaluator and solver under meta-harness control (Zhu et al., 2026)	Curated item banks; adaptive selection by information; exposure control limits reuse (Van der Linden et al., 2000)	LLM generation; admission by programmatic verifier and difficulty strata; per-round retirement of saturated items
Domain breadth	Multi-domain suite (White et al., 2025); code (Jain et al., 2025; Xia et al., 2024)	Configurable topics (LI et al., 2025); code reasoning (Chen et al., 2025b); document-grounded sets (Shashidhar et al., 2025)	Benchmark and agent tasks (Wang et al., 2025; Guo et al., 2026); deep research (Zhu et al., 2026)	Educational and psychological ability testing (Van der Linden et al., 2000); LLM evaluation (Li et al., 2025)	Controlled multi-seed evidence in one puzzle domain; one-seed transfer to executable code tasks (Appendix E.11)
Evaluation design	Temporal evaluation on refreshed tasks (White et al., 2025; Jain et al., 2025); evaluation across transformed task variants (Xia et al., 2024)	Quality analyses of generated evaluation sets (LI et al., 2025; Shashidhar et al., 2025), with contamination-focused construction and evaluation in DyCodeEval (Chen et al., 2025b)	End-to-end evaluation of evolving benchmark or evaluator pipelines (Wang et al., 2025; Guo et al., 2026; Zhu et al., 2026)	Comparability analyses based on equating and invariance (Kolen & Brennan, 2014; Millsap, 2012)	Condition comparisons and a content-addressed matched retirement replay; component effects remain unresolved (Appendices E.7 and E.11)

Why disagreement rather than Fisher information. IRT and adaptive-testing designs (Van der Linden et al., 2000; Li et al., 2025) select items to maximize information about a latent ability estimate, which requires an item-response model with calibrated parameters that we deliberately do not estimate. SE-Eval's objective is adjacent-version resolution itself—the property a static benchmark loses through saturation—so the loop optimizes observed disagreement under programmatic verification. The two objectives coincide only where items sit in the transition band between adjacent versions: our frontier-pair analysis shows generated items concentrate there, and the signed staircases show disagreement carries signal exactly at the boundaries where behavior changes. Away from those bands disagreement is inert or noise-rewarding: ceiling items contribute nothing by con-

struction, and the temperature-1.0 Kimi evaluation adds per-item sampling noise of the same order as the reported effects (Appendix E.8).

Failure under invalid labels. The placeholder-gold episode is a minimal in-domain demonstration: with unverified labels the loop produced smooth, plausible trajectories while optimizing against wrong gold, and no runtime quantity flagged the distortion. Formalizing this as a Goodhart bound—how far the closed-loop objective can drift before measurement validity fails—remains open; our data establish the failure direction (wrong labels $\Rightarrow$ mechanically suppressed disagreement) but not its magnitude in general.

Anchors are not yet a linking set. Test equating and scale linking can use common or anchor items to place scores on a shared scale (Kolen & Brennan, 2014; Lord & Novick, 2008), and measurement invariance (Millsap, 2012) gives the conditions under which item parameters stay fixed across groups and time. The implemented reference set does none of this: it is exempt from retirement and serves as the basis of the difficulty strata, but no equating transform, differential-item-functioning test, drift monitor, or linking-error bound was wired into the loop, and the bundled ablation cannot isolate its contribution. Because all three references share one solve pattern, they also cannot represent a spectrum of difficulty. Turning the references into a linking set—and quantifying how reference-set size, coverage, and difficulty distribution affect linking error—is future work.

Uncalibrated entrants. Generated items enter the pool without item parameters; the controller's per-round empirical tertiles are descriptive binning, not a latent scale, so cross-round comparability of difficulty rests on the retained references and the shared seed pool only. Active learning allocates labeling queries to informative unlabeled examples (Settles, 2009), while active testing selects test points to label for sample-efficient model evaluation (Kossen et al., 2021). SE-Eval's selection is neither: it chooses which generated items persist in the pool, judged against adjacent versions under the external reference set, and the fixed 16-per-round budget does not implement either allocation machinery.

Research directions. Taken together, the study motivates five next steps: (i) formal identifiability of the resolution objective; (ii) Goodhart bounds for closed-loop evaluation; (iii) anchor-based equating with invariance checks; (iv) a component-matched causal experiment; and (v) multi-seed replication in the executable-code domain and further verifiable domains (Appendix E.11). The study provides a controlled testbed and an analysis of implementation failure; a calibrated longitudinal measurement system remains future work.

C Execution and Analysis Deviations

The Qwen confirmatory analysis followed a plan specified before the confirmatory data were examined; the DeepSeek sequence is treated as a robustness replication. The realized execution and analysis differ from the initial plan in the following respects:

- Per-round generation was reduced symmetrically across the four conditions from 50 to 16 items because of measured gateway throughput. Completed 50-item-per-round runs are retained as robustness data.
- Reported confidence intervals use the pre-specified item-level bootstrap. Preliminary computations used seed-level resampling and a cache affected by two contamination mechanisms. The corrected computation changes point estimates only for the comparison with *rand* and does not change the associated inference.
- Realized worker caps were 0.8M tokens for static and 1.2M for adaptive conditions. Because static performs no generation or meta-evaluation, the conditions do not support a compute-efficiency comparison.
- Unavailable intermediate releases reduced the DeepSeek sequence to three versions rather than the planned minimum of four. It is therefore analyzed as a robustness replication rather than confirmatory evidence.

- The analysis plan specified three Qwen generation seeds. Seeds 4–5 were added after the first three seeds had been examined. The three-seed estimate is confirmatory, whereas the pooled n=5 estimate is a post-hoc robustness analysis despite using the same parameters.
- The structural signature used by the runtime deduplication gate (statements and clues) is the basis for the robustness analysis. Across the five seeds, duplicate-row counts are 0/0/24/0/6 for the ablated condition and 0/0/3/1/2 for SE-Eval. A rendered-answer-template basis over-counts name-set collisions and is not used.
- Two platform interruptions and one gateway timeout window interrupted workers. Runs resumed from persistent caches under unchanged parameters, and incomplete workers without summaries are excluded from analysis.

D Human Evaluation Protocol and Results

Two raters (one author and one collaborator) independently rated a blind, randomized sample of 78 pool items (20 per evolution round; round 2 had 18) plus all 32 generated items from the sampled seed using the corrected rendering shown to the evaluated models. Seed/anchor items: clarity 4.28/4.86 (rater 1/2), 100% judged uniquely solvable, 100% agreement with reference answers, 0% duplication. Generated items (seeval): programmatic unique-solution check 32/32, human-judged solvability 81.2%/93.8% (the gap is statement-rendering ambiguity, chiefly missing connectives and operator precedence), clarity 3.41/3.47, true near-duplication 4/32 (12.5%; 17 of 21 raw duplication flags were the same item persisting across rounds). Inter-rater $\kappa = 1.0$ on solvability and duplication for anchors; $\kappa = 0.448$ on generated-item solvability (item-bootstrap 95% CI $[0.0, 0.871]$), reflecting borderline-precedence disagreement. Table 5 reports the raw contingency behind the generated-item agreement: raters agree on 28/32 items (agreement 0.875, exact binomial 95% CI $[0.710, 0.965]$; prevalence 0.812 for rater 1 vs. 0.938 for rater 2). All four disagreements occur on the lowest-clarity items (both raters score clarity 2/5), and their notes attribute the disagreement to operator-precedence ambiguity (unparenthesized or/and/xor mixtures) and near-verbatim item duplication—a human-readability failure rather than a logical-validity failure. The programmatic unique-solution verifier remains the admission gate, while the human evaluation provides an offline validity check rather than an online component. A larger blind evaluation and deterministic semantic-equivalence tests for the corrected renderer remain future work (Appendix E.11).

Table 5: **Raw two-rater contingency for generated-item unique solvability (n=32).** Agreement $28/32$ (0.875), exact binomial 95% CI $[0.710, 0.965]$; prevalence 0.812 (rater 1) vs. 0.938 (rater 2); Cohen's κ=0.448. All four disagreements occur on clarity-2 items (renderer ambiguity).

Rater 1 × Rater 2	Rater 2: Y	Rater 2: N
Rater 1: Y	26	0
Rater 1: N	4	2

E Implementation Details

Generator and meta-evaluator: kimi-k3 via a single versioned OpenAI-compatible gateway; evaluated sequences and versions are listed in §4.1. kimi-k3 is disjoint from the two pre-specified sequences (Qwen, DeepSeek) but is identical to the final version of the exploratory Kimi sequence (Appendix E.8), which is therefore generator-coupled. Thinking-model truncations (finish=length with empty content) are discarded and retried; evaluation is cache-first with item-level caching keyed by content hash; budget exhaustion is an experiment-level stopping rule (one anchor-free worker in sequence A entered analysis with its two complete rounds under this rule, as specified in advance). Analysis scripts, per-worker summaries, pool snapshots, and execution records are retained with SHA-256 manifests. Every reported value regenerates deterministically from the raw layer using a fixed bootstrap seed and $10{,}000$ resamples. Pool sizes per round (sequence A): the frozen baseline pool holds 25 items; adaptive pools stay bounded at roughly 16–25 items per round (the controller adds up to 16 verified candidates and retires by stratified sampling); the AUC gap uses the four available trajectory points (baseline + three rounds, or fewer when the pre-specified budget-stop rule ends a worker early, as for the seed-1 anchor-free run). An earlier one-seed pilot at a different scale (10

generated items per round) gives the same validly scored round-1 ordering, static $<$ rand $<$ SE-Eval ($0.150/0.161/0.225$); its ablated score is omitted because it shares the placeholder-label fault.

Generated item composition. Across the five seeval runs' final-round pools (89 added items), generated items are exclusively 2–3 person knights-and-knaves puzzles (45 vs. 44 items), with logical atoms and connectives in their structured statements distributed as: telling-truth references (311), lying (132), and (88), not (88), or (64), xor (30), iff (25), implies (14). The generator does not produce multi-person ($\geq$4) puzzles at this budget, which bounds the targeted difficulty range and contributes to the domain-scope limitation in §Limitations.

E.1 Mechanism Decomposition

Item-level decomposition of the discrimination gain. Decomposing the final-round pool into kept (anchor) items, added items, and retired items reveals the controller's mechanism: the three anchors have a fixed mean per-item gap of 0.250 (identical across seeds by construction); retired items average 0.157; added items average 0.228 (sd 0.107) across n=79 content-addressed unique-within-run items. The pool-level gain therefore comes from systematically replacing low-discrimination items with higher-discrimination ones—not from any single high-gap item but from the compositional shift. Counting all added pool rows gives mean 0.230 (sd 0.101, n=89 across the five runs). This is consistent with the stratified-retirement design: the controller does not maximize individual item gap (which would collapse onto a single pattern) but instead maintains a distribution of medium-to-hard items that collectively separate adjacent versions.

E.2 Metric Battery: What Disagreement Tracks (Exploratory)

Adjacent-version disagreement scores "old wrong, new right" and "old right, new wrong" identically; the signed direction of change therefore accompanies the primary metric. On the final-round seeval pools, virtually all disagreement is progress-directional: at the main 2026-01→3.5-plus boundary (mean 19.0 differing items per seed) the newer version wins 98.6% of differing items (older-wins 1.4%); at the 3.5-plus→3.7-max boundary (mean 2.2 differing items per seed) newer-wins is 87.5%; and the two remaining boundaries carry essentially no differing items (both versions at the floor before 2026-01, both at the ceiling from 3.7-max on)—disagreement tracks the transition band and does not reward regression where it exists. On the static pool the single resolvable boundary (mean 17.2 differing items per seed) is likewise 100% newer-wins, so the direction is a property of the model sequence, not of pool evolution. Version ordering is stable under evolution (Kendall τ_b between consecutive-round accuracy vectors: mean 0.757, range 0.690–0.787; the static pool is trivially constant by construction), and monotonicity is high in both conditions (Spearman ρ of version index vs. accuracy: 0.922 [0.907, 0.935] seeval; 0.877 [0.866, 0.894] static). Anchor stability is exact by construction: the three references keep a fixed 0.250 adjacent-pair gap across all seeds and rounds. All battery statistics are post-hoc descriptive seed-level means and ranges, not inferential. Fisher information and calibrated discrimination require an item-response model, which is not estimated here (Appendix B.1). The battery inherits the DID-keyed evaluation cache: 24 item-identity content collisions occur across the five seeval workers' pool histories (s1:12, s2:3, s5:9; none in static), the same limitation addressed by the matched replay; the content-addressed recomputation is reported in Appendix E.7.

E.3 Anchor Set Properties

Anchor set composition and sensitivity. Code inspection identifies the reference set as the *first three* seed-pool items; the two other shared items are seed items retained by chance rather than anchors. All three share an identical solve pattern: the two 2025-era Qwen versions fail all three while the three 2026-era versions solve all three (unique pattern count $= 1$). This homogeneity means (i) removing any single reference item would be pattern-redundant in this run, and (ii) the set represents one difficulty boundary rather than a spectrum. The items are fixed because their correctness is program-decidable and they are exempt from retirement; the available ablation cannot identify the causal effect of that exemption because it bundles other controller changes and used invalid placeholder-label scoring. In psychometric terms the set is a retention invariant, not a linking

or equating set: no drift or differential-item-functioning monitoring, invariance-violation rule, or linking-error bound was operationalized in these runs (Appendix B.1).

E.4 Efficiency and Cost

Information density across conditions. As a complementary measure to the AUC gap, we count for each item how many of the ten model pairs it discriminates (i.e., one model solves it and the other does not). In the seed-1 run: the static pool averages 4.40 discriminated pairs per item; the evolving SE-Eval pool increases from 4.24 (baseline) to 4.80 (round 3); random replacement stays flat at 4.08→4.04. The evolving pool thus concentrates more pairwise version information per item over time, while random refresh dilutes it—consistent with the AUC-gap result but from a per-item information-theoretic angle rather than the aggregate trajectory.

Token usage across conditions. Average tokens consumed per worker (sequence A, summary-recorded): seeval 0.92M (n=5), random replacement 0.99M (n=5), anchor-free 0.89M (n=4 complete summaries; the budget-stopped seed-1 worker is excluded); static workers consume ≈0 tokens under cache-first evaluation of the frozen pool. SE-Eval uses slightly fewer tokens than random replacement in these logged workers, while the ablated condition consumes a comparable budget. Because the conditions operate under matched caps rather than total parity—and the ablated runtime scores are invalid—the token logs do not support compute-efficiency or effectiveness-per-token comparisons.

E.5 Generator Behavior

Generator behavior (success, failures, and a rendering fix). Across the full-loop attempt logs (2,121 attempts), kimi-k3 succeeds on 9.3% of attempts (random replacement 11.1%, ablated condition 14.7%, the latter admitting items without verification). The two dominant failure modes are the runtime duplicate gate (34–44% of attempts) and `no_json_object` (39–48%: the thinking model exhausts its token budget on internal reasoning); parse errors account for 3–8%, unique-solution verifier rejections 0.4%. Without anchors (941 attempts), duplicates account for 44% and format failures for 39%; the slightly higher duplicate share co-occurs with the naive add–then–drop-oldest retirement rule, and the gate catches only exact structural matches—descriptive evidence of duplicate pressure, not measurement degradation or a causal anchor effect. Attempt targets are balanced between hard and medium strata. Per accepted candidate the full loop needs 10.8 attempts versus 6.8 in the bundled ablated configuration. This is an acceptance-throughput comparison only: the configurations differ in verification and retirement, and invalid placeholder-label scoring prevents any comparison of cost per retained measurement signal. A post-hoc rendering analysis links the human–programmatic solvability gap to the statement renderer: nested compounds were emitted without parentheses, negation scope over compounds was implicit, and nested xor rendered ungrammatically—49.5% of the 222 final-pool generated statements were ambiguity-exposed. A corrected renderer (parenthesized compounds, scoped negation, list-form xor) eliminates the class by construction; re-evaluating the same 89 items under the corrected rendering (348 of 445 model–item judgments within a fixed exploratory budget) flips answers on 15.9% of ambiguity-exposed versus 4.3% of unexposed judgments, overwhelmingly correctively (wrong→right 47 vs. 2), and shifts the mean adjacent-version gap on these items from 0.235 to 0.223—i.e., the ambiguous rendering contributed roughly $+0.01$ of apparent disagreement. All numbers reported elsewhere use the original rendering.

E.6 Pool Composition and Diversity

Difficulty-coverage caveat. The bins in Figure 3 are fixed descriptive solve-rate bins used for visualization only, not the controller's round-specific empirical tertiles; the panels share one y-scale and plot change from each seed's baseline share, so cross-panel magnitude comparison is meaningful. The figure remains a descriptive diagnostic, not causal evidence that stratified retirement produced the observed shifts or that effectiveness was maintained. Generated-item composition (2–3 person puzzles with mixed connectives) is reported in Appendix E.

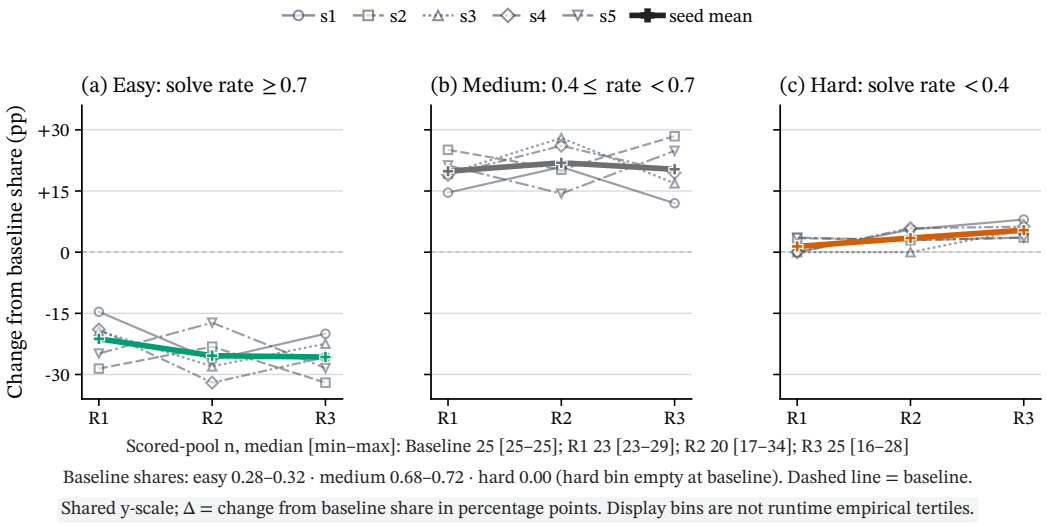


Figure 3: **Seed-resolved change in pool composition under fixed descriptive bins (Qwen SE-Eval).** Thin lines track generation seeds and thick lines show seed means. Values are changes from each seed's baseline share in percentage points, on one shared y-axis; the dashed zero line marks the baseline (easy 0.28–0.32, medium 0.68–0.72, hard 0.00—no hard items at baseline). The bins (solve rate ≥ 0.7, $0.4 \leq$ rate < 0.7, rate < 0.4) are used for visualization only and do not correspond to the runtime controller's per-round empirical tertiles. Pool-size annotations report median [min–max] scored items across seeds. Composition shifts are descriptive and do not by themselves establish that effectiveness was maintained.

Within-pool diversity degradation. Pairwise Jaccard similarity of normalized text (word sets over boilerplate-stripped puzzle and clues) among generated items within each pool: with anchors, mean similarity stays at 0.64–0.67 across rounds and seeds (21–31% of pairs exceed 0.7); without anchors, seed 3 runs persistently higher (0.69, with pairs exceeding 0.7 rising 33%→37%—the same seed that carries 24 duplicate rows), seed 5 is mildly elevated (0.67–0.68), and the remaining seeds stay at 0.64–0.65. These logs show higher lexical redundancy in selected ablated seeds, alongside a naive add–then–drop-oldest retirement rule. They are descriptive structural statistics, not evidence that anchors caused or prevented homogenization: the configurations differ in several components, and the ablated runtime discrimination scores are invalid.

Per-seed paired effects (corrected normalization). Sequence A, seeval−static per generation seed: $[+0.018, +0.054, +0.064]$ for the three pre-specified seeds (all positive; mean $+0.045$) and $[+0.018, +0.054, +0.064, +0.053, +0.043]$ pooling in the two post-hoc augmentation seeds (all positive; mean $+0.046$); seeval−rand per seed: $[+0.027, +0.034, +0.047]$ (n=3; mean $+0.036$) and $[+0.027, +0.034, +0.047, +0.032, +0.037]$ (n=5; mean $+0.036$). The exact two-sided sign test at all-same-sign gives p=0.25 (n=3) and p=0.0625 (n=5); seed is the highest exchangeable unit available.

Signed accuracy staircases (decoupling disagreement from capability). Mean final-pool accuracy per version (5 seeds): the static pool gives $[0.29, 0.31, 1.0, 1.0, 1.0]$—the version ordering is visible as a single $+0.69$ step at the 2026-01→3.5-plus boundary followed by exact ceiling—while the evolved pool gives $[0.07, 0.07, 0.86, 0.93, 0.95]$: the same ordering with continued headroom (0.86–0.95) and two additional positive increments ($+0.06$, $+0.02$) at the saturated pairs. Disagreement gains therefore do not create the ordering; they reflect the pool retaining measurable density in the transition band where behavior actually changes.

Cross-seed pool divergence. Final-round pool composition (structural signatures) across seeds: with the full loop, any two seeds share only the 3 anchor items plus 2 seed-pool items every run happened to retain (5–7 shared items; Jaccard $\approx$ 0.10–0.19 over union sizes 27–50); the remaining

generated items per pool are almost entirely seed-unique. With random replacement, any two seeds share the full 25-item baseline pool, occasionally plus one generated item (Jaccard $\approx$ 0.39–0.41) because random refresh does not use feedback to select which items to keep. This is consistent with generation-seed-dependent selection: the full-loop workers reach different regions of item space rather than one canonical pool, while the three fixed reference items provide cross-seed continuity. The association does not isolate which controller component causes the divergence.

E.7 Temporal Dynamics

Convergence speed. Round-by-round mean gaps (five seeds pooled): the full loop captures 97% of its total discrimination gain in the first evolution round ($0.176 \rightarrow 0.231$, then plateaus at ≈ 0.233); random replacement captures 81% of its smaller gain in round 1 ($0.176 \rightarrow 0.190$, plateau at ≈ 0.194). We omit the ablated runtime trajectory because placeholder-label scoring makes it uninterpretable. For the two valid trajectories, most observed change occurs in the first round and later marginal changes are small; whether that pattern generalizes beyond this sequence and budget remains open.

Frontier-pair targeting. Which adjacent version pairs do generated items actually discriminate (round-3 pools, 3 seeds)? With the full loop, 76% of generated items discriminate the 3.5p frontier boundary (3max-01→3.5-plus), 15% discriminate 3.5p→3.7m, and almost none (2%) discriminate 3.7m→3.8m. Random replacement shows a similar concentration (62% on the 3.5p boundary) but with more spread to the newest pair (10%). Both conditions concentrate on the boundary where the largest capability jump exists—the pool-level AUC gap is dominated by items that separate the pre/post-3.5-plus versions. This is consistent with the failure- driven generator's design (target the frontier pair where disagreement is highest) but also reveals a limitation: the loop struggles to generate items that separate 3.7m→3.8m (two near-ceiling models), echoing the DeepSeek saturation finding.

Matched random-retirement null (post hoc; content-addressed). The original item identifiers hash only the question/answer template, so distinct generated puzzles sharing a template can collide. Under the first-write-wins cache, later colliding puzzles can inherit the first puzzle's outcomes, and the same collision can cause the deduplication gate to skip regenerated templates. The matched replay therefore uses immutable content addresses (SHA-256 over statements and clues) and re-evaluates all affected contents—157 model–item pairs over 40 unique puzzles, 376K tokens, with the same prompt, temperature, and scoring path. For each of the ten seeds (Qwen $\times 5$, DeepSeek $\times 5$), the replay fixes the realized candidate stream, per-round accepted sets, capacity trajectory, anchor set, and retirement counts, and randomizes only which eligible non-anchor items retire (2,000 draws per seed, deterministic shuffle-based RNG). Where the realized stream regenerates a retired puzzle later, that retirement is held fixed so the fixed supply stays realizable (zero rejected paths). The analysis returns no statistics if an item identity cannot be resolved or an outcome is missing. Realized trajectories exceed the matched-null mean in 10/10 seeds (exact sign test p=0.00195): Qwen deltas $+0.028$ to $+0.043$ (one-sided tails 0.0005–0.0025), DeepSeek $+0.005$ to $+0.088$ (tails 0.014–0.393). The content-addressed AUC differs by at most 0.014 from the identifier-based estimate, and re-evaluated first contents reproduce the original cache in 72/76 pairs. This remains a post-hoc diagnostic conditional on realized supply: it does not estimate the end-to-end controller effect or isolate generation targeting.

Meta-evaluator selection quality. Comparing round-1 pool items that survive to round 3 vs. those retired (seed 1, n=23): survived items have mean per-item gap 0.212 vs. retired 0.200 ($\Delta = +0.012$), with nearly identical anchor-consistency rates (77% vs. 80%); runtime per-item difficulty scores were computed in memory and not persisted, so no difficulty-difference estimate is reported. The controller retains items with slightly higher observed discrimination—the selection signal is subtle, balancing discrimination gain against difficulty coverage rather than keeping only the highest-gap items, consistent with the stratified-retirement design. This is consistent with, but does not by itself identify, selection by the meta-evaluation signal; no per-item anchor-consistency signal was wired into admission or retirement in these runs.

E.8 Ablated-Configuration Diagnostics and Scoring Fault

What can and cannot be estimated. Random replacement − static is a valid comparison because both conditions use true labels; its final-round difference ranges from -0.007 to $+0.028$ across seeds. By contrast, SE-Eval − ablated-condition differences are not estimates of an "anchor effect." The ablated condition simultaneously removes the admission verifier, changes retirement, and scores generated items against all-knights placeholder labels. Its runtime discrimination values therefore cannot support temporal fits, per-pair effects, cross-family replication, or causal claims about anchors. The $+0.069$–$+0.144$ configuration contrast and its associated mechanistic interpretations are invalid.

Runtime-score fault and true-gold spot check. The smooth ablated trajectories are implementation diagnostics only. Generated items were scored against placeholder labels, mechanically suppressing measured disagreement as their pool share grew; raw responses were not persisted, so the trajectories cannot be repaired post hoc. A separate seed-1 true-gold re-evaluation (0.56 boundary disagreement versus 0.28 under placeholder scoring) demonstrates substantial distortion but is a single post-hoc spot check, not a reconstructed trajectory or an estimate of the fixed reference set's effect.

Pattern evolution in valid conditions. The three fixed reference items share the solve pattern (fail, fail, solve, solve, solve). Among generated items in the full-loop workers, the fraction matching that pattern rises from 63% (round 1) to 74% (round 3); random replacement changes from 50% to 56%. These are descriptive associations for validly scored conditions. The ablated condition is excluded because its correctness labels are invalid, and the full-loop trend alone does not isolate the effect of reference retention from verification and retirement.

Pool size trajectories. Mean pool sizes across five seeds: the full-loop pool stays bounded (baseline 25, round 3: mean 23, range 16–28) under stratified retirement. Random replacement grows monotonically (25→36→43→45) because it adds without removing, and the bundled ablated configuration grows fastest (25→38→46→49, with individual seeds reaching 61) under its naive retirement rule. These structural trajectories establish different evaluation costs per round, but they do not identify an anchor mechanism.

Synthesis of exploratory findings. The valid exploratory analyses support a narrower synthesis. In the implemented full configuration, observed gains concentrate in the first round, retained items have slightly higher discrimination than retired items, and final pools differ substantially across seeds while sharing the three fixed reference items. Token and pool-size logs characterize resource use. The bundled ablation adds structural observations about acceptance throughput, duplicate pressure, and pool growth, but its placeholder-label fault invalidates its runtime discrimination trajectory. Consequently, these analyses motivate hypotheses about verification, retirement, and fixed references; they do not identify an anchor effect or establish a cross-family failure mechanism.

Exemption-composition replay (post hoc, zero-cost). Under the same replay machinery (re-run with the true three-anchor set), varying *which* three items are exempt from retirement (the actual anchors vs. three random ever-pooled items vs. the three highest-gap items vs. no exemption) under a gap-greedy remix leaves the anchor, top-gap, and no-exemption variants identical (0.230) with random exemption slightly behind (0.224, -0.006)—the anchors' per-item gap (0.250) places them at the top of any quality-directed retirement ranking, so they are self-protecting and the exemption is redundant under this ranking in this domain. Exemption remains appropriate for externally verified references because it guarantees their retention independently of observed gap.

Anchor sensitivity: seed-pool size (exploratory, s7). Running the full loop from a 5-item seed pool (vs. the standard 25): the pool starts lower (baseline gap 0.150 vs. 0.176) but converges to a similar final discrimination (round-3 gap 0.250 vs. 0.233 mean). The total gain is $+0.100$ from the small pool vs. $+0.057$ from the standard pool—the tiny pool has more room to grow and the loop fills it efficiently (pool grows 5→13→14→11 as the controller retires seed items and admits generated ones). Combined with the structural finding that all three reference items share one solve pattern (Appendix E.3), this single run shows that the implementation can operate from a small seed

pool. It does not establish that three references are sufficient, diverse, or causally responsible for the observed trajectory.

DeepSeek ablated trajectories (n=3+2, invalid for inference). The DeepSeek ablated workers used the same placeholder-label scoring fault as the Qwen workers. Their runtime gaps and apparent decline shapes are therefore omitted from inference, regardless of the number of seeds or their visual consistency. They neither replicate a validity failure nor establish cross-family homogenization; only scoring-independent run metadata and structural pool statistics remain usable.

Additional random-replacement seed (exploratory). A sixth DeepSeek growing-pool random-replacement seed lands at AUC 0.389 (trajectory $0.46 \to 0.37 \to 0.38 \to 0.38$), inside the original five-seed spread $[0.325, 0.415]$; pooling all six seeds gives a random-replacement mean of 0.384 (sd 0.031), essentially unchanged. The separate fixed-capacity diagnostic (seed 7, pool locked at 25 throughout; terminal and canonicalized) yields AUC 0.430 (trajectory $0.46 \to 0.42 \to 0.44 \to 0.40$)—a capacity effect of $+0.046$ over growing rand, slightly above the growing-rand spread and comparable to the static pool (0.424). This single-seed result is consistent with a substantial capacity contribution to the observed random-replacement disadvantage: under matched capacity, undirected refresh reaches at least the full-loop mean (0.382) on this saturated sequence. Because the diagnostic is post hoc, unpaired, and based on one seed, it cannot estimate how much of the difference is caused by capacity or isolate feedback from accepted-count, retirement, and evaluation-volume changes. The content-addressed replay holds capacity fixed, so its retirement-null comparison is not confounded by capacity; what it cannot isolate is generation targeting. The valid fixed-capacity diagnostic only suggests that capacity accounts for much of the original random-replacement disadvantage on this saturated sequence.

Kimi sequence (coupled exploratory stress test). The sequence's final version kimi-k3 is the same model ID and gateway endpoint as the item generator and meta-evaluator used in every run, including this one. The sequence is therefore generator-coupled and is analyzed as a coupled stress test rather than independent third-family validation. Results with the kimi sequence (kimi-k2.5→kimi-k2.6→kimi-k3, three seeds, evaluation at temperature 1.0 due to gateway constraints), under the corrected normalization: vs. static $+0.0594$ $[-0.008, +0.128]$ and vs. rand $+0.0517$ $[-0.004, +0.107]$—positive point estimates whose CIs include zero after excluding an in-flight empty-pool worker and applying the corrected normalization. These results are directional only. The Kimi ablated workers used the same placeholder-label fault, so their runtime-score trajectories are excluded from inference and provide no cross-family replication. Their offline unique-solution rate remains a scoring-independent structural diagnostic. Temperature caveat: kimi evaluation uses temperature 1.0 (vs. 0.0 for qwen and deepseek), adding per-item sampling noise. Per-worker baseline gaps measured on the identical 25-item seed pool range from 0.16 to 0.44 across kimi workers (the three temperature-0 static workers span 0.16–0.26; temperature-1.0 workers span 0.20–0.44), so evaluation-sampling noise alone is of the same order as the reported effects—kimi numbers are directional evidence only, and the item-bootstrap CIs do not capture this noise source.

E.9 Protocol-Fixed Fresh-Seed Replication: Protocol and Results

Protocol and prospectively logged amendments were fixed before accepted replication outcomes were analyzed (full frozen text recorded with SHA-256): three fresh generation seeds per family—Qwen s8–10, DeepSeek s8–10, Kimi s4–6—each running SE-Eval and static at 16 items/round for 3 rounds; fixed sample (no data-dependent augmentation), budget exhaustion as the only early stop; family-level estimand (mean per-seed SE-Eval-minus-static trajectory-height difference) with conditional item-bootstrap intervals, exact one-sided seed sign tests, and Holm correction across the three families. For each run, the generator is disjoint from the family being evaluated. The planned generator proved structurally incapable of producing items at any output budget (its reasoning output exhausted the token budget, returning empty content), so generators were reassigned and the reassignment frozen before the remaining replication data were collected (protocol amendment): kimi-k3 for the Qwen and DeepSeek families (the same generator as their original runs, at an enlarged generation budget; generation temperature 1.0 by gateway policy), and qwen3.8-max for the Kimi family, removing its coupling. Kimi evaluations run at temperature 1.0 (gateway-forced), so the sampling-noise caveat of Appendix E.8 applies to its replication numbers as well; the replica-

tion's static reference is cache-deterministic and does not add noise. The one-shot analysis used the frozen analyzer with a fixed bootstrap seed, executed once after all 18 workers reached terminal state. Table 6 reports the results; replication estimates are not pooled with the original seeds.

Table 6: **Protocol-fixed fresh-seed replication (three seeds per family; generators disjoint from the evaluated family).** Family-level mean difference in trajectory height (SE-Eval vs. static) over the three new seeds, conditional item-bootstrap 95% intervals, and exact one-sided seed sign tests; Holm-adjusted across families (α=0.05, one-sided). Kimi evaluations are temperature-1.0 sampling-limited (Appendix E.8).

Family (3 new seeds)	Generator	Mean Δ	Item-bootstrap 95% CI	Sign (positive)	Verdict
Qwen (s8–10)	kimi-k3	$+0.0396$	$[+0.0193, +0.0610]$	3/3 (p=0.125)	sub-MDE
DeepSeek (s8–10)	kimi-k3	-0.0893	$[-0.1671, -0.0136]$	0/3 (p=0.125)	not supported
Kimi (s4–6; disjoint generator)	qwen3.8-max	$+0.1673$	$[+0.1139, +0.2198]$	3/3 (p=0.125)	$\geq$ MDE

E.10 Saturation vs. Headroom Proxy (Table 7)

Table 7: **Saturation vs. headroom proxy (sequence A, seeval, 5 seeds; exploratory).** Accuracy on the unselected frozen seed pool (25 items) vs. unique generated items across rounds; Δ is frozen $-$ fresh. 2026-era versions saturate the frozen pool (100%) while retaining headroom on the fresh stream. The static ceiling evidences saturation only—item difficulty strongly confounds the frozen-vs-fresh difference—so this proxy bounds, rather than establishes, contamination resistance.

Evaluated version	frozen seed pool	fresh generated	Δ (frozen $-$ fresh)
3-max (2025-09)	0.312	0.253	+0.059
3-max (2026-01)	0.304	0.268	+0.036
3.5-plus	1.000	0.830	+0.170
3.7-max	1.000	0.910	+0.090
3.8-max	1.000	0.933	+0.067

E.11 Future Experiments

Three designs follow directly from the limitations above; all reuse the existing controller and statistical design where possible.

Component-matched ablation. The bundled ablation cannot attribute the full loop's behavior to any single component. A planned leave-one-out decomposition includes three matched conditions that each remove exactly one component—*anchors-off* (no reference set; admission verification and stratified retirement kept), *verification-off* (naive admission; anchors and retirement kept), and *retirement-off* (random retirement at matched capacity; anchors and verification kept)—alongside the existing full system and bundled-ablation cells as the on/on/on and off/off/off endpoints of the decomposition. Every condition keeps 16 items per round, three rounds, the same generator and evaluated sequence, and matched capacity, candidate stream, and evaluation volume; runtime scoring uses offline-solver true gold in every cell, avoiding the placeholder-label fault diagnosed in Section 4.5. The primary outcome is the corrected adjacent-version AUC gap with the same item-level bootstrap and sign tests as the main analysis; each component's contribution is estimated as the full-system advantage minus its leave-one-out cell. The analysis and stopping rules are specified before launch, with three pre-specified seeds per condition and per-cell stopping at budget exhaustion or a pre-specified interim gate. Because the hypothesis was formulated post hoc, the resulting estimates remain exploratory and are excluded from the confirmatory analysis.

Component ablation, first batch (executed post hoc; exploratory). A first batch—one seed per cell, Qwen s6, 16 items per round for three rounds, offline-solver true gold in every cell—has been executed. Each leave-one-out cell retains the full system's advantage over static: mean trajectory heights of 0.229 (anchors-off), 0.232 (random retirement), and 0.243 (verification-off) versus 0.224 for the full system (five-seed mean) and 0.178 for static—each cell inside the full system's seed-level spread (0.208–0.244) and every difference below $1.4\times$ the cross-seed standard deviation (0.015).

No single component is therefore identified as the driver at this sample size. The verification-off cell is the most informative: all 21 naive-admitted items had a unique solver solution and were scored against offline true gold, so its trajectory is interpretable and shows no degradation—direct evidence that the earlier collapse was the placeholder-label scoring fault rather than the missing gate. A content-addressed recomputation shifts the random-retirement cell by -0.008 (its entire apparent delta), leaving all conclusions unchanged. The remaining two seeds per cell follow the pre-specified plan, gated on budget approval.

Second structural domain (executed post hoc; exploratory). We transfer the controller to Python function synthesis with unit-test execution as the verifier, using one Qwen seed and the same three-round schedule. SE-Eval increases trajectory height over static by $+0.0503$ $[+0.0005, +0.0998]$ under an item bootstrap; the mean gap rises from 0.070 to 0.130, while static remains at 0.040. This demonstrates implementation transfer to a structurally different executable verifier, but one seed cannot establish cross-domain generality. Broader assessment still requires pre-specified multi-seed studies in code, SQL, proof checking, or constraint satisfaction.

Expanded human evaluation. The two-rater evaluation is small and single-domain. A follow-up can enlarge the blind sample and rater pool, add deterministic semantic-equivalence tests for the corrected renderer, and keep the human-readability check separate from the programmatic admission verifier (Appendix D).

Longer-horizon roadmap. Beyond these experiments, the remaining work organizes into four layers. *Theory*: formal estimands for longitudinal benchmark evolution—resolution, validity, drift, and linking error—together with the conditions under which maximizing disagreement aligns with ability separation rather than inducing Goodhart failure; the relationship between reference-set size and coverage and drift-detection power; and identifiability conditions or error bounds for anchor-based equating (Appendix B.1). *Measurement*: dynamic or hierarchical IRT with Fisher-information and posterior-uncertainty item selection, item-drift and differential-item-functioning monitoring, and signed ability-change tracking, of which the newer-wins decomposition provides a model-free descriptive analysis. *Causal*: the matched experiments above, executed with pre-specified stopping and an independent holdout. *Breadth*: at least two structural domains, at least three disjoint evaluatee families with adequate per-family seeds, and a baseline battery spanning static, matched random, IRT/CAT, uncertainty/information-gain selection, and one-shot automatic generation.

F Full Limitations and Validity Threats

Construct validity. The study operationalizes benchmark quality as adjacent-version disagreement (mean trajectory height, the corrected AUC gap)—a resolution measure, not signed capability growth. Other desiderata—coverage, calibration of absolute difficulty, fairness—are outside the estimand. The anchor-free ablation was pre-specified under a structural operationalization (offline unique-solution rate) that the data did not support. A post-hoc hypothesis that fixed references might help protect measurement validity cannot be tested by this run: the ablated configuration also removes the admission verifier, changes retirement, and uses invalid placeholder-label scoring. Testing that hypothesis would require a prospectively specified, component-matched replication on an independent holdout.

Internal validity. For the pre-specified sequences, the generator and meta-evaluator come from a different model family than the evaluated models (the exploratory Kimi sequence's final version shares the generator endpoint; see Appendix E.8), and all adaptive conditions share a 1.2M-token cap, while static uses a 0.8M cap because it performs no generation or meta-evaluation. The unequal caps preclude compute-efficiency comparisons. Rounds, deduplication, and caching are otherwise controlled, but residual confounds remain possible through gateway-side model updates during the study window. The bootstrap basis and deduplication basis required correction, as detailed in Appendix C. The runtime deduplication gate uses a structural signature that treats logically equivalent paraphrases as distinct.

External validity. The controlled multi-seed evidence is from one programmatically verifiable domain (knights-and-knaves logic puzzles), two pre-specified evaluated families, and a fresh-seed

Kimi replication. A one-seed executable-code transfer is directionally consistent (Appendix E.11) but does not establish cross-domain generality. Generalization to non-verifiable domains—where no executable ground truth exists—remains unresolved and is precisely where external grounding is hardest to transplant. Contamination resistance is evaluated only through a proxy (frozen-pool saturation vs. fresh-item headroom); the budget–information trade-off is not evaluated, since condition-specific budget caps preclude efficiency claims.

Statistical validity. The primary Qwen analysis rests on three pre-specified generation seeds; the two seeds added after the primary results were examined contribute only to the pooled robustness analysis. All three pre-specified paired differences vs. static are positive, but the exact two-sided sign test at $n{=}3$ gives $p{=}0.25$; pooling in the two augmentation seeds (five of five positive) gives $p{=}0.0625$; a normal approximation is anti-conservative at this sample size. The item-bootstrap CIs condition on the realized seeds and do not represent seed- or family-level uncertainty. In the fresh-seed replication, Holm-adjusted conditional item-bootstrap tests reject the positive null for Qwen and Kimi, whereas the exact seed-level tests remain underpowered at $n{=}3$ ($p{=}0.125$ for each all-same-sign family). The DeepSeek robustness analysis (five seeds; three pre-specified plus two post-hoc augmentation) does not replicate the primary effect under the corrected normalization (-0.04, CI spanning zero; four of five seeds negative), and its vs.-rand comparison is null. Qwen's corrected advantage over static has a CI excluding zero but a point estimate below the minimum detectable effect; mechanism attribution is additionally limited by the bundled ablation and its scoring fault. The human evaluation uses two raters on a single-seed sample. Agreement is perfect only for anchor solvability and duplication ($\kappa{=}1.0$); agreement on generated-item solvability is moderate ($\kappa{=}0.448$), consistent with unresolved rendering ambiguity. The sample is small, and these limitations remain unresolved.