

Keating: A Metaharness for Agency-Preserving AI Instruction

The Keating Research Group
April 03, 2026

AI tutors can scale explanation, but scaling explanation is not the same as scaling learning. A tutoring system that answers fluently may still weaken the learner’s own reconstruction of a concept. Keating is designed around that distinction. It is not a single tutoring chatbot; it is a **metaharness** for teaching, a control layer that organizes planning, prompting, retrieval, transfer, verification, and evaluation around the live teaching exchange. We analyze two evidence layers: an archival trace set of 22 raw sessions curated to 16 topic x learner pairs, and a synthetic benchmark implemented directly in the repository. The archival set yields a normalized overall score of 0.61 (95% bootstrap interval 0.515-0.705), with strong topic heterogeneity: **Special Relativity** is highest at 0.75 and **Stoicism** lowest at 0.425. The synthetic layer shows that the current Keating policy, although evolved on **Derivative** alone, improves the full 14-topic harness by 6.703 points over the default policy across 200/200 seeds, with derivative-only evolution improving in 29/30 reruns. The contribution of this paper is therefore twofold: a formal account of a teaching metaharness and a reproducible benchmark-and-analysis stack for studying agency-preserving instruction. The present evidence supports systems and methodology claims; a human randomized trial remains the necessary next step for causal pedagogical claims.

Introduction

One-to-one tutoring remains one of the strongest known educational interventions (Bloom, 1984), and intelligent tutoring systems improve learning outcomes on average across domains (Ma et al., 2014). Recent generative-AI tutors suggest that some of that benefit may be scalable (Kestin et al., 2025); (Thesen & Park, 2025). But tutoring quality cannot be judged only by whether the learner receives an explanation. Learning requires active reconstruction, self-explanation, and transfer, whereas AI systems make it unusually easy for the learner to accept a polished account instead of generating one (Chi et al., 1989); (Rittle-Johnson & Loehr, 2017). This is exactly the terrain where cognitive offloading becomes a serious educational risk (Risko & Gilbert, 2016); (Burnett & Richmond, 2026).

Keating is built for that failure mode. The central idea is simple: if conventional AI tutors are mostly **chatbots that teach**, Keating is a **metaharness that governs teaching**. A chatbot is primarily an interaction policy at the level of the next response. Keating adds a layer above that response policy. It generates and constrains plans, maps, visual artifacts, verification checks, prompt evolution, benchmark traces, and policy search. The live teaching exchange is therefore only one component of a larger pedagogical control system.

That distinction matters both scientifically and technically. For readers from learning science, the key claim is that the system is organized around mechanisms such as diagnosis, retrieval, and self-explanation rather than answer delivery. For readers from machine learning and systems, the key claim is that the object being optimized is not merely a prompt or a chatbot transcript, but an **instructional environment** with inspectable artifacts, objective functions, mutation operators, and safety gates. Recent systems work has started to optimize harnesses and editable meta-level agents in non-educational domains (Lee et al., 2026); (Zhang et al., 2026). Keating is therefore “two steps

ahead” of chatbot tutoring in a precise sense: it does not merely generate tutoring turns; it evaluates and redesigns the conditions under which those turns are generated for teaching.

We make three contributions:

1. We formalize Keating as a teaching metaharness rather than a tutoring chatbot.
2. We provide a mathematical account of the synthetic harness used for policy search.
3. We present a reproducible analysis stack that separates archival trace evaluation from internal synthetic optimization.

Keating as a Teaching Metaharness

Why a metaharness is different from a chatbot

A tutoring chatbot takes a learner message and emits a response. Its quality is mostly determined by the prompt, the model, and short conversational context. By contrast, a metaharness decides what **kind** of instructional act should happen next, what artifacts should exist before that act, what evidence counts as success, and how future policies should be revised when current ones fail.

Keating is metaharnessed along four axes:

1. **Artifact layer.** The system can generate lesson plans, concept maps, animations, verification checklists, prompt-evolution reports, benchmark reports, and policy traces.
2. **Governance layer.** Teaching behavior is parameterized by a policy with explicit controls such as formalism, retrieval practice, and challenge rate.
3. **Evaluation layer.** Policies are benchmarked against a synthetic learner suite rather than revised purely by intuition.
4. **Improvement layer.** Mutation, selection, and archival comparison operate on the teaching harness itself.

In practical terms, Keating does not ask only “what should the tutor say next?” It also asks:

- What diagnostic state should exist before explanation begins?
- What scaffold should force reconstruction rather than agreement?
- What evidence would show that a policy is helping one learner type while harming another?
- Which parameters of the teaching environment should be changed after a failed run?

Those are metaharness questions, not chatbot questions.

The nearest systems analogues are recent agentic frameworks that optimize harness code or editable self-improvement procedures outside education (Lee et al., 2026); (Zhang et al., 2026). Keating differs in the object being optimized. The target is not a coding harness or a general self-improving agent, but a pedagogical environment whose artifacts, objectives, and mutation rules are defined around explanation quality, retrieval, reconstruction, and transfer.

Natural entry points for different readers

Readers from education can think of Keating as an attempt to operationalize a mastery loop: diagnose, teach, probe, repair, retrieve, and transfer. Readers from ML can think of it as a structured controller over an LLM-based instructional policy. Readers from systems can think of it as a benchmarked orchestration layer that moves some of the intelligence from the model’s hidden behavior into inspectable artifacts and explicit objective functions.

The rest of the paper keeps these perspectives aligned. The results section focuses on what the harness learns from data. The methods section formalizes the benchmark mathematically and operationally.

Results

The evidence stack

The paper uses two evidence layers.

1. **Archival external evaluation.** We analyze teaching traces stored in the repository under `test/traces/`.
2. **Internal synthetic benchmark.** We evaluate policies with the harness implemented in `src/core/benchmark.ts`.

The two layers answer different questions. The archival layer asks what kinds of concrete successes and failures appear in recorded sessions. The synthetic layer asks whether the policy-search machinery is robust inside the benchmark it is designed to optimize.

The repository contains 22 raw trace files, including repeated runs for some topic x learner pairs. We imposed a deterministic curation rule: retain the latest trace by timestamp for each topic x learner pair. This yielded 16 retained sessions, matching the checked-in snapshot `test/final_dataset.json`. One retained derivative trace for `Qwen-2.5-1.5B` contained `mastery=8`, `engagement=7`, and `clarity=8` while the rest of the archive used the 0-1 scale; we normalized that record to 0.8, 0.7, and 0.8 and recorded the correction in the generated analysis bundle.

Component	Value	Interpretation
Raw archived traces	22	All preserved teaching transcripts before curation
Retained topic x learner pairs	16	Archival evaluation set used in this paper
Excluded duplicate earlier runs	6	Older runs for the same topic x learner pair
Score corrections	1 record	Single 10x encoding error normalized before aggregation
Synthetic topics	14	Internal benchmark tasks implemented in code
Synthetic learners per topic	18	Pseudo-learners sampled per topic and seed

Table 1: Evidence layers and curation rules.

Archival performance is heterogeneous across topics and learners

After normalization, the mean archival overall score, defined as the unweighted mean of mastery, engagement, and clarity, was 0.61 with a 95% bootstrap interval of 0.515-0.705. Performance varied sharply by topic. **Special Relativity** was strongest at 0.75 (0.596-0.883), followed by **Derivative** at 0.654 (0.454-0.767), **Social Contract Theory** at 0.613 (0.500-0.762), and **Stoicism** at 0.425 (0.283-0.558).

This pattern is substantively useful. The physics and calculus topics are structurally friendly to prediction, worked examples, and misconception repair, whereas Stoicism demands introspective application. Keating remains clear on Stoicism, but clarity does not translate into learner uptake as reliably as it does in the more formal domains.

Topic	n	Overall	Mastery	Interpretation
Special Relativity	4	0.750 (0.596-0.883)	0.695 (0.570-0.815)	Strong transfer from intuitive thought experiment to formal structure
Derivative	4	0.654 (0.454-0.767)	0.600 (0.387-0.775)	Conceptual calculus teaching is strong but not uniformly clean
Social Contract Theory	4	0.613 (0.500-0.762)	0.537 (0.463-0.650)	Mixed engagement and mixed transfer
Stoicism	4	0.425 (0.283-0.558)	0.287 (0.175-0.463)	Explanation often exceeds genuine learner uptake

Table 2: Archival evaluation by topic. Values are means with 95% bootstrap intervals.

Learner-model heterogeneity was also substantial. **Qwen-2.5-1.5B** scored highest overall at 0.779 (0.675-0.867), whereas **Llama-3.2-1B** scored lowest at 0.458 (0.367-0.550). We do not interpret these as broad claims about model families. The dataset is too small for that. The relevant result is narrower: the metaharness is not uniformly robust across simulated learner profiles, even in a small archive.

Student-role contamination is a central failure mode

The traces reveal a particularly important failure mode for agency-preserving instruction: student-role contamination. In some sessions, student turns begin to speak like a teacher or assistant, for example opening with formulaic tutor language instead of reconstructing the concept as a learner. Using a simple heuristic over student turns, 5 of the 16 curated sessions showed at least one contamination marker.

These sessions performed worse. Sessions without contamination had mean mastery 0.575 and mean overall score 0.642, compared with 0.430 and 0.540 for contaminated sessions. Because the sample is small, we treat this contrast as descriptive. Nonetheless, it is exactly the kind of failure a metaharness should detect: not merely whether the content is correct, but whether the learner is sounding fluent without evidencing ownership of the idea.

Synthetic policy gains are robust and generalize beyond the tuned topic

Keating’s current policy (**keating-candidate-22**) was evolved on **Derivative**, not on the entire suite. That makes the full-suite benchmark a useful internal check against narrow overfitting. Across 200 seeds, the current policy improved the default policy by 6.703 points overall (2.5th-97.5th percentiles: 6.341-7.093), winning on 200/200 seeds. The mean delta on **non-derivative** topics was 6.704 (6.276-7.112), essentially identical to the derivative-specific gain of 6.704 (5.172-8.232).

We also reran derivative-only evolution 30 times from the default policy. The best evolved policy beat the baseline in 29 of 30 runs, with mean gain 5.768 points. Within the harness, policy search is therefore stable enough to be useful rather than purely anecdotal.

Synthetic analysis	Result	n	Interpretation
Default vs. current full-suite benchmark	+6.703 points (6.341-7.093)	200 seeds	Current policy beats default on every sampled seed
Derivative-only topic delta	+6.704 points (5.172-8.232)	200 seeds	Large tuned-topic gain
Non-derivative mean delta	+6.704 points (6.276-7.112)	200 seeds	Comparable gain on untuned topics
Derivative evolution reruns	29/30 wins; mean +5.768	30 runs	Mutation-and-gate procedure is usually improving

Table 3: Synthetic robustness checks for the current policy.

Synthetic ablations show what the metaharness currently rewards

The current policy differs from the default policy by nine scalar parameters. One-at-a-time ablations show that the largest synthetic gains come from maximal retrieval practice (+3.137 points when swapped into the default policy), lower challenge rate (+2.324), and higher interdisciplinary bias (+0.877). Increasing diagram bias or reflection bias alone reduces mean benchmark score by about 0.4 points each.

This is a useful diagnosis of the metaharness itself. The benchmark is highly sensitive to retrieval and overload control, but less sensitive to reflective richness in isolation. That does not mean reflection is unimportant pedagogically. It means the current harness is better at rewarding some desirable instructional traits than others.

Parameter swapped into default	Mean synthetic delta	Interpretation
retrievalPractice	+3.137	Strongest driver; the harness strongly rewards enforced recall
challengeRate	+2.324	Reducing overload is the second largest contributor
interdisciplinaryBias	+0.877	Transfer-oriented prompting helps modestly
analogyDensity	+0.445	Analogical pacing helps, but less than retrieval or challenge control
diagramBias	−0.406	Visual emphasis alone is not sufficient in the current harness
reflectionBias	−0.408	Reflection prompts alone are not reliably rewarded

Table 4: One-at-a-time ablations reveal what the synthetic benchmark currently values.

Discussion

The evidence supports three main claims.

First, Keating is best understood as a **teaching metaharness**. Its novelty is not simply that it chats with learners, but that it scaffolds, audits, and evolves the instructional process around the chat exchange.

Second, Keating yields meaningful failure analysis. The archival traces show that high clarity can coexist with weak mastery, that introspective topics are harder than formal technical topics, and that student-role contamination is a visible and educationally important failure mode.

Third, the current policy is robust inside its synthetic optimization environment. Topic-held-out gains, seed robustness, and rerun stability all point in the same direction.

The paper does **not** claim that Keating has already been proven superior on human learners. The archival trace set is small, not blinded, and not paired with delayed post-tests or inter-rater reliability. The synthetic harness is implemented and reproducible, but it remains a model of pedagogy rather than pedagogy itself. The correct reading of the present paper is therefore that Keating is a publishable systems-and-methods contribution with an explicit path toward human evaluation.

The next decisive study is a preregistered randomized comparison between Keating and at least one strong AI tutor baseline, with real learners, blinded rubric scoring, delayed retention tests, and explicit transfer tasks. If the metaharness framing continues to outperform chatbot-style tutoring under those conditions, the claim of educational significance becomes much stronger.

Methods

System overview

Keating is implemented as a policy-controlled teaching scaffold around a Pi runtime. The live system can generate lesson plans, maps, verification artifacts, animations, benchmark reports, and prompt-evolution artifacts, but the present paper focuses on the teaching-policy layer and the synthetic harness. A teaching policy contains nine scalar controls:

- analogy density
- Socratic ratio
- formalism
- retrieval practice
- exercise count
- diagram bias
- reflection bias
- interdisciplinary bias
- challenge rate

These controls do not directly encode a single answer. They encode a region of instructional behavior. The metaharness evaluates those controls against topic structure and learner profiles, then uses the resulting signals to revise the policy.

Mathematical formulation of the harness

For readers from educational measurement, this section defines the latent teaching signals. For readers from ML systems, it specifies the benchmark objective. For readers from applied mathematics, it gives the explicit map from policy and learner parameters to session score.

Let a topic be represented by T , a policy by P , and a learner profile by L . The policy vector is

$$P = (a, s, f, r, e, d, b, i, c)$$

where a is analogy density, s Socratic ratio, f formalism, r retrieval practice, e exercise count, d diagram bias, b reflection bias, i interdisciplinary bias, and c challenge rate.

The learner vector is

$$L = (k, u, n, q, v, p, t, x)$$

where k is prior knowledge, u abstraction comfort, n analogy need, q dialogue preference, v diagram affinity, p persistence, t transfer desire, and x anxiety.

For a topic with formalism level φ_T and visualizability indicator ν_T , Keating computes the following fit terms. All intermediate quantities are clipped to the interval $[0, 1]$ after evaluation.

$$F_i = 1 - |a - n|$$

$$F_r = 1 - |f - \frac{\varphi_T + u}{2}|$$

$$F_d = 1 - |s - q|$$

$$F_g = 1 - |d - (\nu_T v + (1 - \nu_T)\omega_\nu)|$$

$$F_p = 1 - |\frac{e}{e_{\max}} - (1 - k + \omega_x x)|$$

$$F_b = 1 - |b - t|$$

Here ω_ν is the diagram fallback used for weakly visual topics, ω_x is the anxiety-to-practice coupling, and e_{\max} is the exercise-count normalization constant induced by the policy domain. The model also computes an overload term parameterized by the bundle **Theta_0**:

$$O = \lambda_0 + \lambda_f f + \lambda_e \frac{e}{e_{\max}} + \lambda_c c - \lambda_p p + \lambda_x x - \lambda_k k$$

These intermediate quantities are then transformed into the five synthetic learning outcomes by the parameter bundles **Theta_M**, **Theta_R**, **Theta_E**, **Theta_T**, and **Theta_C**:

$$M = \mu_M + \alpha_i F_i + \alpha_r F_r + \alpha_d F_d + \alpha_g F_g + \alpha_p F_p + \alpha_o (1 - O)$$

$$R = M(\rho_0 + \rho_r r)$$

$$E = \mu_E + \beta_i F_i + \beta_d F_d + \beta_g F_g + \beta_b F_b + \beta_o (1 - O)$$

$$T_r = R(\tau_0 + \tau_i i + \tau_t t)$$

$$C = \mu_C + \gamma_o O + \gamma_f |f - u| + \gamma_c |c - p|$$

where M is mastery gain, R retention, E engagement, T_r transfer, and C confusion.

Finally, the session score is a weighted composition with bundle **Theta_S**:

$$S = \sigma_M M + \sigma_R R + \sigma_E E + \sigma_T T_r - \sigma_C C$$

Topic-level benchmark scores are the mean of S over the learner population for that topic, multiplied by a reporting scale parameter. Suite-level benchmark score is the mean over topics.

The harness is therefore defined by its structure plus its calibration:

$$\Theta = \{\omega_\nu, \omega_x, e_{\max}, \Theta_O, \Theta_M, \Theta_R, \Theta_E, \Theta_T, \Theta_C, \Theta_S\}$$

where each `Theta_*` denotes a small family of scalar parameters. The present repository instantiates `Theta` with one concrete numeric setting in `src/core/benchmark.ts`, but the paper intentionally presents the more general parameterized metaharness. This formulation is deliberately interpretable. It is not intended as a psychologically complete model of learning. Its purpose is to make the metaharness legible enough that policy evolution is inspectable rather than opaque.

Harness pseudocode

```
BUILD-LEARNER-POPULATION(seed, count)
1 initialize PRNG with seed
2 learners <- empty list
3 for i <- 0 to count - 1
4   learner.id <- "learner-" + seed + "-" + i
5   learner.priorKnowledge <- RANDOM()
6   learner.abstractionComfort <- RANDOM()
7   learner.analogyNeed <- RANDOM()
8   learner.dialoguePreference <- RANDOM()
9   learner.diagramAffinity <- RANDOM()
10  learner.persistence <- RANDOM()
11  learner.transferDesire <- RANDOM()
12  learner.anxiety <- RANDOM()
13  append learner to learners
14 return learners
```

This procedure samples the synthetic learner population for one topic. In CLRS terms, it is the input-construction phase for the benchmark: before Keating can evaluate a teaching policy, it needs a distribution of learners with varying prior knowledge, abstraction comfort, persistence, and anxiety. The important point for the present paper is that the population size is an argument rather than a fixed constant of the algorithm. The repository currently supplies one concrete learner count, but the metaharness does not depend on that exact choice.

```
SIMULATE-TEACHING(policy, topic, learner, theta)
1 intuitionFit <- 1 - |policy.analogyDensity - learner.analogyNeed|
2 rigorTarget <- CLIP((topic.formalism + learner.abstractionComfort) / 2)
3 rigorFit <- 1 - |policy.formalism - rigorTarget|
4 dialogueFit <- 1 - |policy.socraticRatio - learner.dialoguePreference|
5 if topic.visualizable
6   diagramTarget <- learner.diagramAffinity
7 else diagramTarget <- theta.visualFallback
8 diagramFit <- 1 - |policy.diagramBias - diagramTarget|
9 practiceNeed <- CLIP(1 - learner.priorKnowledge
10   + theta.practiceAnxietyWeight * learner.anxiety)
11 practiceFit <- 1 - |policy.exerciseCount / theta.exerciseNormalization
12   - practiceNeed|
13 reflectionFit <- 1 - |policy.reflectionBias - learner.transferDesire|
14 overload <- CLIP(theta.overloadBias
15   + theta.overloadFormalism * policy.formalism
16   + theta.overloadExercises
17     * policy.exerciseCount / theta.exerciseNormalization
18   + theta.overloadChallenge * policy.challengeRate
19   - theta.overloadPersistence * learner.persistence
20   + theta.overloadAnxiety * learner.anxiety
21   - theta.overloadKnowledge * learner.priorKnowledge)
22 masteryGain <- CLIP(theta.masteryBias
23   + theta.masteryIntuition * intuitionFit
24   + theta.masteryRigor * rigorFit
25   + theta.masteryDialogue * dialogueFit
26   + theta.masteryDiagram * diagramFit)
```



```

27         + theta.masteryPractice * practiceFit
28         + theta.masteryHeadroom * (1 - overload))
29 retention <- CLIP(masteryGain
30                 * (theta.retentionBase
31                   + theta.retentionRetrieval
32                     * policy.retrievalPractice))
33 engagement <- CLIP(theta.engagementBias
34                   + theta.engagementIntuition * intuitionFit
35                   + theta.engagementDialogue * dialogueFit
36                   + theta.engagementDiagram * diagramFit
37                   + theta.engagementReflection * reflectionFit
38                   + theta.engagementHeadroom * (1 - overload))
39 transfer <- CLIP(retention
40                 * (theta.transferBase
41                   + theta.transferInterdisciplinary
42                     * policy.interdisciplinaryBias
43                   + theta.transferDesire
44                     * learner.transferDesire))
45 confusion <- CLIP(theta.confusionBias
46                  + theta.confusionOverload * overload
47                  + theta.confusionFormalismGap
48                    * |policy.formalism - learner.abstractionComfort|
49                  + theta.confusionChallengeGap
50                    * |policy.challengeRate - learner.persistence|)
51 score <- CLIP(theta.scoreMastery * masteryGain
52              + theta.scoreRetention * retention
53              + theta.scoreEngagement * engagement
54              + theta.scoreTransfer * transfer
55              - theta.scoreConfusion * confusion)
56 return (masteryGain, retention, engagement, transfer, confusion, score)

```

This is the core scoring routine. It converts one policy-topic-learner triple into interpretable intermediate quantities and then into a final score. The important structural fact is that Keating does not score a policy directly. It first scores alignments: analogy pacing, rigor matching, dialogue matching, visual fit, practice load, reflection match, and overload. These are then composed into the five outcome variables used by the benchmark. In that sense, the harness is factorized: it makes the path from policy parameters to session score inspectable. Passing the parameter bundle `theta` explicitly makes the generality of the metaharness visible. A reviewer can change the calibration without changing the algorithmic structure.

```

SUMMARIZE-TOPIC(topic, simulations, traceLimit, reportScale)
1 ranked <- simulations sorted in decreasing order by score
2 summary.meanScore <- reportScale
3   * MEAN(score for each simulation in simulations)
4 summary.meanMasteryGain <- MEAN(masteryGain for each simulation in simulations)
5 summary.meanRetention <- MEAN(retention for each simulation in simulations)
6 summary.meanEngagement <- MEAN(engagement for each simulation in simulations)
7 summary.meanTransfer <- MEAN(transfer for each simulation in simulations)
8 summary.meanConfusion <- MEAN(confusion for each simulation in simulations)
9 summary.topLearners <- first traceLimit entries of ranked
10 summary.strugglingLearners <- last traceLimit entries of ranked, reversed
11 summary.dominantStrength <- strongest average alignment signal
12 summary.dominantWeakness <- weakest average alignment signal
13 return summary

```

This procedure aggregates a set of learner-level simulations into a topic-level result. The benchmark therefore preserves both population averages and diagnostic tails. The mean score tells us how a policy performs on average; the strongest and weakest learners tell us **why**. That diagnostic tail is one place

where the metaharness differs from chatbot evaluation, which often reports only a single aggregate success rate. The reporting scale is likewise explicit rather than hardwired into the procedure.

```
RUN-BENCHMARK-SUITE(policy, topics, seed, config)
1 topicBenchmarks <- empty list
2 topicTraces <- empty list
3 for j <- 0 to length(topics) - 1
4   topic <- topics[j]
5   topicSeed <- seed + config.topicSeedStride * j
6   learners <- BUILD-LEARNER-POPULATION(topicSeed, config.learnerCount)
7   simulations <- empty list
8   for each learner in learners
9     simulation <- SIMULATE-TEACHING(policy, topic, learner, config.theta)
10    append simulation to simulations
11    summary <- SUMMARIZE-TOPIC(topic,
12                               simulations,
13                               config.traceLimit,
14                               config.reportScale)
15    append summary to topicBenchmarks
16    append summary.trace to topicTraces
17 weakestTopic <- topic with minimum meanScore in topicBenchmarks
18 overallScore <- MEAN(meanScore for each topic summary in topicBenchmarks)
19 return (overallScore, weakestTopic, topicBenchmarks, topicTraces)
```

This is the full benchmark harness. It loops over topics, builds a learner population for each one, simulates the teaching interaction at the score-model level, and aggregates the results into a suite score. In algorithmic terms, it is a nested evaluation loop: topics on the outside, learners on the inside, summary at the end. The `config` argument makes clear that learner count, trace depth, reporting scale, seed stride, and harness coefficients are benchmark choices rather than theoretical necessities. This is the procedure that produces the benchmark reports cited in the results section.

```
EVOLVE-POLICY(basePolicy, focusTopic, iterations, seed, config)
1 baseline <- RUN-BENCHMARK-SUITE(basePolicy, [focusTopic], seed, config)
2 best <- baseline
3 acceptedCandidates <- empty list
4 exploredCandidates <- empty list
5 seenPolicies <- {basePolicy}
6 initialize PRNG with seed + config.evolutionSeedOffset
7 for iteration <- 1 to iterations
8   candidatePolicy <- MUTATE(best.policy, PRNG, iteration)
9   novelty <- NOVELTY-SCORE(seenPolicies, candidatePolicy)
10  candidateBenchmark <- RUN-BENCHMARK-SUITE(candidatePolicy,
11                                             [focusTopic],
12                                             seed
13                                             + config.evolutionSeedStride
14                                             * iteration,
15                                             config)
16  improves <- candidateBenchmark.overallScore > best.overallScore
17  safe <- candidateBenchmark.weakestTopicScore
18         >= best.weakestTopicScore - config.weakestTopicTolerance
19  novelEnough <- novelty >= config.noveltyThreshold
20  record candidate, benchmark, and gate decision
21  if improves and safe and novelEnough
22    best <- candidateBenchmark
23    append candidate to acceptedCandidates
24    append candidate to exploredCandidates
25    insert candidatePolicy into seenPolicies
26 return (baseline, best, acceptedCandidates, exploredCandidates)
```

This is the metaharness improvement loop. It does not directly train a model. Instead, it mutates the **instructional controller**, evaluates the mutant in the harness, and accepts it only if three conditions hold: the score improves, the weakest topic does not collapse beyond the configured tolerance, and the candidate is sufficiently novel relative to previously explored policies. That combination of mutation, evaluation, and gating is the algorithmic heart of the paper’s metaharness claim. Writing these gates as parameters rather than literals makes the point explicit: Keating’s contribution is the control architecture, not one privileged set of thresholds.

External archival evaluation

We analyzed the 22 JSON trace files stored in `test/traces/`. Because multiple traces existed for some topic x learner pairs, we retained the chronologically latest trace for each pair, yielding 16 sessions spanning four topics (**Derivative**, **Special Relativity**, **Stoicism**, and **Social Contract Theory**) and four learner models (Llama-3.2-1B, LFM-2.5-1.2B, Qwen-2.5-1.5B, and Cloud-MiniMax-M2.5). The retained set exactly matched `test/final_dataset.json`.

Each retained trace already contained three scalar labels: mastery, engagement, and clarity. We treated these as archived outcome labels. One record encoded scores on a 0-10 scale rather than the 0-1 scale used elsewhere, so we normalized that record by dividing by 10 and recorded the correction in `docs/generated/study-analysis.json`.

The external overall score was defined as the unweighted mean of mastery, engagement, and clarity. Because the dataset is small, we report bootstrap intervals rather than formal null-hypothesis tests. We also computed exploratory trace features, including empty turns, word counts, teacher redirection cues, and student-role contamination markers.

Synthetic benchmark and robustness analyses

The internal benchmark uses `src/core/benchmark.ts`. Unless a focus topic is specified, the suite evaluates 14 topics. For each topic and random seed, the benchmark samples 18 synthetic learners and computes topic-level mean scores from mastery gain, retention, engagement, transfer, and confusion. The current study compared the repository default policy with the current evolved policy in `.keating/state/current-policy.json` over 200 seeds.

To probe overfitting, we separately summarized tuned-topic (**Derivative**) and non-tuned-topic mean deltas. To probe mechanism, we performed one-at-a-time ablations by swapping each current-policy parameter individually into the default policy and re-evaluating over the same 200 seeds. To probe optimization stability, we reran derivative-only evolution 30 times from the default policy using `src/core/evolution.ts`.

Statistics and reporting

All derived numbers in the manuscript come from `bun scripts/study-analysis.mjs`, which writes `docs/generated/study-analysis.json` and `docs/generated/study-analysis.md`. The marimo notebook at `analysis/study_analysis.py` provides an inspectable analysis surface for peer review and exploratory review. External descriptive intervals were estimated with non-parametric bootstrap resampling. Synthetic robustness summaries are reported as empirical means and percentile ranges across seeds or reruns.

Limitations

The strongest limitation is scope. No human participants were studied, no intervention was preregistered, and the archival labels do not currently include inter-rater agreement or scorer provenance.

The present paper should therefore be read as a systems-and-methods paper with an audited archival evaluation, not as a completed human-learning trial.

A second limitation is harness shape. The current benchmark strongly rewards retrieval pressure and overload control, but is less sensitive to reflective depth in isolation. That limitation is not hidden; it is part of what the metaharness diagnosis reveals.

A third limitation is data volume. Four topics are enough for informative failure analysis, but not enough for stable cross-domain claims about all forms of instruction.

Code Availability

All study logic required to reproduce this paper is contained in the repository. The main entry points are `scripts/study-analysis.mjs`, `analysis/study_analysis.py`, `src/core/benchmark.ts`, and `src/core/evolution.ts`.

Data Availability

The raw archival traces analyzed here are stored in `test/traces/`. The curated snapshot is `test/final_dataset.json`. Derived analysis artifacts used by the manuscript are written to `docs/generated/study-analysis.json` and `docs/generated/study-analysis.md`.

Bibliography

- Bloom, B. S. (1984). The 2 Sigma Problem: The Search for Methods of Group Instruction as Effective as One-to-One Tutoring. *Educational Researcher*, 13(6), 4–16. <https://doi.org/10.3102/0013189X013006004>
- Burnett, L. K., & Richmond, L. L. (2026). Meta-analytic Investigations of the Effect of Cognitive Offloading on Memory-based Task Performance and Interindividual Variability. *Memory & Cognition*, 54(1), 144–168. <https://doi.org/10.3758/s13421-025-01743-8>
- Chi, M. T. H., Bassok, M., Lewis, M. W., Reimann, P., & Glaser, R. (1989). Self-Explanations: How Students Study and Use Examples in Learning to Solve Problems. *Cognitive Science*, 13(2), 145–182. [https://doi.org/10.1016/0364-0213\(89\)90002-5](https://doi.org/10.1016/0364-0213(89)90002-5)
- Kestin, G., Miller, K., Klales, A., Milbourne, T., & Ponti, G. (2025). AI Tutoring Outperforms In-Class Active Learning: an RCT Introducing a Novel Research-Based Design in an Authentic Educational Setting. *Scientific Reports*, 15, 17458. <https://doi.org/10.1038/s41598-025-97652-6>
- Lee, Y., Nair, R., Zhang, Q., Lee, K., Khattab, O., & Finn, C. (2026,). *Meta-Harness: End-to-End Optimization of Model Harnesses*. <https://doi.org/10.48550/arXiv.2603.28052>
- Ma, W., Adesope, O. O., Nesbit, J. C., & Liu, Q. (2014). Intelligent Tutoring Systems and Learning Outcomes: A Meta-Analysis. *Journal of Educational Psychology*, 106(4), 901–918. <https://doi.org/10.1037/a0037123>
- Risko, E. F., & Gilbert, S. J. (2016). Cognitive Offloading. *Trends in Cognitive Sciences*, 20(9), 676–688. <https://doi.org/10.1016/j.tics.2016.07.002>
- Rittle-Johnson, B., & Loehr, A. M. (2017). Eliciting Explanations: Constraints on When Self-Explanation Aids Learning. *Psychonomic Bulletin & Review*, 24(5), 1501–1510. <https://doi.org/10.3758/s13423-016-1079-5>

Thesen, T., & Park, S. H. (2025). A Generative AI Teaching Assistant for Personalized Learning in Medical Education. *Npj Digital Medicine*, 8, 627. <https://doi.org/10.1038/s41746-025-02022-1>

Zhang, J., Zhao, B., Yang, W., Foerster, J., Clune, J., Jiang, M., Devlin, S., & Shavrina, T. (2026,). *Hyperagents*. <https://doi.org/10.48550/arXiv.2603.19461>