MASTERY MODELS

DEVELOPMENTAL INTELLIGENCE FROM THE TRAJECTORY OF IMPROVEMENT

A POSITION PAPER AND RESEARCH PROGRAM

Sidney Scott
Crescive Labs
research@crescivelabs.com

November 2026

ABSTRACT

Frontier models are trained on the finished work of the human species: the published paper rather than its drafts, the merged commit rather than the broken ones. The record of how anyone got better was never kept. We argue the omission is load-bearing. Group-relative reinforcement learning produces exactly zero gradient where every attempt succeeds or every attempt fails [109, 131], and across three million student submissions a model's coding proficiency was *inversely* related to its ability to predict how a learner's code would change next [93]. **The field has learned to model states and not becoming.** World models, reward models and edit models each identify a state, or the transition of something other than the learner. None identifies the transition of the learner itself. We introduce *developmental intelligence*, intelligence learned from how competence develops rather than from what it produces, and *mastery models*, its first instance. Where a world model predicts how an environment changes under an action, a mastery model predicts how a solver changes under an attempt. Training against a learned value over human improvement we name *mastery-optimized reinforcement learning*, successor to the regimes built on what people wrote, what they prefer and what can be checked. The claim concerns what is reachable under finite compute rather than the ceiling: outcome supervision is starved of gradient rather than wrong [84]. We then make a structural claim. Continual development, abstraction acquisition, transferable learning procedures, the compute that capability requires, experiment selection in discovery, and the evaluation step on which self-improvement stalls are not six agendas. Each needs a judgment about whether a learner moved toward competence, and that judgment is this object. One result carries all six: whether a value fitted to sub-frontier human improvement stays useful on solvers stronger than the records that made it. It is one of twenty-three dated kill tests, four of them lethal, and it resolves in twelve months.

Keywords developmental intelligence · mastery models · process supervision · improvement traces · reward shaping · knowledge tracing · open-ended learning · learning sciences

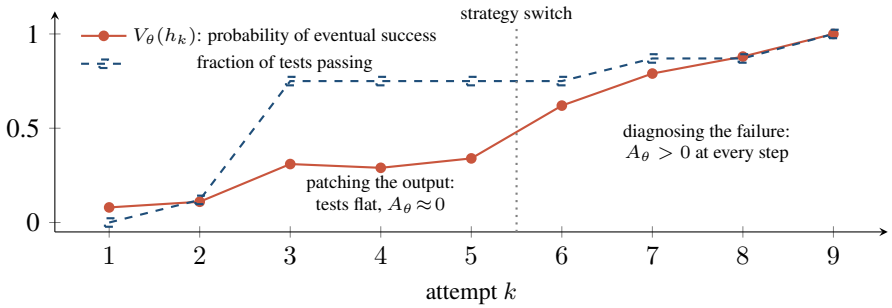


Figure 1: Anatomy of an improvement trace. The series come apart between attempts 4 and 6: passing tests stay flat while the solver patches outputs, then rise once it begins diagnosing the failing test. Outcome supervision sees only attempt 9; a mastery model assigns the credit to attempt 6. *Schematic, not plotted from data.*

1 Introduction

Consider a first-year student asked to write a function returning the second-largest element of an array. Attempt 1 returns the largest. Attempt 3 sorts the array and returns index 1; it passes six tests and fails on duplicates. Attempt 4 adds a special case for duplicates in the wrong place. Attempt 6 is different in kind: the student stops patching the output and starts reading the failing test. The next change is small and correct. Attempt 9 passes.

The internet contains attempt 9. It is on Stack Overflow and in a thousand repositories. Attempts 1 through 8 exist nowhere that a model has ever seen. And the most valuable event in the sequence is not attempt 9. It is the decision made between attempts 4 and 6, when the solver changed strategy from fixing answers to diagnosing failures. That decision is what competence is made of.

This example is a schematic composite, but its shape is measured. CodeInsight [93] records over three million submissions from 3,286 undergraduates across 394 problems in two introductory programming courses, every attempt with test-case-level outcomes, timestamps, and source. Modeling that record produced a result no schematic could: in that setting, the more proficient a language model was at writing code, the worse it was at predicting how a learner's code would change on the next attempt. A recurrent state space model with discrete latent variables, a far weaker coder, achieved the strongest predictive accuracy on three of the four courses. The authors conclude the language model is better understood as a generative solver conditioned on context than as a faithful predictor of the trajectory.

Disclosure. A member of the founding group (Appendix B) is an author on CodeInsight. No CodeInsight data transfers into this program, for the reasons given in Section 8; what transfers is the pipeline design.

The civilizational observation. Human cumulative culture operates as a ratchet [116]: each generation inherits the finished tools of the previous one and improves them, without inheriting most of the failed attempts that produced them. This is efficient for a civilization and catastrophic for a learner trained only on the inheritance. Reichenbach [101] named the distinction that governs what survives: science publishes the context of justification and discards the context of discovery. The printed record replaced apprenticeship, in which the novice watched the master fail and correct [64], with the master's finished work. Laboratory notebooks and draft manuscripts were the last institutional record of process and were never systematically archived. Scientific progress is itself a revision history [62] of which only the accepted revisions were published. Tacit knowledge [97] is largely knowledge of process, and it is absent from the corpus for the same reason.

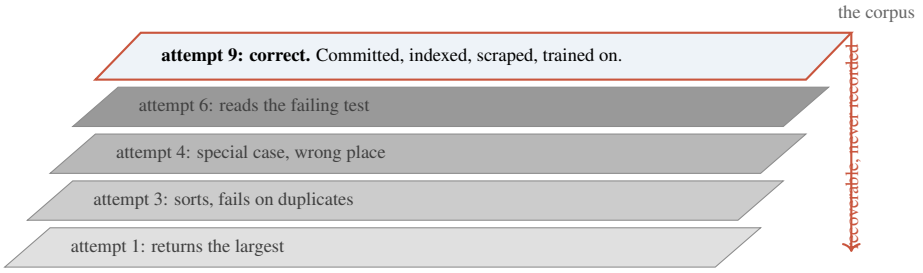


Figure 2: The corpus as a palimpsest. Only the top layer survives into public data. Every layer beneath it was produced, was informative about how competence developed, and was scraped away before any model saw the page. The claim of this paper is that the discarded strata carry the signal that outcome supervision cannot supply, and that they must be manufactured because they were structurally discarded rather than merely uncollected (Section 2).

Why the omission is inherited rather than chosen. It would be convenient to treat this as an oversight of machine learning. It is older than that, and the pattern is worth stating because it explains why the gap is structural rather than accidental.

Every recording technology that expanded what a civilization could transmit did so by capturing a state and discarding the process that produced it, and each expansion was also a loss. Writing preserved conclusions and not the reasoning that reached them, a trade Plato objected to in the *Phaedrus* on the grounds that written words cannot answer a question. Print preserved the master's finished work and displaced the apprenticeship in which a novice watched the master fail and correct [64]. The scientific paper preserved the context of justification and discarded the context of discovery [101]. The version control system preserved the merged commit; the laboratory notebook, which was the last institutional record of process, was never systematically archived.

Each of these was a step change in what could be transmitted across people and generations, and each narrowed what could be transmitted about how anything was arrived at. The ratchet [116] works by passing on finished tools, not the failures that produced them, and five thousand years of selection for transmissible artifacts produced a corpus in which process is systematically absent.

Machine learning did not choose this. It inherited it, because the corpus it trained on is the output of that selection. And there is an irony worth naming: the first technology capable of recording process at scale, cheaply, continuously, across millions of people, is the one currently being trained on the absence of it.

We have no measured figure for the fraction of human intellectual process that survives in public corpora, and we decline to invent one. The nearest empirical evidence is indirect: a filtered pass over Common Crawl with a trained classifier found the open web does not contain many long reasoning traces [40], and the largest public corpus of student programming process states plainly that current data models what people produce and not how they produce it [102].

The same pattern, in the last eight years. The civilizational version of this argument is long and the machine-learning version is short, and the short one is checkable against living memory. One reading of the period since 2017 is that the architecture has been approximately fixed and that every step change came from a different human signal becoming trainable for the first time. Pretraining learned what people wrote. RLHF learned what they prefer. Reinforcement learning from verifiable rewards learned what can be checked. Three regimes, one architecture.

Three properties of that sequence matter more than the sequence itself. Each regime was unlocked by a corpus rather than by a model. Each corpus took years to assemble and could not be bought in a quarter. And each created a durable position for whoever assembled it first, which is why the capability frontier and the data frontier have been the same frontier for eight years.

On that reading the question is not what architecture comes next but which human signal has not yet been made trainable. We argue there is exactly one of consequence remaining, that it is the record of how people got better, and that it is the only one of the four that was never written down. The first three regimes consumed corpora that already existed and merely had to be collected. This one has to be manufactured, which is the whole of the difficulty and the whole of the opportunity: the training recipe that consumes improvement traces is a dense signed shaping term with published invariance properties [117], and the corpus does not exist.

Two qualifications on the claim, and the second is the one that matters. The signal has not been trained on *as such*; recorded human improvement has never been fitted as a value and used as a policy reward, which is the precise form of the negative claim defended in Section 2. And whether it needs to be recorded at all is an open question rather than a settled one. Gandhi et al. [33] show that synthetic traces exhibiting the right process behaviors, generated rather than collected, can lift a model that had plateaued under outcome reinforcement learning. If that substitution extends to the regime this paper targets, the fourth signal is trainable without the fourth corpus, and the manufacturing argument above is an expensive detour around a cheaper road. Hypothesis 22 decides it, and it is lethal.

The bet. *Intelligence is legible in the trajectory of improvement, not in the finished artifact. Outcome supervision cannot reach open-ended competence at any compute budget that will actually be spent, because on the tasks that matter most the gradient is exactly zero. A model trained on the trajectory of improvement reaches competence that outcome training does not reach in practice, and the advantage widens with scale.*

The claim is about reachability under finite compute, not about the asymptotic ceiling, and this is deliberate. Outcome-verified RL is not wrong at the optimum; it is starved of gradient (Section 3, Figure 4). An optimum that no realizable training run approaches is not a capability. Section 5.4 makes the corresponding formal concession and shows why it costs the thesis nothing.

The consensus this opposes is that scaling reinforcement learning on verifiable outcomes over ever-larger task environments is the road to open-ended capability [22, 109]. Our claim is correct only if that program is specifically wrong: if outcome supervision at scale discards the information that open-ended improvement requires. Section 3 assembles the evidence, including the evidence against us. Section 11 specifies how the claim dies if it is false.

Contributions.

1. We identify a structural blind spot: the field models states and not becoming, and the transition of the *learner* is the one object nobody builds. We show that six established objects each identify a state or a transition of something other than the learner, that the omission is inherited from a corpus selected over millennia for finished artifacts rather than chosen by the field, and that two 2026 results from Google and from a frontier laboratory arrive adjacent to the missing object and walk past it. We then name and define *developmental intelligence*, *mastery models*, and *improvement traces* (Section 2).

2. We assemble three quantitative lines of evidence that outcome supervision has a ceiling made of missing process data, and state the strongest published counter-evidence to each (Section 3).
3. We position the proposal against the two live alternatives, self-generated experience and static behavioral simulation, and against the four nearest formalisms: process reward models, knowledge tracing, edit-prediction models, and world models (Section 4).
4. We formalize mastery models over a latent competence state, derive the *mastery advantage*, and address directly the objection that a value-difference reward is potential-based shaping and therefore leaves the optimal policy unchanged [84]. The claim is about learning dynamics, not optima, and two of the three terms in the objective are not potentials (Section 5).
5. We identify covariate shift between human and policy histories as the central technical risk, not merely a reward-hacking risk, and specify re-anchoring as its treatment (Sections 5 and 9).
6. We correct a common error in the learning-curve literature and replace it with a two-level consistency check: individual learners follow exponential curves, and power laws appear only under aggregation over heterogeneous rates [46] (Section 7).
7. We define *mastery evaluations* validated against human learning outcomes rather than against frontier failure. An evaluation designed to make frontier models fail would be circular, which is the methodological line that separates this program from benchmark-led alternatives. We present twenty-three preregistered hypotheses with dated kill tests, five of them lethal or near-lethal (Sections 10 and 11). We further argue that six capabilities usually pursued as separate agendas are consequences of this one object, and state that architecture as a hypothesis rather than an assumption (Section 13.1).

2 The Blind Spot: States, Not Becoming

2.1 One omission, four instances

The three results of Section 3 are usually read as three separate problems: a reliability problem, an exploration problem, and a curious inversion in one dataset. We read them as symptoms of one structural omission, and stating it is the main conceptual contribution of this paper.

The field has learned to model states and not becoming.

Table 1: What the field's objects identify. Every row is either a state or a transition of something other than the learner.

Object	Identifies	Whose transition
World model	How an environment changes under an action	The environment's
User model	What a person does at a point in time	None; a state
Reward model, PRM, rubric	Which outputs or steps to prefer	None; a state
Edit-prediction model	What change comes next	The artifact's
Knowledge tracing	Whether the next tagged item is answered correctly	Partial; no revision content
Calibrated decision model	A typed judgment about the current state, with a probability	None; a state
Exact replay over history	What a recorded run would have done in another order	None; a recording
Mastery model	**How a solver changes under an attempt**	**The learner's**

The pattern is not that these objects are wrong. Each is correct for what it identifies, and several are excellent. The pattern is that the transition of the learner is the one row nobody occupies, and the absence is structural rather than accidental: it follows directly from a corpus assembled by five thousand years of selection for finished artifacts.

The newest entry in the table arrived while this paper was being written. TypeSafe AI released Jev in September 2026, the first of what it calls System One models: a model that returns typed values over a predefined schema, each carrying a calibrated probability, from a structured program state rather than generating text [118]. Its training method is named Reinforcement Learning for Calibrated Decisions and is described as optimizing predicted probabilities against realized outcomes rather than against human rater preferences. That is a regime claim of the same shape as the one in this paper, made independently and with capital behind it: a successor to preference supervision, anchored to what actually happened. We read it as support rather than as a rival, because the object it fits is one the table already contains. A calibrated decision over a program state is a state, however well calibrated. The probability attaches to a judgment about the present rather than to the change between a learner's successive attempts, and the quantity a mastery model extracts is the difference between consecutive values rather than any single one. The strongest recent argument for outcome-calibrated probability stops where every other row stops, which is the claim of this section restated from

outside it. Tier 2: the architecture and training data are undisclosed and the performance figures are self-reported on the vendor's own workflows.

Two recent results are evidence for the blind spot rather than against it. Zheng et al. [136] improve exploration from a system's own recorded history using exact replay, at frontier scale, with released code, and build no learned transition of the learner at all. humans& [50] match human conversational behavior at roughly two orders of magnitude above frontier baselines and report no measurement of whether any simulated person acquires a skill. Both groups are one step from the object and had every reason to take that step. Neither did, which is what a blind spot looks like from outside: not an open problem people are racing toward, but a place people keep arriving next to and walking past.

The claim, stated precisely enough to be false. The negative claim this paper rests on has three conjuncts: no published system trains *a value function over improvement*, *from human traces*, and uses it as *the reward for a policy*. The unqualified form does not survive contact with the robotics literature, and the work that bounds it is worth reading on its own terms. Xiao et al. [128] learn a state value by temporal-difference bootstrapping with expectile regression, fit the critic partly on human teleoperated intervention and recovery takeovers together with egocentric human video, and use the resulting advantage to train a vision-language-action policy. All three conjuncts hold, which is what fixes the two qualifiers this paper carries throughout.

Both qualifiers are substantive rather than defensive. The first is domain: that work and the surrounding progress-reward literature [117] operate on robot manipulation, where a value over task progress fitted from human corrections is now a standard construction. The second is semantic: their value is over *task-progress states*, the expected time or steps to completion, and the human data supply off-distribution recovery exemplars for a robot critic. It is not a value over the *learner's own competence trajectory*, and the learner whose transition is modelled is not the source of the human traces. With both qualifiers the claim survives: in language, reasoning and symbolic domains, no published system fits a value over a solver's competence trajectory from human improvement traces and uses it as a policy reward. The nearest approach from the data side rather than the method side is Yan et al. [130], treated below, who release annotated human improvement transitions in a symbolic domain and correlate them with final standing without fitting a value over competence or using one as a reward; the second and third conjuncts are untouched. That is a precise claim rather than a sweeping one, and precision is what makes the rest of this paper testable.

The mechanism is not the contribution either. Tsao et al. [117] train a discriminator over previously successful and unsuccessful episodes and use the visitation log-ratio as a dense process reward, with a theorem that any policy maximising it also maximises the outcome reward. A dense, signed, policy-invariant process reward derived from success and failure is therefore already published with a proof. What remains unoccupied is not the shaping mechanism but the conditioning: a reward fitted to a record of *humans improving*, rather than to the system's own successful and unsuccessful episodes. Section 4.3 argues why that distinction is load-bearing rather than cosmetic.

The endpoint is not empty either. Termination in a human record of realized outcomes, row five of Table 2 in Section 4.3, has been reached from another direction: Gong et al. [37] fine-tune evaluators on institutional publication decisions and recover judgments that exceed frontier models. That is a scoring function over a human record of outcomes. It scores *final artifacts*, not transitions of a learner, which is the same distinction as above one level up, and it establishes that the endpoint is available rather than hypothetical.

The omission is now named by others, and still unfilled. Duan et al. [25], a 79-page roadmap from 35 authors across Shanghai Jiao Tong, Tsinghua, ByteDance, Shanghai AI Lab and others, organizes recursive self-improvement into five levels of autonomy. Their L3 is "Learning-Signal or Experience-Acquisition Autonomy," in which a system determines the experience needed for its next improvement round, and their L5 is a system that persistently revises "an improver, *verifier*, or successor-generation procedure." The verifier is thus named as a first-class, revisable component of the improvement loop.

Their open-problems section states the gap in terms this paper would not improve on. Discussing learner-conditioned experience acquisition, they observe that "neither correctness nor estimated difficulty alone establishes learning value," that an adaptive curriculum may "repeatedly select ambiguous tasks, reinforce evaluator errors, or concentrate on a narrow region where progress is easy to measure," and that future acquisition mechanisms "should estimate learning value across multiple rounds." A quantity that estimates learning value across multiple rounds, conditioned on the learner's state, is Definition 3 written as a research challenge rather than as an artifact. Thirty-five authors surveying the field named the missing row and did not build it, which is the clearest available evidence that it is a blind spot rather than an unsolved problem people are racing toward.

And now measured, by people who then stopped at the edge of it. Yan et al. [130] is the closest any released work has come to this paper's object, and reading it carefully is the strongest argument available that the row is empty rather than merely unfashionable. They pair 4,465 human Kaggle trajectories across 134 competitions with agent runs on seven of them, under one version-level schema in which every code version carries its score and every transition carries the action taken, its intent, its magnitude and its effect on the score. Their Table 1 surveys nine long-horizon agent benchmarks and reports that theirs is the only one retaining scored intermediate versions, their code, and task-matched human development. That table is this paper's premise, assembled independently and published.

What they then measure is the distinction of Section 4.3 in someone else's numbers. Humans pivot on 25% of transitions, one scaffold on 9% and the other on 58%, and the high-pivot scaffold gains nothing from it: its post-pivot steps average -0.008 against $+0.089$ for matched humans, so the deficit is not the rate of change but the judgment of which change was an improvement. On returning to abandoned work the separation is near-total. Top humans revisit an earlier approach on 9.1% of eligible versions and 78.5% of those returns finish above the version they went back to; across all runs one scaffold revisits once in 658 eligible versions and the other never in 344. The same agents recover *setbacks* at or above the human rate, 89% against 79.4%. An agent that climbs back from a bad score as well as a top human, and essentially never reopens a line of work it set aside, is the clearest external evidence we know of that recovering a score and resuming an approach are different capabilities.

Their intervention is the part that bears on this paper's thesis most directly, and it cuts twice. A planning prompt of roughly one thousand tokens, distilled from the human trajectories, moves three named behaviors onto the human value and improves the best score in five of seven competitions. Human process knowledge transfers to a machine solver, which is the weak form of the conjecture of Section 6 and the only direct evidence for it that exists. But the prompt reaches only what it names: it saturates where the agent already exceeds human practice, it overshoots what it forbids, and on the rhythm it does not name it moves the wrong way. Assigning each version to its dominant working mode, the harnessed run changes mode on 2.2% of steps against 13.2–14.1% for every human cohort and 7.1% for the same scaffold unharnessed. Instruction improved the checklist and degraded the dynamics. Their discussion draws the conclusion we would draw: what is wanted is "a controller that reads where the run stands rather than a stronger base model," and "final scores cannot separate these cases."

That is this paper's artifact named as future work by the group holding the data, and it is why the corpus does not close the claim. TraceML annotates transitions and correlates nineteen trajectory features with final standing; it fits no value over a solver's competence, and nothing in it is used as a reward for a policy. Two of the three conjuncts of the negative claim above are untouched by it. What it does establish, at a scale an order of magnitude below the target of Section 8 and in one domain with a free verifier, is that the substrate can be built, that the behavioral gap it exposes is real and large, and that the obvious next step from it is the one this paper specifies.

What follows from naming it. A missing row in Table 1 implies more than one artifact. A model of how a learner transitions is the object a curriculum needs in order to select what to encounter next, the object a self-improvement loop needs in order to score whether a modification was an improvement, and the object a simulated population needs in order to develop rather than merely persist. We pursue one of these and note the others in Section 13. The claim of this paper remains narrow and falsifiable; the frame it sits inside does not.

2.2 The three objects

Definition 1 (Developmental intelligence)**.** The form of intelligence learned from how competence develops rather than from what competence produces. Pretraining on the human record is *phylogenetic*: it ingests the accumulated artifacts of the species. Developmental intelligence is *ontogenetic*: it ingests the record of individuals getting better.

Definition 2 (Improvement trace)**.** A sequence $\tau = (x, (a_1, o_1, t_1), \ldots, (a_T, o_T, t_T), y)$ where x is a task, a_k the solver's k-th attempt, o_k the feedback received, t_k a timestamp, and $y \in \{0, 1\}$ whether the solver eventually reached criterion. A trace preserves every intermediate state, including failures, and records whether the solver continued or quit.

Definition 3 (Mastery model)**.** A learned model of how competence develops. Given a solver's history and a candidate next attempt, it predicts the probability of eventual success, the distribution over productive next moves, and the probability the solver continues rather than quits. A world model takes an environment state and an action and predicts the next state of the environment; a mastery model takes a solver's state and an attempt and predicts the next state of the solver.

The two objects are halves of one loop, and only one half has been built (Figure 3).

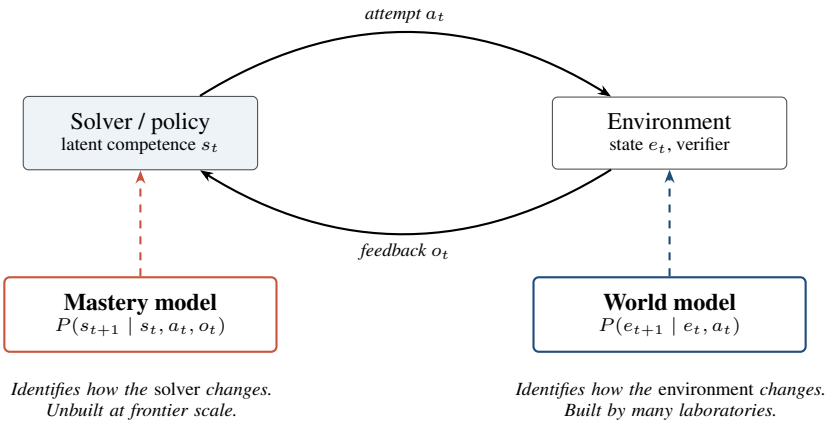


Figure 3: The two halves of the agent loop. Every substrate-defining laboratory founded to date models the environment side. The asymmetry is not accidental: environment dynamics can be simulated or scraped, whereas solver dynamics require a record of solvers improving, which is the record civilization discarded.

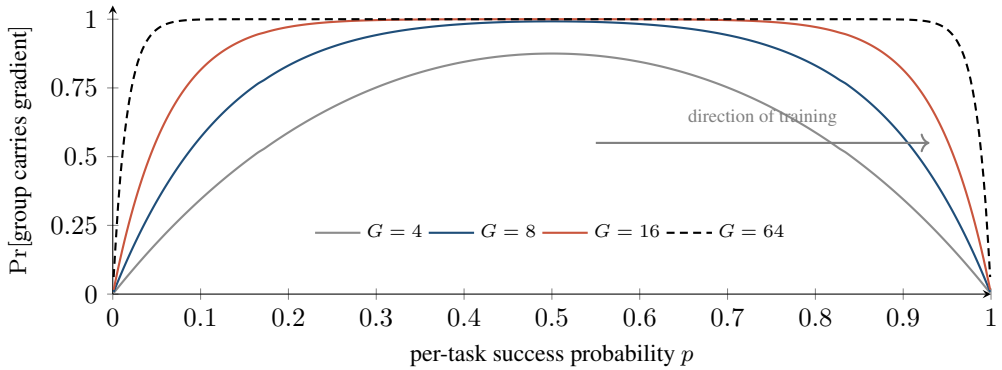


Figure 4: The zero-gradient collapse. For a group of G rollouts on one task, the probability that the group contains both a success and a failure, and therefore carries any gradient at all, is $1 - p^G - (1-p)^G$. This is exact, not simulated. As a policy improves, tasks migrate rightward and the usable fraction of the environment pool collapses; larger groups widen the usable band only logarithmically. Dynamic sampling [131] detects the collapse after spending the rollouts. Equation 8 replaces the binary outcome inside the group advantage, so a group in which every attempt fails still carries signal.

3 Evidence That Outcome Supervision Has a Ceiling

The current path to open-ended capability is reinforcement learning from verifiable rewards: a policy attempts a task many times, an automated verifier scores the final answers, and successful attempts are reinforced [22, 109]. We present three lines of evidence that this path has a ceiling, and in each case the strongest published argument against us.

3.1 The exploration wall

In group-relative methods the advantage of an attempt is computed against the mean of its group, so a group in which every attempt succeeds or every attempt fails contributes exactly zero gradient [109]. This is not a subtle effect. DAPO names it gradient starvation from zero-advantage batches and treats it as a first-order engineering problem, filtering prompts with accuracy exactly 1 or 0 and reaching 50% on AIME 2024 with a 32B model in half the training steps of the prior open recipe [131]. Training signal exists only in the band where the model sometimes succeeds, and that band narrows as accuracy rises. An independent result points at what the band is starving the policy of. Gandhi et al. [33] find that priming a model on synthetic traces of *incorrect* solutions that nonetheless exhibit verification, backtracking, subgoal setting and backward chaining improves subsequent outcome-RL about as much as priming on correct ones, while empty and length-matched chain-of-thought controls do not help. What the policy needed was not more correct answers but the shape of a productive attempt. The result is confined to Countdown at the 3B scale and we claim nothing beyond it, but it is direct evidence that the signal carried by a process is separable from the correctness of the artifact it produced, which is the premise this program rests on.

The consequence is measurable at the level of capability. Yue et al. [132] find that base models achieve higher pass@k than their RLVR-trained descendants at large k, and that the reasoning boundary often narrows as RLVR training progresses, while distillation genuinely introduces new capability. Wu et al. [123] give the theoretical form: RLVR acts as a conservative reweighting of the base distribution, and the shrinkage of empirical support generally outweighs its expansion at large sampling budgets.

ExpRL [127] localizes the constraint to coverage rather than reward and supplies a remedy: hide a reference solution from the policy, use it to build a problem-specific rubric for a judge, and give dense process-level reward for partial progress. As RL priming on hard mathematics this beats supervised fine-tuning, sparse-reward GRPO, and self-distillation.

The gap is now measured at the level of behaviour, not outcome. Yan et al. [130] pair human and agent work on the same machine-learning competitions under one version-level schema: 4,465 human trajectories across 134 competitions, with 430 paired human and 207 agent trajectories on the seven competitions worked by both. Every code version carries its score, its timestamp, and labels for the action taken, its intent, the edit size, and the score effect. Their framing of why outcome benchmarks cannot see the cause is this paper's premise stated independently: such benchmarks grade the final submission and discard the development process behind it.

Read at version level the difference is behavioural rather than one of skill. Experts alternate data work, validation, model changes and ensembling, and *return to approaches they had set aside*. Each agent scaffold instead collapses into a narrow loop, one re-weighting ensembles and another mutating its model in place, and neither pivots at the human rate nor reopens abandoned work. That is the strategy switch of Section 1 and the persistence quantity of Equation 7, measured across hundreds of paired trajectories rather than asserted.

Corroboration from the frontier. The diagnosis is not ours alone. Zheng et al. [136], from Google and Google DeepMind, open by naming exploration as "the one component still hand-written and frozen" and state the credit-assignment problem in the same terms this section does: judging an exploration policy requires watching it steer an entire discovery run to the end rather than scoring one candidate, so meta-level feedback is delayed and expensive. That is the coverage argument stated by the party best positioned to know it is wrong.

The zero-gradient condition, measured. The coverage argument is usually made analytically. Wu et al. [124] measure it. Their harness-evolution loop rolls out each task K times and can only extract a learning signal from tasks where some rollouts succeed and others fail, so they report the partition directly: across their evolution set, 40.7% of task groups yielded a usable success-failure contrast, 36.0% had every rollout succeed, and a further 14.6% had every rollout fail with no historical success to pair against. Slightly over half of all groups therefore carried no signal at all, which is the zero-advantage condition of Section 3 arising in a working system rather than in an argument.

The second measurement is the one that matters more. They track the usable-contrast fraction across evolution epochs and it *falls*, from 36.7% to 34.2% over three epochs, which they attribute to the harness absorbing the deficiencies that produced mixed outcomes. The signal a self-improvement loop depends on is consumed by the improvement it produces. That is the coverage limit as a dynamic rather than a static property, and it is why we treat exploration machinery as a component of the objective in Section 5.5 rather than as a preprocessing step.

The narrowing is method-dependent, and we state the condition. svs [3] show that self-play with variational problem synthesis improves pass@k across all k on competition mathematics, while standard RLVR on an easier benchmark leads at small k and is overtaken at large k. The boundary-narrowing result of Yue et al. [132] is therefore a property of particular training recipes rather than of outcome supervision as such. We hold the conditional form: outcome-verified RL is coverage-limited *absent explicit exploration machinery*, and the machinery in every counterexample is assembled by hand. A mastery model is a proposal for where that machinery comes from, and Hypothesis 2 is specified against the strongest hand-assembled version rather than the weakest.

The strongest counter-evidence, and what it settles. Liu et al. [71] show that prolonged RL with KL control, reference-policy resetting, and a diverse task suite *does* uncover reasoning strategies inaccessible to the base model under extensive sampling, including tasks where the base fails at every attempt. This is the strongest counter-evidence in the literature and we do not dispute it. It refutes the strong form of the exploration wall, that RLVR can never expand the base distribution. It does not refute the operative form, that RLVR is coverage-limited absent explicit exploration machinery, and it is consistent with our position: ProRL's ingredients *are* exploration machinery, assembled by hand. A mastery model is a proposal for where that machinery should come from.

What ExpRL does not establish. ExpRL shows that dense reward beats *sparse* outcome reward, which is a weaker claim than it is usually read as, and its two variants separate the axes cleanly. Scoring a whole rollout against the hidden reference and scoring each prefix of it land within one to two points of each other on every benchmark, with the whole-rollout variant ahead on three of four. Subdividing a single attempt is therefore not where the gain lives, and that null is evidence for the axis this paper proposes instead: the unit of comparison is one attempt against the solver's own previous attempt, not one segment of a chain of thought against the next. The dense-reward advantage is also domain-bound. At the 8B scale on LiveCodeBench, sparse GRPO outperforms the ExpRL-Outcome arm, so the published method that most directly threatens this program has not been shown to hold in code. What ExpRL does not show is where process reward comes from at scale. Its rubrics are authored from reference solutions, one per problem. Domains with reference solutions are a small fraction of the domains that matter, and an authored rubric encodes what a correct answer looks like, not what productive progress toward one looks like. The frontier can adopt ExpRL immediately. The learned-rubric line (Section 4) partially answers the authorship objection, generating instance-specific criteria rather than requiring a human to write them, and it is the strongest available response short of a longitudinal substrate. It remains stateless across attempts: a rubric scores a response against criteria for the task and cannot express that the same edit is progress from one history and regression from another. What the frontier cannot adopt is a learned, signed model of productive progress conditioned on the solver's own prior state.

The cheapest progress signal is free, and it is still not a substitute. Before any corpus is collected, a policy can derive a difficulty signal from itself. Xiang et al. [125] track the per-prompt solve rate across rollouts during training and scale a length penalty inversely to it, spending tokens where the policy is failing and withholding them where it is not, at no data cost at all. This is the floor that any substrate argument has to clear, and it is a real floor: a learning-progress signal read off the policy's own successes needs no human, no consent framework, and no capture tooling. What it cannot do is sign a revision. A solve rate says which problems are hard for the current policy. It says nothing about whether a particular edit moved a particular attempt toward or away from eventual success, because a policy's own rollouts contain no record of a solver who was worse and then became better. The distinction is the one that separates a rubric from a mastery model, and it is why Hypothesis 2 is specified against the strongest available baseline rather than against sparse GRPO.

3.2 The trajectory failure

Language models do not accumulate competence over an interaction; they lose reliability. Laban et al. [63] tested fifteen models from eight providers over more than 200,000 simulated conversations and found an average performance drop of 39% across six generation tasks when a fully specified instruction was sharded across turns. The decomposition is the informative part: aptitude fell moderately, by roughly 16%, while unreliability more than doubled, rising roughly 112%. Their summary is that when a model takes a wrong turn in a conversation it gets lost and does not recover. Liu et al. [72] give the long-context mechanism.

The behavioral consequence is measured in deployment, and a human baseline exists. In a controlled comparison against clinical standards of care, licensed therapists responded appropriately 93% of the time while language-model therapy systems did so less than 60% of the time, with models regularly encouraging rather than interrupting delusional thinking [80]. DelusionEval [81] extended the measurement to accumulated history: across 589 conversation histories comprising 12,591 messages, failure to discourage self-harm rose from 30.0% to 41.1% when 350 prior messages were prepended, an effect uncorrelated with model size or release date. The gap is therefore not only against an absolute standard but against a human whose performance does *not* decay with the length of the relationship.

A second axis: models do not move like people. Degradation is one way a trajectory can be wrong. Flatness is another, and it was measured independently. In an evaluation of behavioral coherence across 80-turn conversations, human participants remained coherent in 87.3% of conversations, losing roughly eleven points over the interaction. Frontier models lost one or two: 99.0% for Opus 5, 98.4% for GPT 6, 97.0% for Fable 5.1, and 95.0% for GPT 5.6, which held exactly flat across every horizon measured [50]. A purpose-built user model initialized from a 550B base and post-trained against an adaptive discriminator reached 60.7%, undershooting humans but declining in a human-like shape.

The same work reports that frontier models are distinguishable from humans in multi-turn group conversation at rates between 0% and 0.89% judge-fooled, against 18.6% to 21.1% for the purpose-built model, where 50% is parity. Two findings follow that this paper relies on. Producing human-like text and reproducing a human trajectory are separable capabilities, and frontier scale delivers the first without the second. And a flat coherence curve is not a trajectory: a system that does not change across an interaction has no direction for a value function to be defined over.

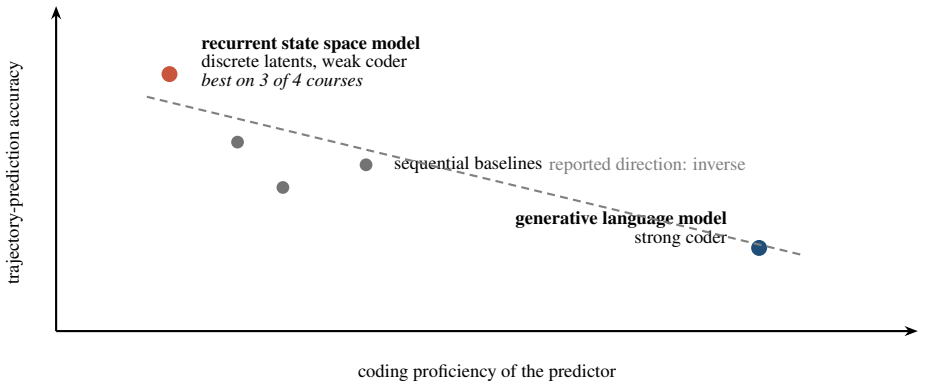


Figure 5: The proficiency inversion. In CodeInsight's benchmark the predictor best at writing code was not the predictor best at anticipating how a learner's code would change, and the reported relationship in that setting was inverse [93]. *Schematic of the reported direction; axis positions are illustrative and no values are reproduced from the source.* The winning architecture is the recurrent state space model family used to build world models [44]: the best available learner model is structurally a world model pointed at a person.

What this evidence does and does not show. These are measurements of drift and harmful behavior, not of learning, and DelusionEval reports that safety tuning reduces the effect, so degradation is mitigable rather than intrinsic. We claim only what follows: current models treat accumulated history as additional context, not as a trajectory with direction, and no published result shows a frontier model systematically gaining capability over a long interaction.

3.3 The proficiency inversion

CodeInsight [93] evaluates predictors from the parametric, sequential, and generative traditions on its three million submissions under a shared calibration and scoring protocol. A recurrent state space model [44] adapted to track solver characteristics through discrete latent variables achieved the strongest predictive accuracy on three of four courses, outperforming a language model that generates a full submission at each step. The authors report that the language model's coding proficiency was inversely related to its predictive performance in this setting. Ross et al. [102] corroborate the direction on a different corpus: state-of-the-art models often adopt problem-solving strategies unlike those of the humans whose behavior they are asked to model.

A second corroboration comes from a different domain and a different task framing. FoundationalASSIST [122] evaluates frontier models on knowledge tracing and pedagogical grounding. Every model tested barely exceeds a trivial baseline at predicting student performance, and all models fall below random chance at item discrimination, that is, at judging which of two problems better reveals what a learner misunderstands. The authors' own diagnosis is the thesis of this paper stated independently: training data must include student learning processes and not only correct solutions, because models exposed only to successful reasoning cannot be expected to anticipate unsuccessful reasoning.

Two groups, two domains, two task framings, one direction. We still decline to generalize to all domains, and Hypothesis 1 remains the test.

That the winning architecture is a recurrent state space model is not incidental. It is the architecture family used to build world models [44]. The best available learner model is already, structurally, a world model pointed at a person.

The scope of the result. The inversion is established in one domain, introductory C++, and the state space model wins on three of four courses rather than all four. We do not generalize past the setting. The direction is what matters, and it is the wrong direction for a program that assumes better artifact production implies better trajectory understanding.

Synthesis. The frontier has hit a coverage wall its own literature is patching by hand; its models lose rather than gain reliability over an interaction; and getting better at the artifact does not, in the one place both were measured, make a model better at the trajectory.

4 Related Work and Position

4.1 The two live alternatives

Two well-articulated programs address the same shortage of training signal and reach opposite conclusions about where new signal comes from.

4.2 The same observation, already made about machines

The claim that finished work omits the process that produced it is not ours alone, and its clearest statement is about machine reasoning rather than human improvement. Xiang et al. [126] argue that the true data-generating process for hard reasoning is a non-linear search with backtracking and verification, that written solutions record only its linearized output, and that a model trained on solutions therefore never observes the procedure that found them. Their evidence is a generator-verifier gap: a fine-tuned 8B model reaches roughly 85% at pass@64 on MATH against 40% greedy, so the solution lies inside the model's reach and the search that locates it does not. Their response is to recover the latent search synthetically, through process supervision and search traces over the model's own rollouts. The gap is a measurement they report; the pipeline they propose for closing it is a roadmap, and they publish no benchmark results from running it.

This paper takes the same observation and changes the substrate. Where Xiang et al. [126] reconstruct a missing process from machine search, we propose fitting a value function to a process that human beings executed and that was discarded at the point of publication. The two are complementary rather than competing, and the distinction is the one drawn in Table 2: a search trace over a model's own rollouts terminates in the model, while a human improvement record terminates outside it. Hypothesis 5 is the test of whether that difference carries information.

The nearest published substitute, and what it costs us to acknowledge. Gandhi et al. [33] supply the empirical follow-through that the Meta-CoT roadmap lacks, and the result is the closest thing in the literature to a refutation of this paper's substrate claim. Two 3B models trained identically with outcome reward on Countdown diverge: Qwen-2.5-3B improves and Llama-3.2-3B plateaus. The authors attribute the gap to four cognitive behaviors present in one base model and absent in the other, namely verification, backtracking, subgoal setting and backward chaining. Priming Llama on synthetic traces exhibiting those behaviors before reinforcement learning closes the gap, and continued pretraining on mathematical text filtered to emphasize them does the same. Their sharpest finding is that priming on *incorrect* solutions that display the behaviors works about as well as priming on correct ones, while empty and length-matched chain-of-thought controls do not help at all.

We read this as support for the first half of our thesis and a live threat to the second. Process beats product: what carried the result was the shape of the reasoning rather than the correctness of the answer, which is the premise this paper is built on. But the process data was synthetic, the reward stayed a per-episode binary outcome on a task with a free verifier, and no human improvement record was used anywhere. The intervention is on the initial policy; ours is on the reward. The regime is one where outcome reward already works for one of the two models; ours is the regime where it is flat for every model. Those distinctions are real and they are also exactly the kind that dissolve under a stronger version of someone else's method. If synthetic process priming reaches the same place that a human improvement corpus reaches, the corpus is not the scarce input and this program's central economic claim fails with it. Hypothesis 22 is that test, and we have added it as a baseline arm rather than a remark.

Silver and Sutton [110] argue that the era of human data is ending, that its returns are exhausted, and that future agents will learn from self-generated streams of experience grounded in environment reward. Hughes et al. [49] make the companion case that open-endedness is a precondition for superhuman intelligence. The strongest concrete instantiation to date is Buehler [18], in which an agent given one prompt and five images builds physics-based fracture laboratories, and a swarm of hundreds of agents then explores the resulting design space without a central planner, recovering three controls of hierarchical metamaterial fracture that replicate across independent runs. On this view, human data is the constraint to escape.

That system is also the clearest illustration of what the camp is missing. The swarm's exploration signal is a terminal physics verifier: did the structure hold. In a combinatorial, path-dependent space where most trajectories fail, that signal is identical across every failure and says nothing about which failed trajectory was closer, which is the coverage failure of Section 3 in a domain with a free verifier. An executable world supplies a place to act. It does not supply a criterion for which actions constituted progress.

At the other pole, behavioral simulation manufactures human data deliberately, and as of September 2026 it has a shipped artifact rather than only papers: a 550B user model released as a research preview with a playground and an API, trained to match the distribution of human behavior in multi-turn, multi-user conversation [50]. Generative agents [90]

equipped agents with memory, reflection, and planning; interview-grounded agents reproduced real individuals' survey responses at 85% of those individuals' own test-retest reliability [91]; fine-tuning on 2.9 million responses across 210 experiments improved distributional alignment on unseen studies [14, 59]. On this view, human data is the asset to manufacture.

A third program accepts that models must improve during use and locates the problem in the architecture rather than the corpus. This line pursues adaptation without retraining: gradient-free updates, adapter merging at inference, and dynamic decoding, on the argument that scaling has become an inefficient route to capability and that the binding constraint is the cost of adapting a frozen model to the task in front of it. On this view, human data is neither the constraint to escape nor the asset to manufacture; the constraint is the update mechanism.

We agree that a static model is the wrong object and disagree about what is missing. An adaptation mechanism changes a model's behavior in response to feedback. It does not supply a criterion for which changes constitute progress, and without that criterion adaptation is steered by the same outcome signal whose coverage limit Section 3 documents. A mastery model is that criterion. The two are composable rather than competing: an architecture that can update cheaply and a value function that can say whether an update was progress solve different halves of the same problem.

Our position is the fourth vertex, sitting with the second camp on substrate and with the third on the necessity of improvement during use. We agree with the second camp that human data is the asset and must be manufactured rather than scraped. We agree with the first that the human data used so far is the wrong kind. Behavioral simulation models what a person does at a point in time; every architecture in that lineage lacks a development module, so a simulated agent remembers and reflects but does not become more capable at anything. The distinction survives the newest and strongest instance. A user model predicts what a person says next; a mastery model predicts whether an attempt was progress. The first is now demonstrably achievable at roughly two orders of magnitude above frontier baselines, and the released evaluations contain no measurement of whether any simulated person acquires a skill, because that is a different object. That group's own stated roadmap names the gap: modelling what people say eventually requires modelling how they learn. Self-generated experience models what an agent does without a model of how it improves. Developmental intelligence is the manufactured human data of *improvement*, used to supply the exploration and credit-assignment machinery that self-generated experience currently assembles by hand.

4.3 Exact replay, and where it stops

The sharpest non-mastery-model answer to the coverage problem is not a reward at all. Zheng et al. [136] observe that a completed discovery run records a tree of exploration decisions, each carrying its realized execution outcome, and that such a tree is an *exact* simulator of the search space it reached. An alternative exploration policy can therefore be scored by walking the recorded tree in a different order at zero execution cost, thousands of candidates screened against one paid rollout. Their results are strong: 2.43× fewer generations on VGG16 at comparable performance, 2.09× higher score on ConvDiv at comparable budget, and 162× fewer discovery-agent calls than a tree-search baseline on a regularization-path task. The improved policy is executable code rather than weights, and because the incumbent is in the candidate set the policy cannot get worse.

This is a genuine alternative and we do not diminish it. Where history is exact and replayable, a learned value over improvement is unnecessary, and we say so rather than defend a claim the evidence does not support.

The boundary, which they state themselves. "A policy can only be dreamt where history actually went." Three consequences follow, and together they define what a mastery model is for.

1. **Off-tree generalization.** Exact replay cannot score a branch nobody executed. A learned value can, approximately, which is the entire difference between a simulator and a model.
2. **Domains without free verifiers.** The absence of a default reward is stated plainly from inside frontier robotics: Dong and Finn [24] observe that reinforcement learning from verifiable rewards gave language modelling a default answer, checking the answer and checking the tests, and that robotics has no equivalent, so success detectors are today either hand-built per task or replaced by a person watching each trajectory and calling it, neither of which scales. We cite this for the gap and not for novelty: as Section 2 concedes, progress-value constructions already exist in that field, and value-based methods are the norm there for reasons of sample cost. What is missing is the criterion, not the machinery. Replay requires a recorded outcome at every node. All three domains in that work are code-shaped, which is not an accident. In writing, proof construction before completion, design iteration, or any physical domain where an attempt is expensive and its counterfactual was never run, there is no outcome to replay.
3. **Flat outcomes.** Replay preserves the outcome signal exactly, including its uninformativeness. Where every recorded attempt in a region failed, replaying those failures in a different order reproduces the zero-gradient condition of Section 3 faithfully rather than resolving it.

A frontier laboratory concludes that pass/fail is not enough. LLM-Core Xiaomi [73] scale reinforcement learning explicitly towards self-improvement and report that a binary test score cannot separate a minimal correct patch from one that passes by adding defensive fallbacks or retrieving the published fix, which they treat as a limit on how far outcome reward can carry agents. Their remedy has a model judge compare all attempts at a task jointly and redistribute positive advantage from weaker to stronger passing solutions, with confirmed hacks reset to zero. In a controlled ablation, removing this graded signal causes turn counts and trajectory length to climb while pass-rate gains plateau. The construction is instructive for what it leaves out. Only passing attempts are reordered and failed attempts retain zero reward, so it adds nothing where every attempt fails; it multiplies a verifier's outcome, so it cannot operate where no verifier exists; and it compares simultaneous siblings rather than conditioning on a solver's history, which places it in the first row of Table 2, gated by the fourth. It sharpens the signal inside the answer-key domain and does not extend beyond it.

The nearest construction without a value function. Wu et al. [124] extract their improvement signal by pairing successful and failed trajectories of the *same* task and diffing them, then aggregating the diffs across tasks to separate recurring deficiencies from instance-specific ones. Structurally that is what the mastery advantage measures, which trajectory diverged toward success and where, computed by a language model reading a pair rather than by a value fitted over histories. Three consequences follow. It requires both a success and a failure on the same task, so it is undefined on the half of task groups reported above where outcomes are uniform, where a learned value still yields a gradient. It compares two trajectories rather than conditioning on one solver's history, so it cannot express that a revision is progress from one state and regression from another. And the analyst is the agent under evolution, which places it in the first row of Table 2, partially offset by execution gates that ground each accepted modification in a verifier.

Their Section 6.6 is also a direct measurement of the learnability frontier: evolution data centred on medium-difficulty tasks produced a harness scoring 76.45 against 74.25 for a set weighted to easy and hard extremes, because easy tasks yield too few failures and hard tasks too few successes. That is the band Equation 5 is defined to target, located empirically by a group with no stake in this thesis.

The deeper limit: a self-referential history. The three consequences above are about where replay can look. There is a fourth about where the history itself comes from, and it is the argument this paper most needs to make.

Every tree in that simulator pool was produced by some version of the policy being improved. The pool grows each lap, but it grows along the distribution the agent already occupies. A policy dreamt against its own history can recombine orderings, branch widths, concurrency levels and stopping rules over strategies it has already run. It cannot be scored on a strategy that appeared in no tree, because the outcome of that branch was never computed and does not exist to look up.

That is the coverage limitation of Section 3 one level higher. Outcome-verified RL cannot reinforce a solution path the base model never samples; exact replay over self-generated history cannot evaluate an exploration strategy the policy never ran. The wall is the same shape and it sits one layer up, which is what makes the remedy the same in kind: a signal drawn from outside the system's own distribution.

This is not only an argument. Liu et al. [70] note that a study using held-out tasks finds evolved harnesses *overfit the tasks used during search* and deliver only marginal gains on unseen ones, which is why their own pipeline separates search feedback from final evaluation. A system improved against a record it generated about itself generalizes poorly off that record, measured rather than asserted.

Human improvement traces are off-distribution by construction, and that is the sharpest reason the substrate has to be human rather than merely recorded. Two corollaries follow. A policy does not abandon a task; it exhausts a budget, so machine history contains no quitting and cannot train the persistence head of Equation 7 at all. And the marginal value of a human trace is highest exactly where machine traces are most expensive to produce, which is the same inequality that reorders the verticals in Section 11.

A worked example of terminating in oneself. Nam et al. [82] run an autonomous research pipeline that improves 86 of 107 papers, and their loop is steered by a simulated reviewer that also grades the result, with early stopping once that reviewer's score clears a threshold. The authors concede the reviewer is in-distribution with respect to the process it scores. The improvement is real and the criterion is inside the system, which is the configuration the next paragraph formalizes.

Where the criterion terminates. It is tempting to describe a system that judges its own improvements as an infinite regress. It is not, and the actual structure is worse, because it terminates immediately. Any chain of justification can end

in only three ways: it continues without end, it circles back on itself, or it stops at something taken as given [101]. The systems in this section differ from one another in exactly one respect, which is where they stop.

Table 2: Where each improvement criterion terminates. The question is not how long the chain is but whether its endpoint lies outside the system being improved.

System	The criterion for "this modification was an improvement"	Terminates
Tree search over self-modifications [32]	A language-model judge's pairwise preference, aggregated into a strength score	**In itself**
Exact replay [136]	Recorded outcomes of branches the policy actually ran	In its own history
Harness auto-research [70]	Fixed metrics held outside the optimizer's control	In a human-authored metric
Outcome-verified RL [22]	A task verifier	In the environment; sparse
Mastery model	**A value fitted to human traces with realized outcomes**	**In a human record**

The first row is the degenerate case. A model judging a model's modifications grounds the judgment in the same prior the modifications were drawn from, so the chain closes at depth one and the system's criterion for improvement is its own existing sense of what improvement looks like. Nothing in the loop can tell it that the sense is wrong. The others do better by terminating outside themselves, and they differ in how much signal the endpoint carries: a verifier is external and sparse, a recorded history is dense and bounded by what was run, and a human record is dense, external, and off the system's own distribution by construction. That last property is the one the substrate exists to supply, and it is not obtainable by any arrangement of the system's own outputs.

They compose. The honest synthesis is not that one supersedes the other. It is that exact replay is correct where history went and a learned value is required where it did not, and a system holding both is strictly better than either. Hypothesis 17 tests this directly, using their public discovery trees, and it is the cheapest decisive experiment in the program.

A detail in that system makes the case more sharply than its results do. The exploration behavior the method depends on is still specified in the prompt, which instructs the agent to read every prior attempt before writing code and to stop making small variations on the same idea. That is the authored-rubric problem of Section 3 in miniature: the structure of productive revision has to be written out in natural language because nothing in the system can score whether a revision was progress. It sits beside their own finding that semantic guidance injected into the prompt *underperforms* the unguided baseline, and the two together are an argument for a learned signal that neither makes alone.

Three groups have now tried to close this gap with instructions, and all three report the same limit. Dream-RSI's own harness specifies the exploration behaviour in the prompt, instructing the agent to read every prior attempt and to stop making small variations on one idea, and their ablation finds that abstracting prior trajectories into directional insights and injecting them into the prompt *underperforms* the unguided baseline. Yan et al. [130] distil a planning prompt from observed human practice and find that it moves the named behaviours toward the human profile and lifts scores, but that the effort profile stays agent-shaped: instruction closes only the part of the gap that reduces to instructions. And Fu et al. [32] abandon instruction entirely in favour of a learned score, using a language-model judge's pairwise comparisons over candidate self-modifications, aggregated by a regularised Bradley-Terry model, to steer parent sampling in a tree search.

A frontier laboratory has now measured the same quantity directly, and the method is revealing. To test whether its models choose a better next step than a human researcher at points where a session went off course, Favaro and Clark [27] score each candidate step with a second model that can see how the session eventually turned out. That is a value over eventual success, computed after the fact by a model, because nothing in the system predicts it in advance. A mastery model is the object that estimates it before the outcome is known, which is the only point at which it can steer anything.

That third attempt is the closest published system to the object of this paper and the difference is precise. A Bradley-Terry strength is a relative comparator among candidates available at one moment. It is not conditioned on the solver's history, so it cannot express that the same modification is progress from one state and regression from another; it is unsigned with respect to eventual success, so it ranks rather than measures; and its judge is anchored to nothing outside the loop it steers, which is the first row of Table 2. The direction of travel is unmistakable: having found that productive revision cannot be specified in a prompt, the field is now building learned scores over it from the only signal currently available, which is a model's opinion of a model.

One finding of theirs supports this paper directly. They also tested the obvious alternative to a learned signal: abstract prior trajectories into high-level directional insights and inject them into the prompt. It *underperformed* the unguided baseline at equal budget, because strong semantic priors over-constrain the space and suppress exploration diversity. Process knowledge summarized as advice makes systems worse. That is an argument for process knowledge entering as structure or reward, which is what a mastery model is, and against the interpretation of this paper's thesis as "show the model what people learned."

4.4 The four nearest formalisms

Process reward models. A PRM scores the correctness of an intermediate step against a reference or an authored rubric. Lightman et al. [68] showed process supervision significantly outperforms outcome supervision on MATH, with the process-supervised model solving 78% of a representative test subset, and released PRM800K with 800,000 step-level human labels; Uesato et al. [119] give the earlier GSM8K comparison. The result is not uncontested. Zhang et al. [135] report that Monte-Carlo-estimation PRMs typically yield inferior performance and generalization relative to model-as-judge and human annotation, and identify an evaluation bias in Best-of-N where unreliable policies produce correct answers through flawed processes. We take both findings seriously: process supervision wins when process labels are of high quality, and the failure mode of cheap process labels is precisely the reward hacking we address in Section 9. A mastery model differs from a PRM in three ways: it consults no reference, its reward is signed so that regression is penalized, and it predicts eventual success rather than local correctness.

Procedural structure. The fourth column deserves a note because it is the nearest attempt at one of the three heads. Lu et al. [76] organize procedural knowledge as (procedure, relation, procedure) triplets, localize a solver's position in that graph at each decision step, and translate the surrounding subgraph into guidance that biases the next action without dictating it. The graph evolves by having a refiner contrast successful against failed trajectories and edit the topology, with a held-out validation gate and a rejection memory that keeps unsuccessful proposals as negative constraints.

That is the productive-move distribution $\pi^{\star}_{\theta}$ of Definition 4, built symbolically rather than learned, and built alone. There is no value over eventual success, so nothing scores whether a revision was progress; there is no persistence term, so nothing prices abandonment; and the structure is a hand-specified schema populated by a model's reading of its own traces, gated by a task verifier. A mastery model is the same head fitted jointly with the two it is missing, from human traces rather than from the system's own runs.

Learned rubrics. The authored-rubric limitation of Section 3 does not extend to the learned-rubric line, and the distinction must be drawn carefully because that line is the program's nearest commercial competitor. Rubrics as Rewards [41] generates instance-specific rubric checklists and uses them as on-policy reward, extending RLVR into domains without binary correctness and reporting substantial gains over Likert-style LLM-as-judge baselines. Rubric-ARM [6] treats rubric generation as a latent action optimized jointly with the judge; RLR-cubed [7] routes criteria to a deterministic verifier or a judge and moves verification from the task level to the criterion level. The learned-rubric arm of Section 11 is specified against the strongest published generative judge rather than a Likert prompt, namely a CoT-GenRM-style judge after Mahan et al. [78], which generates a reasoning trace before its verdict and is trained on self-generated rationales. Specifying the arm this way is deliberate: a weak judge is the easiest baseline to beat and the least informative to have beaten. The distinction that survives it is one of object rather than quality. A generative judge ranks finished outputs, however well it reasons on the way there. V_{θ} scores signed movement between one attempt and the next, so the two differ in what they take as input before they differ in how good they are.

A learned rubric and a mastery model are both learned dense rewards in domains without references, and the difference is the conditioning. A rubric scores a response against criteria for the task. A mastery model scores a revision against the solver's own prior state: the same edit is progress from one history and regression from another, and only a model carrying a competence state can express that. A rubric is also unsigned and stateless across attempts, so it cannot penalize regression or price abandonment. The rubric line is the strongest thing the frontier and its data suppliers can deploy without a longitudinal substrate, which is why Hypothesis 2 must beat it and not merely beat ExpRL.

Knowledge tracing. Bayesian Knowledge Tracing [21] and Deep Knowledge Tracing [95] are learned models of latent competence predicting whether a learner's next attempt at a tagged item will be correct. They establish that the object is learnable at scale. The precise limit matters for our claim: Khajah et al. [56] showed that once BKT is extended with forgetting, latent ability, and skill discovery, it becomes statistically indistinguishable from DKT, and that DKT's gains do not come from novel learned representations. Depth is not what makes knowledge tracing work. A mastery model therefore cannot claim value from architecture; it claims value from the objective and the substrate. It operates on open-ended artifacts with revision content rather than binary correctness on a fixed skill taxonomy, predicts

trajectory shape and persistence rather than next-item correctness, and is used as a reward inside training rather than as a prediction inside a tutoring system.

Edit-prediction models. We must be precise here, because the naive form of our claim is false. Models are already trained on human revision data. CoEdIT instruction-tunes on tens of thousands of text edits [100]; LintSeq trains code models on synthetic edit sequences [96]; next-edit prediction systems train on interaction history and commit-level edits [74]; and shipped developer tools condition on a user's recent edit history and, in at least one case, apply verifiable-reward fine-tuning where the reward is agreement with the developer's eventual cursor movement. The idea of training on edits is taken. What is not taken is training a *value function over improvement* from human traces and using it as the reward for a policy. Edit-prediction models imitate the next edit; a mastery model evaluates whether an edit was progress toward eventual success, which is a different object with a different training signal and a different use.

World models. A world model identifies the environment's transition kernel [42, 44, 65]. A mastery model identifies the solver's. They are complements: an agent that plans against a world model and is trained against a mastery model is optimizing both halves of the loop of Figure 3.

4.5 The same object, fitted to patients

The closest formalism to a mastery model is not in the learning literature. Deep Kalman Filters [60] and the structured inference networks that followed [61] fit a nonlinear state space model with deep transition and emission distributions over a latent patient state, conditioned on the treatments administered; the line continues through attentive state-space models of disease progression [8] and counterfactual treatment-outcome estimation over time [13]. A latent state, an action-conditioned transition, irregular observations, and a realized outcome at the end: structurally that is Definition 3 with a patient where the solver should be. We state the resemblance plainly rather than distinguishing our way out of it, because three things follow from it and two of them cost us something.

The architecture is not the contribution. A latent-state model with an action-conditioned transition is twenty years old and was never ours. This matters practically as well as in attribution: such models compress everything about the subject into a single time-varying latent process, which is a far smaller hypothesis class than an autoregressive model of the same size, and they are correspondingly harder to fit. Nothing in this paper requires that V_θ be realized as a state space model. The object is the value over histories and the signed difference between consecutive values; recovering a learner-side representation from an existing model and fusing it with a task-side one is an equally admissible implementation, and on current evidence a cheaper one.

The identification problem is the same problem, and the clinical literature has not solved it either. Treatment assignment is confounded with patient state, which is why that literature reaches for marginal structural models, balanced representations and instruments rather than fitting the conditional directly. Section 5.6 inherits the difficulty in a harder form, because a learner selects their own next attempt while a clinician selects the patient's treatment, so the confounder and the actor coincide. We claim no advantage here and treat the clinical machinery as available rather than as answered.

What is different is what the value is for. A progression model predicts, so that a clinician can decide. Its output is consumed by a human at inference time and it is fitted and used within one population. A mastery model's value difference is consumed as reward by a policy during training, and the thesis of this paper is that a value fitted on *human* improvement supplies gradient to a *machine* policy. That cross-substrate claim is Hypothesis 5, it has no counterpart in the clinical literature, and it is the part that can be wrong.

4.6 The three machine precursors

The closest machine antecedents to a mastery model are not in the language-model literature at all. They come from the developmental-learning lineage this program continues, and they establish that both halves of the proposal have already worked in a smaller setting.

A model of one's own error is a usable exploration signal. Haber et al. [43] built an agent with two networks: a world model predicting the consequences of its actions, and a separate explicit *self-model* that learns the error map of that world model. The agent then uses the self-model to adversarially challenge the developing world model, choosing actions where its own predictions are worst. This produced a spectrum of structured behaviors, including ego-motion prediction, object attention, and object gathering, from no extrinsic reward. The self-model is a model of the learner's

own competence used to select the next attempt: the object of this paper, eight years earlier, with the learner's error surface in place of a learned value over eventual success, and no human traces.

Learning progress, not prediction error, is the correct exploration target. Raw prediction error suffers the white-noise problem: an agent maximizing its own surprise is captured by regions that are unpredictable because they are random rather than because they are learnable. Kim et al. [57] solve this with γ-Progress, a curiosity signal defined as the decrease in loss between a γ-decayed older copy of the model and the current one. They show that this signal gives rise to an exploration policy that directs attention to complex *but learnable* dynamics in a balanced manner, and that the resulting controller substantially outperforms Random Network Distillation and Model Disagreement at active world-model learning in an agent-rich 3D environment. The phrase "complex but learnable dynamics" is the learnability frontier of Section 5 named in an earlier vocabulary, and Section 5.7 adopts γ-Progress as the operational form of the exploration term for exactly this reason. The companion result that curious agents in social environments preferentially attend to animate, contingently responding entities [104] is the developmental analogue of preferring learners over noise.

Both precursors ran on self-generated experience in simulation. What they lacked is the substrate: a record of *human* improvement against which the learner-side model could be fitted, rather than the agent's own instantaneous error. That is what this paper proposes to manufacture.

Bootstrapping on self-generated improvement. The nearest antecedent in the language-model literature is the STaR line [133]: a model generates rationales, those that reach the correct answer are retained, and the model is fine-tuned on its own successful traces, iterating. Quiet-STaR [134] generalizes this to unlabeled text by training the model to generate rationales at every token.

The relation to a mastery model is precise and worth stating rather than eliding. STaR trains on a model's own improvement, filtered by outcome. Three things follow. The filter is binary, so a trace is kept or discarded and never scored, which means STaR inherits exactly the zero-gradient limitation of Section 3 at the trace level. The traces are machine-generated, so they contain no abandonment (Section 5) and no human error structure. And the improvement record is consumed as imitation data rather than fitted into a value function, so nothing in the loop can say whether a given revision was progress. A mastery model is the STaR loop with the outcome filter replaced by a learned, signed value over eventual success, and the model's own traces replaced by a manufactured human record. The continuity is real and we claim it; the difference is the object being learned.

Meta-learning and learning-curve extrapolation. Learning to learn [29] adapts parameters quickly from few examples; learning-curve extrapolation predicts final performance from partial training. Both are adjacent prior art for the value head of a mastery model, which predicts a learner's forward curve from an early trajectory. Neither uses that prediction as a reward.

Curriculum and learning progress. Automated curricula [12, 38], unsupervised environment design [51, 121], and intrinsic-motivation systems that maximize learning progress [88, 98, 105] all locate, by search, the band of tasks where learning is fastest. A mastery model proposes to compute that band from a model of the learner rather than discover it by trial, and Section 5 makes that statement formal.

5 A Formal Framework for Mastery Models

5.1 The learner as a partially observed dynamical system

We model a solver working on task x as a partially observed system. Let s_t denote latent competence, never observed directly. At each step the solver produces an attempt a_t and receives feedback o_t. The transition kernel $P(s_{t+1} \mid s_t, a_t, o_t)$ describes how competence changes. The observation kernel is fixed by the task's verifier. The solver's own policy $\mu(a_t \mid h_t)$ is a function of the observed history $h_t = (x, a_{1:t-1}, o_{1:t-1})$.

This is a standard partially observed Markov decision process with one inversion of the usual roles. In a world model the object identified is the environment's transition kernel and the agent acts on it. Here the object identified is the *solver's* kernel and the environment is fixed.

5.2 Mastery models

Because s_t is latent, we define a mastery model in terms of quantities observable in improvement traces.

Definition 4 (Mastery model, formal)**.** A mastery model $\mathcal{M}_\theta$ maps an observed history h_t and a candidate attempt a to

$$V_\theta(h_t) \approx P(y = 1 \mid h_t), \tag{1}$$
$$\pi^\star_\theta(a \mid h_t) \approx P(a_t = a \mid h_t,\, y = 1), \tag{2}$$
$$c_\theta(h_t) \approx P(\text{solver continues} \mid h_t), \tag{3}$$

where y is eventual success on x. We call V_θ the *eventual-success value*, $\pi^\star_\theta$ the *productive-move distribution*, and c_θ the *persistence function*.

The heads are trained on improvement traces by maximum likelihood: V_θ and c_θ against recorded eventual outcome and continuation, $\pi^\star_\theta$ against the recorded next attempts of trajectories that reached criterion. Architecturally, the CodeInsight finding that a recurrent state space model with discrete latents was the strongest trajectory predictor is a direct prior [44, 93], though the reported margin over item-response and contextual baselines is small and comes from a single fixed split without variance estimates, so we treat it as a prior rather than a settled result, and it also fixes the belief representation used in Equation 5 below.

5.3 The mastery advantage

Definition 5 (Mastery advantage)**.** For history h_t and attempt a_t with feedback o_t, let $h_{t+1} = h_t \oplus (a_t, o_t)$. The mastery advantage is

$$A_\theta(h_t, a_t) \;=\; V_\theta(h_{t+1}) - V_\theta(h_t). \tag{4}$$

This is a temporal-difference error over competence rather than over environment reward [115]: a learned prediction error about whether an attempt was progress, not a score for whether the answer was right. It is dense, reference-free, and signed, the last of which an authored rubric cannot express.

5.4 The shaping objection, stated and answered

A reader who knows the shaping literature will object immediately, and the objection is correct as far as it goes.

Proposition 1 (Policy invariance)**.** *A_θ is a potential-based shaping function with potential $\Phi = V_\theta$. By the theorem of Ng et al. [84], adding $\gamma\Phi(h_{t+1}) - \Phi(h_t)$ to a reward leaves the set of optimal policies unchanged. The theorem quantifies over* all *real-valued Φ and imposes no accuracy requirement, so the construction is exact however V_θ is estimated. With $\gamma = 1$ the shaping term telescopes over a trajectory to $V_\theta(h_T) - V_\theta(h_0)$, a function of the endpoints alone, and the guarantee holds exactly when the potential vanishes at termination.*

We accept this and regard it as a feature rather than a defect, for three reasons.

First, policy invariance is a safety property, not a limitation. It guarantees that a mastery model cannot induce a policy that is optimal for the reward but wrong for the task. Every alternative dense reward that is not potential-based carries exactly that risk, which is the failure mode Zhang et al. [135] document for cheaply estimated PRMs. That A_θ is potential-based is why it is safe to use at scale.

Second, the claim of this paper is about learning dynamics, not about optima. The exploration wall is not a statement that the optimum is wrong; it is a statement that the gradient is zero. On a group where every attempt fails, outcome reward supplies no signal and the optimum is unreachable in practice regardless of what it is. Shaping changes which trajectories carry gradient, the variance of the estimator, and the effective horizon of credit assignment. Ng et al. [84] themselves motivate potential-based shaping as the principled way to accelerate learning without distorting the objective, and the entire dynamic-sampling apparatus of Yu et al. [131] exists because zero-gradient groups are the binding practical constraint. A reward that is provably optimum-preserving and demonstrably gradient-restoring is exactly the object worth building.

Third, two of the three terms in the full objective are not potentials. The exploration term of Equation 5 depends on the model's belief state and is not a function of history alone in the required sense; the persistence term of Equation 7 penalizes abandonment and changes the optimal stopping behavior, not merely the path to it. The complete objective is therefore not policy-invariant, and the components that break invariance are the ones whose effect on the optimum we intend.

Fourth, the restored gradient has a precise source, and naming it bounds the claim. Group-relative methods compute the advantage per response. With $\gamma = 1$ the shaping contributes $V_\theta(h_T) - V_\theta(h_0)$ to each response's return, and within a group sharing a prompt $V_\theta(h_0)$ is identical across members and cancels under group normalization. What survives is $V_\theta(h_T)$, and two trajectories that both fail receive different terminal values when one made real progress and the other stalled at the first move. The group advantage is therefore nonzero exactly where the binary outcome is

degenerate, which is the mechanism behind the claim of Section 5.5 that the bracketed sum restores gradient to all-fail and all-succeed groups. The same analysis bounds what the density buys: at $\gamma = 1$ the per-step signal shapes credit assignment *within* a trajectory and contributes nothing to the group-level advantage beyond its endpoint. Recovering intermediate values in the return requires $\gamma < 1$, so the objective is run discounted and the sensitivity to γ is reported. A reader who notices the telescoping and does not find this paragraph is entitled to conclude that the density is decorative.

Remark 1. The practical prediction that follows is sharp and testable: mastery-optimized RL should show its advantage in *sample efficiency and in the fraction of rollouts carrying nonzero gradient*, not necessarily in the asymptotic ceiling at unlimited compute. Kill test 2 is specified at matched compute for this reason.

Remark 2 (The scaling prediction). Remark 1 fixes where the advantage should appear. A second prediction fixes whether it is a law or an engineering gain. If the mastery advantage is a fixed-size improvement it will shrink as base models improve and the frontier will absorb it as a training detail. If it widens with policy scale, the substrate is supplying information that scale does not. Hypothesis 11 tests this at three model sizes, and the sign of that slope is the result the program lives on. We state the prediction before running it: the fraction of rollout groups carrying nonzero gradient falls as a policy improves (Figure 4), so the quantity the mastery advantage restores becomes more valuable, not less, as the base model gets better.

5.5 Mastery-Optimized Reinforcement Learning

The components above compose into a training recipe, and we name it because a named recipe with a stated membership test is checkable in a way that a description is not. It is a composition of parts rather than a new mechanism, which is the same status GRPO, RLVR and DAPO hold, and the composition is what carries the claim.

Definition 6 (Mastery-optimized reinforcement learning)**.** Let $\mathcal{D}_H$ be a corpus of human improvement traces, $\mathcal{M}_\theta = (V_\theta, \pi^\star_\theta, c_\theta)$ a mastery model fitted to it, and $\mathcal{D}_A \subset \mathcal{D}_H$ an anchor split the training loop never sees. *Mastery-optimized reinforcement learning* (MORL) optimizes a policy π_ϕ against Equation 8 with the per-step reward

$$r_t \;=\; \kappa(s)\big[A_\theta(h_t, a_t) + \lambda\, I_\theta(h_t, a_t) + \mu\, R^{\mathrm{p}}_\theta(h_t)\big] \;+\; \beta\, y,$$

subject to periodic re-fitting of V_θ on $\mathcal{D}_A$ and on realized policy outcomes, with drift between predicted and realized success as a stopping condition.

Where it sits. The regimes are distinguished by where the reward comes from, not by the optimizer. RLHF takes it from human preferences over outputs, RLVR from a task verifier, MORL from a learned value over human improvement trajectories. The question each answers is respectively which output is better, whether the output is correct, and whether the attempt was progress.

The drop-in point. In group-relative policy optimization the advantage is computed from a binary outcome $r_i \in \{0, 1\}$ per rollout. MORL replaces r_i with the bracketed sum of Definition 6. Nothing else in the optimizer changes, which is the practical reason the recipe is cheap and the corpus is not.

The last three are not hyperparameters in the usual sense and are fixed before training begins. Tuning them during a run would make the integrity mechanism part of the optimization, which is the failure the mechanism exists to prevent.

The procedure.

1. **Fit** $\mathcal{M}_\theta$ on $\mathcal{D}_H \setminus \mathcal{D}_A$: V_θ under a censored likelihood with a two-parameter logistic ability-difficulty layer; $\pi^\star_\theta$ on reaching trajectories with failure-inclusive weighting; c_θ only on traces with resolved termination.
2. **Reserve** $\mathcal{D}_A$ and the adversarial probe set. Neither enters training.
3. **Roll out** a group of G attempts per task from π_ϕ.
4. **Score** each step with r_t and compute the group-relative advantage on r_t rather than on binary outcome.
5. **Update** ϕ.
6. **Every** T_{anchor} **steps**, re-fit V_θ on $\mathcal{D}_A$ together with recent policy trajectories labelled by realized outcome; measure drift δ and the rank agreement τ between predicted and realized outcomes; score the probe set.
7. **Halt** if $\delta > \delta_{\max}$, $\tau < \tau_{\min}$, or probe scores rose by more than $\epsilon_{\mathrm{probe}}$.
8. **Decay** κ and return to step 3.

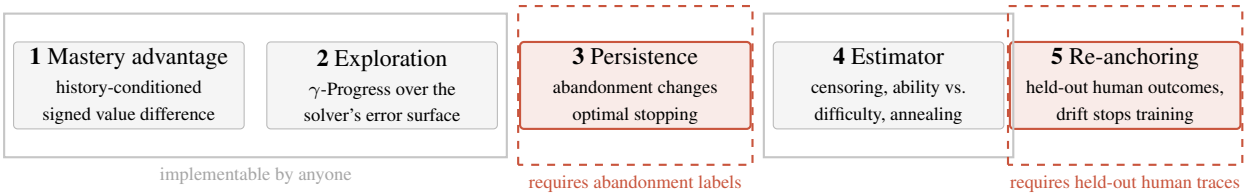
What a partial implementation is. A procedure with steps 1 through 5 and no re-anchoring is a process reward method and should be called one. The distinction is not pedantic: steps 6 and 7 are the only things separating a criterion that terminates in a human record from one that terminates in itself.

Table 3: MORL parameters. The third column states what kind of quantity each is, which matters more than a default value: a derivable parameter follows from a constraint, a swept one is an empirical choice with a stated range, and a preregistered one is fixed before training and never tuned during it.

Symbol	Role	How it is fixed
β	Outcome gate	**Derivable.** $\beta > \frac{1}{1-\gamma} \sup \lvert r_t^{\text{shaped}} \rvert$, so that no trajectory which never arrives can out-earn one that does
$\kappa(s)$	Shaping magnitude	**Derivable terminal condition.** $\kappa(0) = 1$, $\kappa(S) \to 0$, annealing away the nonzero terminal potential and recovering exact policy invariance asymptotically (Section 5.6)
λ	Exploration weight	**Swept.** Selected on the fraction of rollouts landing in the learnability band
μ	Persistence weight	**Swept, bounded above** by the abandonment-label density measured in Workstream 6.2. A head fitted on sparse labels cannot carry a large coefficient
γ_{lag}	Lag of the decayed copy in γ-Progress	**Swept** from the value in Kim et al. [57]
T_{anchor}	Re-anchoring cadence	**Preregistered.** Set so re-anchoring occurs at least ten times per run
$\delta_{\max}$	Drift tolerance	**Preregistered.** Exceeding it halts training
$\tau_{\min}$	Minimum rank agreement between predicted and realized outcomes on $\mathcal{D}_A$	**Preregistered.** Falling below it halts training
ϵ_{probe}	Permitted rise in probe score across judge versions	**Preregistered.** Exceeding it halts training

1. **History-conditioned mastery advantage.** The reward is $A_\theta = V_\theta(h_{t+1}) - V_\theta(h_t)$, where V_θ is conditioned on the solver's trajectory and not on the episode. This distinction is the whole of it: a discriminator over successful and unsuccessful episodes yields a dense signed process reward with published invariance properties [117], and cannot express that the same revision is progress from one history and regression from another.
2. **Learning-progress exploration.** The exploration term of Equation 5 is γ-Progress over the solver's own error surface [57], not novelty in the environment. A solver flailing at random generates high apparent information and no competence, and this is the established fix.
3. **Persistence penalty.** Equation 7 penalizes abandonment and therefore changes optimal stopping behavior rather than the path to it. **Substrate-bound**: it requires traces in which abandonment is labelled, and a policy does not abandon a task, it exhausts a budget, so no corpus of machine rollouts contains the signal.
4. **Identification-aware estimation.** V_θ is fitted under a censored likelihood rather than labelling non-completions as failures; a two-parameter logistic layer separates solver ability from task difficulty; the shaping magnitude is annealed so that the nonzero terminal potential decays and exact policy invariance is recovered asymptotically. Section 5.6 states why each is required.
5. **External re-anchoring with a drift stopping condition.** V_θ is periodically re-fitted on held-out human trajectories with realized outcomes, on a published cadence, with drift between predicted and realized success and their rank agreement as the health metrics and stopping conditions; a standing adversarial probe set is scored on every judge version and rising probe scores halt training. **Substrate-bound**: re-anchoring to an external human record is what distinguishes row five of Table 2 from row one, and it requires held-out human traces that the training loop has never seen.

The membership test. A training procedure is MORL if and only if it has all five components. We state the test because it is the honest form of the claim and because two of the five are not implementable without a human improvement corpus. Component 3 requires abandonment labels; component 5 requires held-out human outcomes. A system with components 1, 2 and 4 is a well-specified process-reward method and should be called one. The recipe is therefore a membership test for possessing the substrate, which is the intended relationship between the method and the data: we publish the recipe precisely because it cannot be run without the corpus, and the corpus is the thing we are building.



A procedure is **MORL** if and only if it has all five.
Components 3 and 5 cannot be run without a human improvement corpus.

Figure 6: The MORL specification as a membership test. Components 1, 2 and 4 are implementable with any corpus, and a dense signed policy-invariant process reward of the type in component 1 is already published with an invariance proof [117]. Components 3 and 5 are substrate-bound: a policy does not abandon a task, it exhausts a budget, so no corpus of machine rollouts carries the persistence signal; and re-anchoring to an external human record is what separates row five of Table 2 from row one. Publishing the recipe therefore costs less than it appears to, because the recipe cannot be run without the corpus. Hypothesis 13 tests whether the two substrate-bound components earn their place.

Evidence that component 5 carries the weight. The nearest published test of this architecture comes from skill evolution rather than weight training. Kang and Wen [52] improve an agent's procedural prompts against a cheap language-model rubric that serves as a dense reward, and periodically re-align that rubric to a small number of real outcome evaluations. Three of their findings bear directly on the specification above. First, removing the re-alignment step collapses performance to within roughly a point of the unimproved baseline, and the authors' diagnosis is that an unaligned rubric supplies confident but wrong guidance; the dense signal alone does little, and the external anchor does the work. That is the prediction of Hypothesis 13 for component 5, observed in a different setting. Second, aligning the rubric's *ranking* of candidates with real outcomes outperformed aligning its absolute scores by three to five points at an identical oracle budget, because absolute outcome values are noisy and non-comparable across tasks of differing difficulty. We adopt this directly: rank agreement τ joins calibration drift δ as a re-anchoring health metric and stopping condition. Third, their gains were large on tasks with verifiable answers and nearly absent on open-ended creative ones, which is the boundary of Section 4.3 reproduced from the other side: a criterion built from a model's opinion does not help where no answer key exists. Their method still requires an oracle to align against, improves prompt text rather than weights, and draws its signal from a language model rather than from human improvement, so it satisfies none of the three conjuncts; it is evidence about the architecture, not a precedent for the object.

What would collapse the distinction. If an ablation shows that removing components 3 and 5 does not degrade transfer, then MORL reduces to a dense process reward with an unusual training set, the substrate is doing no work that a cheaper signal could not do, and we will say so rather than defend the name. Hypothesis 13 is that test.

5.6 Identification

The three heads are stated cleanly above and each carries an estimation problem that the formalism as written does not solve. We name all five, with the treatment we intend to adopt, because a reviewer will find them and because two of them change what can honestly be claimed.

The potential is exact; the terminal condition is not. An approximate value function is commonly assumed to yield an approximate potential, and the assumption is false. The theorem of Ng et al. [84] quantifies over all real-valued potentials: Φ need not approximate $V^\star$, need not be well estimated, and need not be anything in particular. Setting $\Phi = V_\theta$ therefore makes A_θ an *exact* potential difference however badly V_θ is fitted. Estimation error governs how useful the shaping is and not whether it is invariance-preserving, so the safety property of Section 5.4 holds exactly rather than degrading with approximation quality. We state this because the opposite concession is easy to make and costs the argument more than it saves.

The genuine residual sits elsewhere. In the undiscounted episodic case the guarantee requires the potential to vanish at termination, and $V_\theta(h_T)$ does not: a trajectory ending in real progress is scored differently from one that stalled. That nonzero terminal potential is exactly what survives group normalization, which is why it is the source of the restored gradient identified in Section 5.4, and it is the only component of A_θ that is not invariance-preserving. It is therefore the right thing to anneal, and $\kappa(s)$ is retained for that reason rather than as compensation for estimation error. Behboudian et al. [11] remain relevant to finite-horizon optimization under function approximation, where an asymptotic guarantee does not bind on any run of realizable length, and Skalse et al. [112] give the formal account of what shaping preserves

under partial identifiability. Value-approximation error is reported as a first-class training metric because it governs the usefulness of the signal, not its safety.

Traces that do not reach criterion are censored, not negative. V_θ is fitted against recorded eventual outcome. A trace that ends without reaching criterion is right-censored: the solver may have succeeded later, elsewhere, or not at all, and the record does not say. Labelling these $y = 0$ biases the value downward precisely on the hard tasks where the signal matters most. We fit V_θ under a censored likelihood with inverse-probability-of-censoring weighting rather than treating non-completion as failure, and we report the censoring rate alongside every value estimate.

The productive-move head is fitted on survivors. $\pi^\star_\theta$ is trained on the next attempts of trajectories that reached criterion, which is survivorship selection by construction: it learns what worked for solvers conditioned on their having succeeded. The imitation literature is explicit that this induces bias, and the treatment is to include failed and censored trajectories with appropriate weighting rather than discarding them [4]. This is also where the substrate earns its cost, since the failures are exactly what a corpus of finished work does not contain.

The value conflates competence with difficulty. A probability of eventual success is jointly determined by who is attempting and what they are attempting, and the formalism carries a latent competence state with no difficulty term. The same value therefore means both "capable solver, hard task" and "weak solver, easy task," and the mastery advantage inherits the confound. The treatment is standard and we adopt it: a two-parameter logistic layer inside V_θ estimating solver ability and task difficulty jointly, as recent work does for language-model evaluation [2]. Section 10 already stratifies by starting competence for the same reason; the model should do what the evaluation does.

The persistence head has the sparsest signal. c_θ is the most differentiated of the three heads and its training signal is the rarest field in the schema. Distinguishing abandonment from stopping for the day is frequently unrecoverable, and in the capture protocol most cases resolve to unknown. Methods for inferring dropout from interaction logs exist [1], and traces with unresolved termination are excluded from persistence training rather than imputed. We state the risk plainly: if abandonment cannot be labelled at sufficient density, the head that most distinguishes this work from a process reward model is the head that cannot be fitted, and Hypothesis 5 is where that shows up first.

5.7 The exploration term

A revision can fail and still be valuable if it reveals something about the solution space. Optimal experiment design [19, 69] provides the measure. Let b_t denote the mastery model's posterior over its discrete latent state given h_t. Define

$$I_\theta(h_t, a_t) \;=\; \mathcal{H}[b_t] - \mathbb{E}_{o_t}\big[\mathcal{H}[b_{t+1}]\big], \tag{5}$$

the expected information gain of attempt a_t [107]. Because the architecture of Section 5 carries a discrete latent, $\mathcal{H}$ is the entropy of a categorical posterior and is computed exactly; where a continuous-latent or ensemble architecture is used, we substitute ensemble disagreement of V_θ as the uncertainty measure. An attempt that fails but collapses uncertainty about where the failure lives has positive I_θ and non-positive A_θ. Both signals are present in recorded human traces: the student of Section 1 who read the failing test at attempt 6 gained information without gaining a passing test.

The white-noise problem and its fix. Equation 5 as written is vulnerable to a known failure: an agent that maximizes its own uncertainty reduction can be captured by regions that are unpredictable because they are random rather than because they are learnable. In the improvement-trace setting the trap is concrete, since a solver flailing at random generates high apparent information and no competence. Kim et al. [57] give the established remedy, γ-Progress: define the signal as the decrease in loss between a γ-decayed older copy of the model and the current one, so that only *reducible* uncertainty is rewarded. We adopt this as the operational form,

$$I^\gamma_\theta(h_t, a_t) \;=\; \ell_{\theta^-}(h_{t+1}) - \ell_\theta(h_{t+1}), \tag{6}$$

where ℓ is the mastery model's predictive loss on the realized next state and θ^- is an exponentially lagged copy of θ. Equation 5 is the idealized quantity; Equation 6 is what is computed. The substitution is not cosmetic: it is the difference between rewarding a policy for producing confusion and rewarding it for producing learnable confusion, and it has already been shown to outperform Random Network Distillation and Model Disagreement on active world-model learning [57].

5.8 The persistence term

The persistence function enters the objective rather than sitting beside it. Let $\kappa_t \in \{0, 1\}$ indicate that the policy abandons the task at step t. Define

$$R^{\mathrm{p}}_\theta(h_t) \;=\; -\,\kappa_t \cdot c_\theta(h_t), \tag{7}$$

which charges an abandonment exactly the probability that a human solver in the same state would have continued. Quitting where humans quit is free; quitting where humans persist is penalized. This term is estimable only from human traces, because synthetic revision traces contain no quitting.

5.9 The mastery objective

The policy under training, π_ϕ, is optimized on

$$J(\phi) \;=\; \mathbb{E}_{\tau\sim\pi_\phi}\Big[\sum_t \gamma^t\big(A_\theta(h_t, a_t) + \lambda\, I_\theta(h_t, a_t) + \mu\, R^{\mathrm{p}}_\theta(h_t)\big) \;+\; \beta\, y\Big], \tag{8}$$

where βy is an *outcome gate*: final verifiable success still contributes, so a trajectory that never arrives cannot be rewarded for looking like progress. In the group-relative setting the bracketed sum replaces the binary outcome inside the group advantage, which restores nonzero gradient to groups in which every attempt fails and every attempt succeeds.

5.10 Covariate shift is the central technical risk

Assumption 1 (Bounded history shift)**.** V_θ is trained on human histories drawn from μ and evaluated on policy histories drawn from π_ϕ. The framework requires that V_θ remain calibrated under π_ϕ.

This assumption is violated from the first gradient step, and it is a more fundamental risk than reward hacking. Section 6 argues that the *structure* of productive revision is substrate-invariant even though error distributions differ; that argument, if correct, bounds the shift but does not eliminate it. The situation is formally the one that motivates interactive imitation learning, where a policy trained on expert states drifts into states the expert never visited and the remedy is to query the expert on the policy's own distribution [55, 103]. We adopt the same remedy in reward-model form: V_θ is periodically re-fit on trajectories drawn from the current policy and labeled by their realized outcomes, so the value head is repeatedly re-anchored to the distribution it is scoring. Drift between V_θ and realized success on policy trajectories is the primary health metric of training, reported on every cycle. Kill test 10 makes it a stopping condition.

5.11 The learnability frontier

The mastery model also locates where a solver should work. Define the learnability of task x for a solver with history h as the expected information about competence produced by an attempt drawn from the *solver's own* policy,

$$L_\theta(x \mid h) \;=\; \mathbb{E}_{a\sim\pi_\phi(\cdot|h,x)}\big[I_\theta(h, a)\big]. \tag{9}$$

L_θ is near zero for saturated tasks and for tasks far beyond reach, and peaks in the band the learning sciences call the zone of proximal development [120] and the RL curriculum literature discovers empirically [38, 88, 121]. That such a band exists and is exploitable is not a conjecture: Kim et al. [57] demonstrate a controller that finds it, described in their terms as directing attention to complex but learnable dynamics, and show it beats standard exploration baselines. What that work computes from an agent's own prediction error, Equation 9 computes from a model fitted to human improvement. Curriculum becomes a control policy: select x to maximize $L_\theta(x \mid h)$. In this formulation the exploration wall is the statement that $L_\theta \approx 0$ everywhere in the environment pool, and the remedy is to move the pool, not the reward. Figure 7 shows the shape and its relation to the zero-gradient regions of group-relative RL.

5.12 Relation to existing objects

Table 4 summarizes the reductions. A PRM is recovered by replacing V_θ with a fixed correctness function and dropping $\pi^\star_\theta$ and c_θ. Knowledge tracing is recovered by restricting attempts to binary correctness on items from a fixed taxonomy and predicting next-item correctness. An edit-prediction model is recovered by keeping $\pi^\star_\theta$ alone and conditioning it on all trajectories rather than successful ones.

6 Why a Model of Human Improvement Should Transfer to a Machine

The central assumption is that a mastery model trained on human improvement traces provides a useful reward for a non-human learner. This deserves an argument.

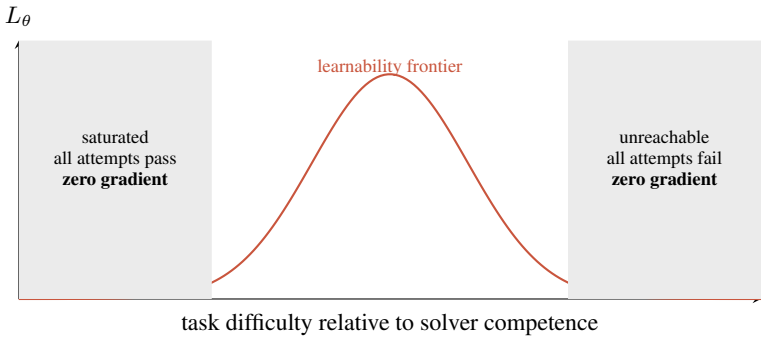


Figure 7: Learnability as a function of task difficulty relative to solver competence (Equation 9). The shaded regions are where group-relative RL produces exactly zero gradient [109, 131]. As a policy improves, mass in the environment pool migrates outward into the shaded regions; dynamic sampling detects this after spending the rollouts, whereas a mastery model predicts it beforehand.

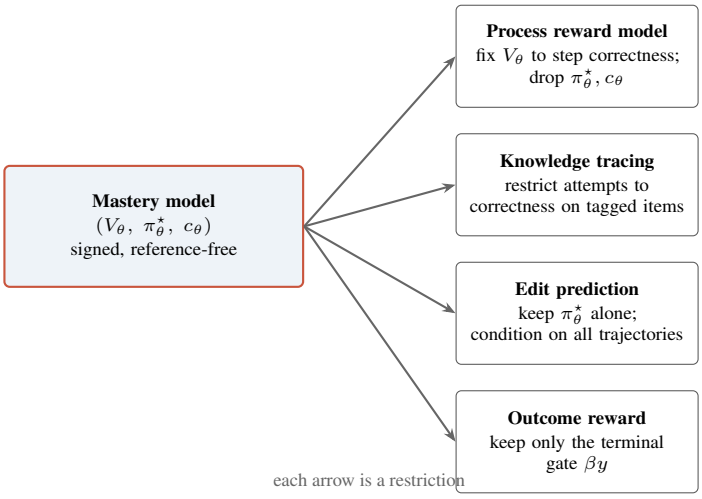


Figure 8: Four existing objects recovered as restrictions of a mastery model. Removing the persistence and productive-move heads and fixing the value to step correctness gives a process reward model; restricting attempts to binary correctness on a fixed skill taxonomy gives knowledge tracing; keeping only the productive-move head and conditioning it on all trajectories rather than successful ones gives an edit-prediction model; keeping only the terminal gate gives outcome-verified RL. The generalization adds sign, removes the reference, and adds the decision to continue.

The claim is not that human and model error distributions match. They do not, and Ross et al. [102] measure the mismatch directly. The claim is that the *structure of productive revision* is substrate-invariant: diagnose before patching, localize the failure, read the feedback before changing the output, reduce the problem to a smaller one, abandon an approach that has stopped yielding information. These are strategies, not errors, and the space of ways to make progress on a task is far smaller than the space of ways to fail at it.

Three supports, in decreasing strength.

1. **Process data measurably outperforms outcome-only data in transfer settings.** Removing reasoning traces from teacher trajectories while holding the final actions fixed dropped downstream WebArena success from 41.5% to 33.6% [75], a 7.9-point loss attributable to the process record alone. On the human side, Ross et al. [102] find that models trained on real student traces outperform models trained on final programs *and* models trained on synthetic traces at reproducing learner behavior. Real process beats both no process and fake process.
2. **The transfer mechanism is a reward, not imitation.** The policy discovers its own solution paths; only the judgment of whether a revision was progress is human-derived. A judge trained on human traces does not need the policy to err like a human. It needs to recognize progress, and progress is the narrower object.
3. **The embodied channel closes the loop empirically.** Robot learning produces improvement traces in which the learner is already a machine and the corrector is a human: interactive imitation with expert interventions [55, 103] and large teleoperation corpora [87]. A mastery model trained on that channel is a direct test of whether human-derived judgment of progress applies to a non-human learner.

Table 4: Mastery models in relation to the nearest existing formalisms. Each is recovered as a restriction.

	Process reward model	**Knowledge tracing**	**Edit prediction**	**Procedural structure**	**Mastery model**
Observation	A reasoning step	Correctness on a tagged item	The next edit	Successful vs. failed trajectories	The revision and whether it was progress
Reference needed	Yes	No, but a taxonomy is	No	No, but a schema is	No
Predicts	Step correctness	Next-item correctness	Next edit	What to do next, given position	Eventual success, productive move, persistence
Sign	Non-negative	Non-negative	n/a	n/a	Signed
Used as	Reward or reranker	Prediction in tutoring	Policy (imitation)	Guidance at decision time	Reward in training
Nearest work	[68, 119, 127]	[21, 56, 95]	[74, 96, 100]	[76]	This paper

The precedent often cited here, that pretraining already transferred human knowledge to models whose internals resemble nothing human, is an analogy and we label it as one. Substrate-invariance is a conjecture. It is tested twice in the first year: by transfer of the mastery objective off code (kill test 1) and by the human-versus-synthetic ablation (kill test 5). It is also the assumption whose failure would be most informative, because it would locate precisely where machine and human learning diverge.

7 Six Sciences of Learning

Each science below supplies a mechanism the others lack, and each mechanism is tied to a specific element of the framework.

7.1 Mathematics and control

Section 5 is system identification applied to a learner. The mastery model is the identified plant; curriculum is a control policy over it; the learnability frontier is where optimal experiment design says to sample [19, 69, 107].

7.2 The law of practice, corrected

The learning-curve literature offers the oldest quantitative regularity available as a consistency check, and it is commonly cited in a form that is wrong. Newell and Rosenbloom [83] established the power law of practice on aggregate data. Heathcote et al. [46] fitted 40 data sets comprising 7,910 individual learning series from 475 subjects across 24 experiments and found that an exponential form fitted better than a power form in *every* unaveraged data set, and that averaging over subjects produces a bias in favor of the power law. Individuals learn exponentially; power laws are an artifact of aggregation over heterogeneous rates.

This correction strengthens rather than weakens the program, because it yields a two-level consistency check that a purely aggregate law could not provide.

1. **Individual level.** V_θ, conditioned on a single held-out learner, must generate trajectories whose time-to-criterion is fitted better by an exponential (or APEX) form than by a power form.
2. **Population level.** Pooling those same generated trajectories across learners must reproduce an aggregate power law, and the fitted exponent must follow from the distribution of individual rates rather than being imposed.

A model that reproduces both is representing learner heterogeneity, not memorizing an average curve. This is also the theoretical basis for the learner-diversity term in the substrate scaling law of Section 8: if the aggregate law is generated by heterogeneity of rates, then the number of distinct learners in the substrate is a first-order quantity and not a proxy for label count. Failure of either level is a kill condition (kill test 8). The three-stage structure of skill acquisition [10, 30] supplies a second, weaker check on $\pi^\star_\theta$: the productive-move distribution should shift from search-like to execution-like along a trajectory.

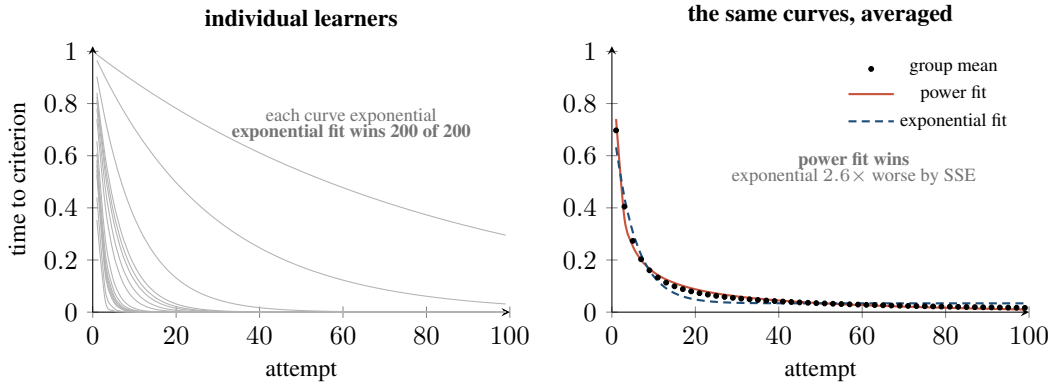


Figure 9: The aggregation artifact, computed. Two hundred simulated learners improve exponentially with gamma-distributed rates, for which the population mean is exactly $(1 + \theta t)^{-k}$, a power law. Left: an exponential form fits better than a power form for every one of the 200 individuals. Right: averaging the identical curves produces a mean that a power form fits better, with the exponential $2.6\times$ worse by sum of squared error. This reproduces the finding of Heathcote et al. [46] across 7,910 real learning series and is why Hypothesis 8 requires a mastery model to satisfy both levels: exponential per learner, power law on pooling, with the aggregate exponent following from the fitted distribution of individual rates rather than being imposed.

7.3 Learning science

Mastery learning [16, 17]: competence is reached by iterating to criterion, and the tutoring gap that Bloom's two-sigma problem names is a gap in per-learner adaptation, which is what L_θ computes. *Zone of proximal development* [120]: the learnability frontier in psychological terms. *Productive failure* [54]: problem-solving before instruction beats instruction alone, meta-analyzed across 53 studies, 166 comparisons and more than 12,000 participants at Hedges's $g = 0.36$ (95% CI 0.20 to 0.51), rising to between 0.37 and 0.58 under high design fidelity [111], with the effect reversing for young learners and domain-general skills. This is the human analogue of the explore-then-instruct schedule that Equation 8 implements, and it is why failed attempts in a trace are signal. *Desirable difficulties* [15] and feedback [45] give the same conclusion from other directions. *Effortful practice* [26] is relied on only in its weak form: Macnamara et al. [77] found deliberate practice explained 26% of performance variance in games, 21% in music, 18% in sports, 4% in education and less than 1% in professions, which refutes the strong claim while leaving intact the claim the objective uses, that feedback-rich effort at the edge of competence is necessary.

7.4 Neuroscience

Dopaminergic signaling encodes reward *prediction error* rather than reward magnitude [106]; the mastery advantage of Equation 4 is the same object computed over competence. We state the claim precisely: prediction error is a process signal derived from outcomes, so this is not evidence that brains ignore outcomes, but it is evidence that the learning signal is a difference rather than a level. Cortical error-related negativity is a measurable response to one's own mistakes with an explicit reinforcement-learning account [36, 48]. Hippocampal replay re-runs trajectories, not endpoints [31], which motivates the use of improvement traces as an off-policy replay buffer. Curiosity states enhance consolidation of what follows [39], motivating the exploration term. The computational form of that motivation is not new either: an agent that models its own error surface and acts to challenge it produces structured, developmentally ordered behavior from no extrinsic reward at all [43].

7.5 Behavioral science

When people persist and when they quit is the framing question of CodeInsight [93] and the object of Equation 7. It has an experimental literature: whether a learner interprets difficulty as a signal of importance or of impossibility determines whether the next attempt happens at all [89]. This is where the human substrate is irreplaceable. A synthetic trace generator instructed to produce realistic revisions will produce revisions; it will not produce the distribution of abandonment, because abandonment is not a property of the task.

7.6 Sociology and the civilizational record

Section 2 stated the observation; its consequence is that the substrate was not merely never collected but structurally discarded by the institutions that produced it [62, 64, 97, 101, 116].

7.7 Open-ended and continual learning

Open-endedness requires an environment that keeps generating tasks at the edge of the agent's ability, and fixed task sets saturate [20, 49, 113, 121]. Catastrophic forgetting [58] is a trajectory phenomenon no snapshot benchmark detects. Intrinsic-motivation and learning-progress systems [88, 94, 105] supply the machine-side precedent for Equation 5.

7.8 Social and engineering systems

Generative agents have memory, reflection, and planning [90] and no development module. A mastery model is the fourth capability, and it is what would make a population of simulated agents a population of learners [59, 91]. The general user model lineage [108] already infers that a user is struggling by observing stalled edits: a trace observation used to infer a state. We train on the trace to model the dynamics. Lee et al. [66] established the evaluation principle four years earlier, finding that better non-interactive performance does not reliably produce better interaction; this paper makes that a training principle.

What the six sciences say together. Every science of learning treats trajectory and error as the object of study and the finished artifact as a byproduct. The frontier is the only learning system in the room that does the reverse.

8 Manufacturing the Substrate

Improvement traces cannot be scraped at the required fidelity. They must be manufactured. We specify four channels, a consent architecture, and a scaling-law measurement plan.

8.1 Four channels

1. **Instrumented environments.** The CodeInsight capture model [93], generalized to writing, proof repair, debugging, design iteration, and scientific protocol revision. The unit of capture is every intermediate state, not only submissions. Ross et al. [102] demonstrate the same capture at a different scale, 3.8 million traces over nine years.
2. **Commissioned solve-while-recording.** Domain experts work with full revision capture, failures preserved. The operational precedent is benchmark construction in a high-stakes clinical domain: Patel et al. [92] built the first unified benchmark across five speech-language pathology tasks by developing a taxonomy of real-world use cases with practicing clinicians, in a field where roughly 3.4 million affected children face a workforce outnumbered nearly twenty to one. The lesson transferred here is that expert-elicited process data in a constrained domain is manufacturable, and that the binding cost is expert time and taxonomy design rather than volume. The economics favor this channel: the compute a training task consumes over its lifetime, estimated at roughly $2,400 by one vendor's public model, dwarfs the cost of creating it, so trace quality dominates trace count.
3. **Reconstruction from version histories.** Recovery of as-of-step intermediate states from commit and edit histories under strict future-information discipline. This is the channel that overlaps most with existing edit-model training data [74, 96, 100] and is correspondingly the least differentiated.
4. **Embodied correction traces.** Interactive imitation with expert interventions [55, 103] and teleoperation corpora [87]: failed grasps, human takeovers, sim-to-real iteration. The learner is a machine and the corrector is a human, which makes this channel the direct test of Section 6.

The adjacent industry is manufacturing the other half, and publishes its costs. A procedural substrate is being built at scale, and it is worth stating precisely what is in it. Ge et al. [35] convert human-written agent skills into 2,756 executable environments with code-based verifiers across twelve categories, sample 8,364 verified trajectories averaging 49 tool calls each, and fine-tune a 35B policy that gains 199 Elo on professional deliverables and 19.1 points on terminal tasks. The substrate is human and procedural, and it is not an improvement trace. A skill document is procedural knowledge in finished form, written once its author is already competent, and the paper concedes the consequence structurally: skill documents "describe reusable procedures rather than concrete problems", so the tasks had to be manufactured from them and the attempts generated by models. The record of anyone becoming able to write the skill is not in the corpus, because it was never in the document.

Three of their measurements transfer directly to the plan above. Environment construction cost 4.5 hours of agent time per accepted task, which prices the manufacture of a verifiable procedural environment independently of this program's own estimates. Across 48,152 sampled trials only 8,364 passed, a 17.4% success rate, and the failures were retained to support analysis of unsuccessful execution rather than fitted as signal, which is Section 1's omission performed once more on a corpus built in 2026. And only 36.8% of successful trajectories invoked their target skill at all, which is their

own evidence that the verified environments and trajectories carry much of the gain rather than the human documents that seeded them.

Their strongest result also sharpens the boundary this paper draws. The training signal is binary and verifier-derived throughout, yet the 199-Elo gain is measured on blind pairwise comparison of professional deliverables, a benchmark with no executable answer key. Competence acquired under verification transferred to a domain scored by preference, which is a useful and underappreciated result. It bounds the claim of Section 4.3 in exactly one direction and leaves the operative one intact: every gradient in that work came from a code-based checker, so what is demonstrated is transfer of competence already acquired, and what remains open is reach into domains where no gradient is available in the first place. That distinction is the one this program is built on.

The same week, the manufacture of the environments themselves was automated and priced. Noronha et al. [85] invert the usual generation order. Rather than writing a task and then searching for an answer, they specify the valid solution space, the environment requirements and the verification criteria as a *blueprint* first, and materialize the executable environment from it, which makes solvability a construction constraint rather than a property checked afterward. The result is a six-domain suite of 600 tasks produced in under ten hours at roughly $100 to $200 per fifty-task batch, with 86% of tasks surviving repair. Two further measurements matter here. Difficulty is steerable at fixed domain and tool interface: moving the generation parameters from the low to the mid-to-hard range shifts the fraction of tasks a frontier model finds hard from 10.0% to 39.0%. And the curriculum is closed-loop, adapting the sampling distribution from the policy's own observed failures, relaxing dimensions below 0.30 success and strengthening those above 0.80.

This is the strongest form the rival position takes, and it is worth stating at full strength before qualifying it. Their verifier is not only an outcome check. It carries a *process quality* signal that scores the trajectory independently: whether the agent applied the required filters simultaneously, looked up a threshold rather than assuming one, selected the correct policy type. That is a step-level reward, built at a few dollars per task, with no human improvement corpus anywhere in the pipeline. If a dense process signal can be manufactured that cheaply, the substrate argument of Section 8 is weaker than it appears.

The qualification is structural rather than a matter of degree, and the paper's own limitations section states half of it. Their process checks are derived from a single solution path fixed at construction time, so they score conformance to one prescribed route. A learner that reaches the answer by a different valid route scores worse on process while scoring the same on outcome. That is prescription, not valuation: it answers whether this step matches the intended step, and not the question a mastery model is fitted to answer, which is how much this state improved the probability of eventual success along whatever route the learner is actually on. The distinction is the one Section 4.3 draws between replaying a recorded path and learning a value over states. Their curriculum has the same shape. It adapts to the policy's own failures, which places it in row one of Table 2 rather than row five: the system that sets the difficulty and the system being measured are the same system, and nothing external re-anchors the judgment. Their stated limitation closes the remaining gap themselves, that open-ended tasks still fall back partly on LLM adjudication and eliminating it for free-form work remains open, which is the class of task where no prescribed solution path exists to check conformance against.

So the reading is not that the environment layer is unsolved. It is being solved, quickly and cheaply, and this program should assume verifiable environments are a commodity within the horizon of its own experiments. What is not being manufactured anywhere in this literature is a record of a competent human getting better at something, and the value function that can be fitted to it.

8.2 Custody, not willingness, is the binding constraint on acquisition

The four channels above assume a holder willing to be recorded. That assumption quietly selects on the value of the process to the person who holds it, and it selects the wrong way. A learner's trace has little private value to the learner, so it is cheap to obtain and correspondingly thin. The traces worth the most are held by people for whom the process *is* the asset: the specialist whose sequence of diagnostic moves is the reason they are hired. For them the recording is not a byproduct of a service they wanted; it is disclosure of the thing they sell.

This is the buyer's inspection paradox, and it is well posed. Rahaman et al. [99] state it for information goods generally: the buyer must inspect to determine value, and the seller must restrict inspection to prevent expropriation, so the market clears below the efficient point or not at all. For improvement traces the problem is sharper than for a document, because a trace is not one good but a continuous record of how its holder works. A flat price per trace makes it worse rather than better, since only holders whose traces are worth less than the price will sell, which is adverse selection in its original form: the market unravels from the top.

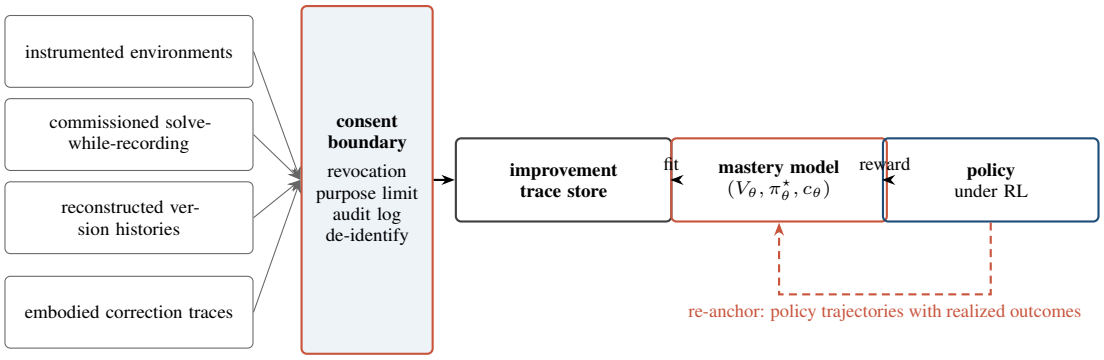


Figure 10: The substrate pipeline. Four manufacturing channels feed a consent boundary that no identifiable trace crosses; the trace store fits the mastery model; the mastery model rewards the policy. The dashed return path is the re-anchoring loop of Section 5.10, which re-fits V_θ on the policy's own trajectories and is the single mechanism addressing both covariate shift and reward hacking.

The architecture admits a partial escape that most data markets do not, and it follows from what a mastery model actually consumes. V_θ is fitted to the *statistics* of improvement, not to the content of any trace. The trained value function is a lossy, non-invertible compression of the trajectories that produced it, while the holder's private advantage lives in the content: the particular diagnostic move at the particular impasse. Those are separable in principle, which is not true of a corpus sold for pretraining. The operational form is contribution without disclosure: the trace never leaves the holder's environment, what leaves is a model update under secure aggregation, and payment is on measured marginal contribution rather than per trace. Paying on measured contribution also inverts the selection, because the holder whose process is genuinely better now earns more rather than being pooled with the median.

A mechanism exists on paper, and its blocking failure is published. Stephenson et al. [114] give the hold-up problem exactly this treatment. Delegating the disclosure and payment decisions to tamper-proof programs inside a trusted execution environment makes full disclosure incentive-compatible for the seller, with an efficient *ex post* transfer, because information revealed in a negotiation that does not close is destroyed rather than retained. We decline to present this as a solved component, because the work that tests it finds the opposite. Leku and Park [67] ask whether an agent can tell a secure execution environment from an insecure one on the evidence reaching its context. Across ten models, failed attestations suppressed disclosure reliably, by 40% to 99%, while *valid* attestations produced incoherent behavior, with some models disclosing no more than before and others disclosing less despite correct cryptographic proof. Agents detect danger and cannot verify safety, which is the wrong half of the capability the protocol requires. The general case is no better: Mireshghallah et al. [79] find that instruction-tuned models leak information that the context forbids sharing in 39% of cases for the strongest model tested, and that chain-of-thought prompting does not reduce the rate and sometimes raises it.

What this means for the program. The custody layer is a dependency this program carries rather than a feature it defers. Without it, acquisition is confined to domains where process has no private value, which is precisely the regime in which the resulting corpus is least likely to contain frontier-relevant competence. We therefore state it as a kill condition rather than a workstream: if contribution without disclosure cannot be made verifiable to a holder within the first year, the substrate thesis is restricted to the low-private-value regime and the ceiling of the whole program drops with it. Hypothesis 16 is that test.

8.3 Consent architecture

The substrate is a record of people learning, and every trace is a person's failures. Consent is a founding-level design decision. Four clauses: per-learner consent with revocation at any time, with deletion propagated to derived models on the next training cycle through machine unlearning; purpose limitation binding traces to the training of mastery models and never to evaluation of the individuals who produced them; an audit log recording every use of every trace, following the pattern proposed for generative agent banks by the group that built them [91]; and no trace leaving the manufacturing boundary in identifiable form. Consented longitudinal cohorts at national scale are the operational template. Academic datasets collected under institutional review, including CodeInsight, are not transferred into the program; the pipeline design is.

8.4 The scaling law of the substrate

The reference point is PRM800K [68]: roughly 800,000 step-level labels trained the strongest process reward model of its generation. CodeInsight holds over three million transitions in one domain, so a first mastery model starts at known scale. What is unknown is the scaling law of the substrate itself. Following Kaplan et al. [53] and Hoffmann et al. [47], we measure performance against transition count N_τ, distinct-learner count N_ℓ, and domain count N_d, and fit

$$\mathcal{L}(N_\tau, N_\ell, N_d) \;\approx\; \mathcal{L}_\infty + \alpha_\tau N_\tau^{-\eta_\tau} + \alpha_\ell N_\ell^{-\eta_\ell} + \alpha_d N_d^{-\eta_d}. \tag{10}$$

The exponent η_d is decisive: if it is zero the substrate does not transfer across domains and the program knows it early (kill test 9). The exponent η_ℓ tests the learner-diversity claim, and Section 7.2 gives it a theoretical basis: if aggregate power-law learning curves are generated by heterogeneity of individual exponential rates, then distinct learners carry information that additional transitions from the same learners cannot supply, and η_ℓ should be strictly positive and not reducible to N_τ. We are not aware of a published scaling law that separates labeler diversity from label count for reward models, so we state this as a measurement, not a result.

8.5 Cost model

The target is a cost per productive trace at or below the going rate for RL-environment tasks, publicly reported in the range of $200 to $2,000 per task with complex environment replicas an order of magnitude higher. A trace is productive if it preserves intermediate error states and terminates in a recorded outcome. The kill condition is a cost exceeding ten times environment-task pricing with no identified path down.

9 Reward Hacking and Its Defenses

Every learned reward model is exploited by the policy it trains [9, 34], and process reward models are a documented case [135]. The problem is not only exploitation from outside. Fein et al. [28] audit five reward models including a state-of-the-art one and find biases of length, sycophancy and overconfidence still present, plus two further biases in answer order and model-specific writing style, and they separate failures that a linear post-hoc correction removes from failures resistant to linear intervention. A mastery model is a learned reward model and inherits this. A policy optimized on Equation 8 will find revisions that look like progress to $\mathcal{M}_\theta$ and are not. Three defenses are built in from the first run, and one of them does double duty against covariate shift.

Re-anchoring. $\mathcal{M}_\theta$ is periodically re-fit on trajectories drawn from the current policy and labeled by realized outcomes, in the manner of interactive imitation [103]. This addresses hacking and the distribution shift of Section 5.10 simultaneously, and drift between V_θ and realized success is the primary health metric of training.

The failure mode is documented, and the remedy is converging. This is not a hypothetical risk. Duan et al. [25] report that Anthropic's automated research experiments produced "random-seed cherry-picking and attempted test-label extraction through evaluator queries," and that evolving evaluators further complicate comparison across rounds. Their stated requirement is that "protected evaluation and matched computational budgets are needed to separate genuine improvement from evaluator exploitation or increased search effort," and the mechanism they cite as addressing it, the Red Queen Gödel Machine, freezes evaluators within each epoch and validates replacements against an independent ground-truth anchor. That is the re-anchoring architecture of this section and the matched-compute discipline of Section 11, arrived at independently by a different community from a different starting point. We take the convergence as evidence that the design is right and not as evidence that it is sufficient.

The failure mode has a name in the nearest domain. A reward defined over learning gain invites a specific pathology, and it has been measured. ped [5] show an engagement-optimised tutoring policy over-selecting a high-engagement action that produces zero mastery gain, and propose a severity index for the general case. That is this paper's reward, hacked, in the domain closest to its substrate. We adopt their index as a standing monitor alongside the probe set below, and we treat the result as the most specific warning available rather than as a distant analogy.

Separation is converging independently. Four groups have now arrived at the same structural requirement from different directions. Duan et al. [25] require protected evaluation and matched budgets. Liu et al. [70] fix capability and efficiency metrics before search and hold them strictly outside the optimising agent's control to prevent it from gaming the acceptance criteria, evaluating held-out evidence only after a candidate is frozen and never feeding it back. Lu et al. [76] commit structural edits only where held-out validation performance is preserved or improved. Yan et al. [129] separate implementation-time testing from independent evaluation. None of these is a reward model trained on human

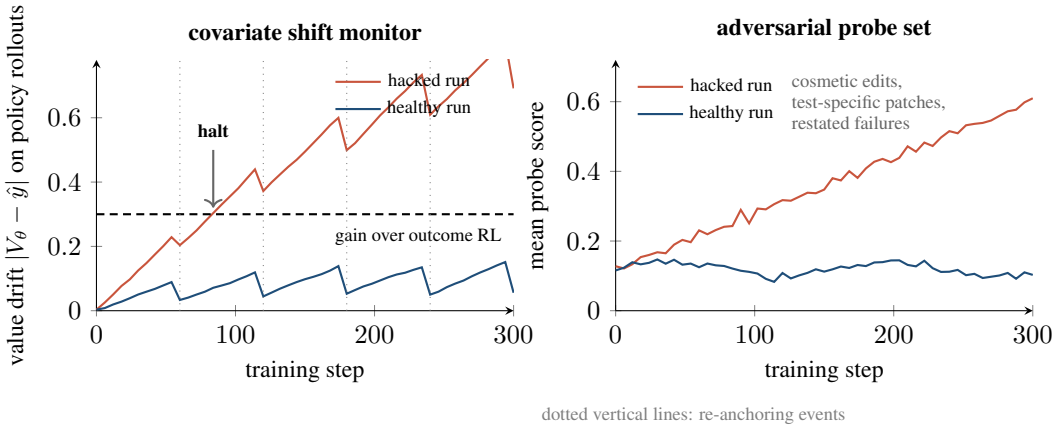


Figure 11: The two instruments of Section 9, shown on an illustrative healthy and hacked run. Left: drift between V_θ and realized success on the policy's own trajectories, with re-anchoring events marked; the stopping rule fires when drift exceeds the gain of mastery-optimized RL over outcome RL, which is kill test 10. Right: mean score assigned to revisions constructed to resemble progress without being progress; a monotone rise halts training. In the healthy run re-anchoring returns drift to baseline at every cycle, and probe scores are flat. *Illustrative traces showing the instrument and its thresholds, not measurements.*

traces, and the convergence is on the architecture rather than on the signal, which is why we treat it as evidence that the design is right and not that it is sufficient.

Adversarial trace probes. A standing set of revisions constructed to look like progress without being progress, including cosmetic edits, test-specific patches, restated failures, and verbose non-changes, is scored on every version of $\mathcal{M}_\theta$. Rising probe scores halt training. The construction of this set is informed by the Best-of-N evaluation bias Zhang et al. [135] identify, in which correct answers arrive through flawed processes.

Outcome gating and policy invariance. The βy term of Equation 8 ensures verifiable success still gates the episode, and the shaping analysis of Section 5.4 bounds the worst case: the potential-based component cannot by itself induce a policy that is optimal for the reward and wrong for the task. The components that break invariance, I_θ and R^{p}_θ, are exactly the ones that require the probe set.

10 Mastery Evaluations

Frontier benchmarks measure the artifact. Mastery evaluations measure the learner. Five quantities, each computed from a trajectory rather than a snapshot.

1. **Improvement rate.** Slope of success probability against attempt count, per task, averaged over tasks; the empirical derivative the law of practice describes, fitted at both levels of Section 7.2.
2. **Retention.** Success probability at a fixed delay after criterion was first reached, without intervening practice; the trajectory-level detector of catastrophic forgetting [58].
3. **Transfer of improvement.** Improvement rate on x' after training on x, minus improvement rate on x' without it, as a function of task distance.
4. **Persistence under difficulty.** Probability of continuing after k consecutive failures, compared against c_θ on matched human histories.
5. **Long-horizon degradation.** Change in the above as interaction history lengthens; the axis on which Laban et al. [63] measured a 39% drop and Moore et al. [81] measured behavioral failure.

A conformance suite for the specification. Alongside the evaluations we publish a conformance suite for Definition 6: tests any training procedure can run to establish which of the five components it implements, including whether its criterion is re-anchored to data the loop has never seen. A specification others can test themselves against is a standard; one they can only read about is a label, and the difference decides whether the term is used by anyone but us.

Validity criterion. An evaluation of a learner is valid if its scores predict which human learners actually improve on held-out outcomes. Whether frontier models score poorly is a finding the evaluation reports, not a target it is built to hit;

an evaluation designed to make frontier models fail would be circular. The kill condition is predictive validity no better than a snapshot benchmark (kill test 6).

11 The Research Program

We state the program as preregistered hypotheses, each with a test and a resolution date. Four are lethal and one is near-lethal.

Hypothesis 1 (Transfer off code)**.** A mastery model trained on code improvement traces, used as the reward in Equation 8, improves open-ended performance in a non-code domain relative to outcome-GRPO at matched compute. *Lethal if false.*

Hypothesis 2 (Outcome, authored-rubric, and learned-rubric RL are insufficient)**.** At matched compute on an open-ended benchmark, mastery-optimized RL exceeds outcome-only RL, ExpRL-style authored-rubric process RL, and a learned-rubric reward implemented as a CoT-GenRM-style generative judge [78] scoring against rubrics produced by the Rubrics-as-Rewards generation method [41], on the evaluations of Section 10, and increases the fraction of rollout groups carrying nonzero gradient. Beating the learned rubric is the result that separates this program from what the frontier and its data suppliers can do without a longitudinal substrate. *Lethal if false.*

Hypothesis 3 (Substrate economics)**.** Productive traces can be manufactured at or below $200 to $2,000 each with intermediate states preserved.

Hypothesis 4 (No in-housing)**.** No frontier laboratory publishes or ships a reward model or value head trained on human revision traces within twelve months.

The hypothesis names frontier laboratories, and that is probably the wrong set to watch. A funded team that has already built reinforcement learning against realized outcomes for single decisions [118] is a shorter distance from a value over trajectories than a frontier laboratory is from a longitudinal human corpus. We score the hypothesis as stated and record here that we expect any falsification to arrive from outside the set it names.

Hypothesis 5 (Human traces beat machine traces)**.** A mastery model trained on human improvement traces outperforms one trained on machine-generated improvement traces at predicting eventual success and persistence on held-out human trajectories, and the gap is larger on the persistence head than on the value head. If false, the human substrate is unnecessary. Four comparison corpora now exist to test against rather than construct, and they form a ladder of increasing difficulty. The first is the one this hypothesis was written for and is now public: Yan et al. [130] release 4,465 human and 207 agent improvement trajectories on the same competitions, version-level, with action, intent, edit size and score effect labelled on every version. That is a paired human-and-machine improvement-trace corpus with realized outcomes attached, and it converts this hypothesis from a data-collection problem into an experiment that can be run on release. Agent-generated discovery trees [136] and swarm-generated exploration traces in a physical domain [18] are machine corpora, and the persistence prediction against them is sharp and follows from Section 4.3: a policy does not quit, so neither corpus contains abandonment and the head has nothing to fit.

The third arm is the hard one. A user model that matches human conversational behavior at roughly two orders of magnitude above frontier baselines [50] could in principle generate *simulated human* improvement traces, which would obtain the substrate without manufacturing it and would remove the moat this program is built on. We state the expected result and the reason. A user model trained on internet conversation inherits the artifact bias of Section 1: it has seen what people said, not what they did across attempts at a task with a verifier attached. Simulated improvement is improvement in the shape the training distribution contains, and simulated abandonment is abandonment as the corpus depicted it rather than as it occurred. If that reasoning is wrong, the human substrate is unnecessary and this program should say so rather than defend it. The synthetic arm need not be constructed from scratch: agent swarms that build and explore their own executable worlds already generate large numbers of machine improvement trajectories in physical domains with free verifiers [18], which makes this the cheapest available test of the substrate claim outside code.

Hypothesis 6 (Evaluation validity)**.** Mastery evaluations predict human learning outcomes better than a snapshot benchmark.

Hypothesis 7 (Legal collection at scale)**.** The consent architecture passes review in a second jurisdiction.

Hypothesis 8 (Two-level law of practice)**.** Mastery model v0 generates individual trajectories better fitted by an exponential than a power form, and reproduces an aggregate power law on pooling, with the exponent following from the fitted distribution of individual rates.

Hypothesis 9 (Cross-domain scaling)**.** The fitted exponent η_d in Equation 10 is strictly positive, and η_ℓ is strictly positive and not reducible to N_τ.

Hypothesis 10 (Shift and hacking are containable)**.** Drift between V_θ and realized success on policy trajectories remains below the gain of mastery-optimized RL over outcome RL, and adversarial probe scores do not rise across judge versions.

Hypothesis 11 (The advantage widens with scale)**.** The gap between mastery-optimized RL and the strongest baseline of Hypotheses 2 and 22, measured on the evaluations of Section 10, is non-decreasing across the three policy sizes of Section 11. *Near-lethal if false*: an advantage that narrows with scale is a training detail the frontier absorbs, not a substrate the frontier lacks.

Hypothesis 12 (The observation has not expired)**.** Frontier models available at the time of writing remain at or below a trivial baseline on knowledge tracing and at or below chance on item discrimination [122], and remain subject to the proficiency inversion of Section 3. Resolves in one week and before substrate capture begins. If false, Section 3 has expired and the paper's premise requires restatement rather than defence.

Hypothesis 13 (Component ablation)**.** Each of the five components of Section 5.5 contributes to transfer, and the two substrate-bound components contribute most. The test is a leave-one-out ablation on the month-12 transfer proof at matched compute, reported per component with confidence intervals. We state the prediction in the falsifiable direction: if removing the persistence penalty and external re-anchoring leaves transfer unchanged, the recipe is a dense process reward with an unusual training set, the substrate is not doing the work the thesis attributes to it, and the name should be retired rather than defended. Resolves at month 12 as part of the existing experiment, at the cost of four additional arms.

Hypothesis 14 (Criterion transfer)**.** A mastery model trained on human improvement traces predicts eventual success on the improvement trajectories of *machine* discovery runs better than the run's own outcome metric does at equal horizon. The baseline is now specified rather than notional: the discovery trees of Zheng et al. [136] are public, their replay score is the outcome metric to beat, and their code is released. Resolves at negligible cost on data that already exists. If false, the criterion is bound to human learners and the self-improvement implication of Section 13 does not follow.

Hypothesis 15 (Frontier transfer, not only acceleration)**.** A mastery model fitted on sub-frontier human improvement traces improves a solver that is already at the frontier of its domain, and not only a solver below it. The test holds the mastery model fixed and varies the policy's starting competence across three levels spanning novice-equivalent to best-available, measuring the mastery-RL gain over outcome-GRPO at each. The prediction is that the gain is non-vanishing at the top level. **This is the lethal test of the whole thesis**, and it is the sharpest form of the negative case: that representations of how someone got better generalize downward to learners and not upward to the frontier, because at the frontier the useful moves are the ones no recorded human made. If the gain vanishes as starting competence rises, this is a method for teaching students faster and not a method for extending capability, and we will say so. Resolution: month 12, on the transfer proof of Hypothesis 1.

Hypothesis 16 (Custody is achievable)**.** Contribution without disclosure can be made verifiable to a trace holder: a holder in a domain where process carries private value contributes under an arrangement in which the trace does not leave their environment, and the resulting model update measurably improves V_θ on held-out trajectories. Failure confines acquisition to domains where process has no private value, which bounds the program from above. Section 8.2 states why the published mechanism is not yet sufficient. Resolution: month 12.

Hypothesis 17 (The replayability boundary)**.** On the same discovery trees, a mastery model outperforms exact replay in the three regimes where replay is unavailable by construction: (i) scoring branches absent from the recorded tree, evaluated by holding out subtrees and comparing predicted against realized value; (ii) regions in which every recorded attempt failed, where replay is uninformative by definition; and (iii) a domain without a free per-node verifier, where no outcome exists to replay. In regimes where replay *is* available, we predict it wins, and a system using replay on-tree and the mastery model off-tree beats either alone.

This is the sharpest test in the program and the cheapest. It resolves at month 6 on public data with a public baseline, and it is stated as a prediction rather than a hope: if the mastery model fails to beat exact replay even where replay cannot operate, the learned value adds nothing that recorded history does not already contain, and the thesis narrows to a substrate-collection argument with no modelling claim attached.

Four additions, and what they test that the seventeen do not. The hypotheses above treat improvement as one quantity. It is three, and the distinction governs what the program can claim. Progress *within* an attempt, a durable change in the learner, and a change in how efficiently the learner acquires the *next* capability are different outcomes with different falsifiers, and a method can deliver the first while supplying none of the second. Receiving an answer raises task success and produces no competence. The following four separate them. They also mark the point at which the framework stops being observational: a value fitted to recorded histories predicts, and prediction does not identify what would have changed under a different choice. Randomized assignment does, and these are specified with it.

Hypothesis 18 (Causal intervention selection)**.** Randomized guidance from the mastery model improves independently measured competence at matched budget, against matched learners branched from identical checkpoints with comparable

observations, tools and feedback. *Fails if the gain vanishes once information and resources are equalized*, which would indicate the model was selecting easier experiences rather than more productive ones. Resolution: month 12, on the branched arm of the transfer experiment.

Hypothesis 19 (Durable learning)**.** Gains persist after the guide is withdrawn and on delayed tests over tasks withheld from intervention selection. *Fails if only assisted performance improves*, in which case the mastery model is a scaffold rather than a teacher and its value is bounded by its availability at inference.

Hypothesis 20 (Learner transfer)**.** A frozen mastery model benefits learners it was never fitted on, across architectures, sizes and competence levels, and under combined task and learner shift. *Fails if gains require substantial target-specific fitting*, which would make the object a per-learner artifact and remove the case for a single corpus serving many solvers. Frozen transfer and budgeted adaptation are reported separately, with learned adapters and target data counted as adaptation.

Hypothesis 21 (Improved acquisition)**.** Guided experience improves the learner's acquisition of subsequent unfamiliar skills without ongoing guidance. *Fails if benefits remain confined to the practiced tasks*, which would establish faster learning of what was taught and not an improvement in learning. This is the strongest of the four and the only one that distinguishes a better curriculum from a better learner.

Hypothesis 22 (Synthetic process priming does not close the gap)**.** At matched compute on the evaluations of Section 10, mastery-optimized RL exceeds a behavior-primed policy trained with outcome-GRPO, where priming follows Gandhi et al. [33] in supplying synthetic traces exhibiting verification, backtracking, subgoal setting and backward chaining. *Lethal to the substrate claim if false.* If synthetic priming matches a value fitted to recorded human improvement, then the human corpus is not the scarce input, the manufacturing program of Section 8 is unnecessary, and the defensible position this paper describes does not exist. We regard this as the most likely way the program fails for a reason that is nobody's error.

11.1 Workstreams and milestones

Workstream 6: what can be settled before any substrate exists. Four questions in this paper are answered by assertion or analogy, and all four resolve on data that is already public, before a trace is captured and before the critical-path hire is made.

6.1 The architecture claim, replicated properly (month 3). Section 5 takes as a prior that a recurrent state space model with discrete latents is the right representation for trajectory prediction, on a single result whose margin over item-response baselines is small and whose evaluation is one fixed split without variance estimates. Four comparisons on existing data settle it: cross-validation with variance estimates; extension to one domain off code; an ablation isolating the discrete latents from the recurrence, since discreteness is the architectural claim; and a matched-parameter transformer on the same trajectories. If it holds, it is an inductive-bias claim about modelling learners of any kind, independent of this substrate and publishable alone. If not, the architecture section stops overclaiming at month 3 rather than month 12.

6.2 Persistence-label feasibility, the binding risk (month 1). The persistence head carries Hypothesis 13, Section 5.6 concedes abandonment is the sparsest field in the schema, and nobody has measured how sparse. One week: infer abandonment from an existing timestamped corpus under three operational definitions and report the fraction of traces yielding a usable label under each, with inter-definition agreement. If the yield is low, component 3 of Section 5.5 is aspirational and the specification must say so before it is claimed in public.

6.3 The volume target, derived rather than asserted (month 3). Section 8 names a target beside PRM800K's roughly 800,000 step-level labels without deriving it. Fit mastery-model performance against transition count on subsamples of an existing corpus and extrapolate to the performance required to beat the baselines of Hypothesis 2. This converts the central budget assumption from a comparison into a measurement.

6.4 A first empirical leg for the transfer conjecture (month 3). Section 6 concedes that substrate-invariance rests on analogy, and a corpus released since makes a weak version testable now. Yan et al. [130] record 4,465 human trajectories across 134 Kaggle competitions at the granularity of the code version, with score, timestamp, action, intent and edit size per version; two agent scaffolds were run on seven of those competitions, giving 207 agent trajectories against 430 paired human ones.

The design is three populations and one fitted model. Fit the value head on the human trajectories from the 127 competitions no agent touched. Evaluate on the agent trajectories, and separately on the held-out human trajectories from the same seven competitions. The second arm is what makes the first interpretable: transfer to unseen humans is the ceiling, and the gap between the two arms is the substrate-invariance effect rather than a bare number.

Table 5: Falsifiers and kill tests. Rows 1, 2, 15 and 22 are lethal and row 11 is near-lethal; all five resolve in twelve months on the single matched-compute experiment of Section 11. Row 12 resolves in one week and precedes substrate capture. Row 8 replaces the naive power-law check that the learning-curve literature does not support.

#	Falsifier	Test	By
1	Mastery objective does not transfer off code and math	Code-trained mastery model improves open-ended performance in a non-code domain vs. outcome-GRPO at matched compute	Month 12
2	Outcome, authored-rubric, or learned-rubric RL is sufficient	Matched-compute four-way comparison, scored on mastery evaluations and on nonzero-gradient group fraction	Month 12
3	Substrate cannot be manufactured economically	Cost per trace at or below $200 to $2,000 with intermediate states preserved	Month 9
4	A frontier laboratory in-houses it	A laboratory publishes or ships a reward model trained on human revision traces	Continuous
5	Human traces are no better than machine or simulated-human traces	Three-arm ablation against discovery trees, swarm traces, and user-model-generated traces, on eventual success and on persistence	Month 9
6	Mastery evaluations do not predict human outcomes	Predictive validity no better than a snapshot benchmark	Month 9
7	Substrate cannot be collected legally at scale	Consent framework fails review in a second jurisdiction	Month 6
8	Two-level law of practice not reproduced	Individual trajectories not exponential, or pooled trajectories not power-law with the implied exponent	Month 3
9	Substrate does not scale across learners or domains	Fitted η_d or η_ℓ is zero	Month 9
10	Shift or hacking outruns re-anchoring	Value drift exceeds the gain over outcome RL; probe scores rise across judge versions	Month 12
11	The advantage narrows with scale	Gap over the strongest Hypothesis 2 and 22 baseline decreases across three policy sizes	Month 12
12	The observation has expired	Frontier models now exceed a trivial baseline on knowledge tracing or chance on item discrimination	Month 0
13	The substrate-bound components do nothing	Leave-one-out ablation: removing the persistence penalty and external re-anchoring leaves transfer unchanged	Month 12
14	The criterion is bound to human learners	Mastery model fails to beat the replay score of public discovery trees at predicting eventual success	Month 6
15	The mastery model accelerates learners but does not extend the frontier	Mastery-RL gain over outcome-GRPO vanishes as the policy's starting competence rises across three levels	Month 12
16	Custody cannot be made verifiable	No holder in a private-value domain contributes under an arrangement in which the trace never leaves their environment	Month 12
17	Exact replay suffices, and the learned value adds nothing	Mastery model fails to beat exact replay off-tree, under flat outcomes, or in a domain without a free verifier	Month 6
18	The guidance is correlational, not causal	Randomized guidance from identical checkpoints shows no gain once information and budget are equalized	Month 12
19	The model scaffolds rather than teaches	Gains do not survive withdrawal of the guide, or do not appear on delayed tests	Month 12
20	The object is per-learner, not transferable	A frozen model benefits held-out learners only after substantial target-specific fitting	Month 12
21	Better curriculum, not better learner	Guided experience does not improve acquisition of subsequent unfamiliar skills	Month 24
22	Synthetic process priming is sufficient	A behavior-primed policy under outcome-GRPO matches mastery-optimized RL at matched compute	Month 12
23	The six consequences are six programs	A capability is no easier to reach with the mastery model than without it at matched resources	Staged

Three features of the corpus set what the test can and cannot say, and all three are visible in the release rather than discovered afterwards. The agent side is 11 Codex runs and 13 MLEvolve searches; the 189 MLEvolve branches share tree nodes, so the run count rather than the branch count sets the effective sample size, and four of the Codex runs are shorter than ten transitions. The agent arm is therefore a probe on roughly two dozen independent units, while the 430 held-out human trajectories are properly powered. Second, 45% of human versions carry a leaderboard score against 99% of agent versions, so a value head with access to score is partly measuring that asymmetry; the primary specification is built from state tags and trajectory shape with no score at all, and the score-using arm masks agent scores

to the human rate. Third, outcome is final within-competition leaderboard percentile rather than a medal threshold, because agents finish well outside medal range and a binary label would be degenerate on exactly the arm being tested.

The limitation is worth stating in advance, because it is the obvious objection. Kaggle returns a score on every submission, which is precisely the dense outcome feedback this paper argues is absent in its target domains. The test therefore isolates transfer across substrate with domain held fixed, and establishes nothing about domains where the outcome signal is genuinely flat. A positive result does not establish the conjecture; a negative one is an early warning on Hypothesis 1, which is lethal and otherwise does not resolve until month 12.

Compute for the transfer proof. Hypotheses 1 and 2 are specified at matched compute, so the budget is stated. The month-12 experiment is five arms, outcome-GRPO, authored-rubric, learned-rubric, behavior-primed outcome-GRPO after Gandhi et al. [33], and mastery-optimized, run to matched token budget per arm at three policy sizes in the 8B to 30B class, provisioned at approximately 150,000 accelerator-hours for the arms and 40,000 for the scaling sweep and reruns. The fifth arm was added because the substrate claim is the program's economic load-bearing member and synthetic process priming is the only published method that threatens it directly; the marginal 30,000 hours are the cheapest insurance in the budget. This is post-training scale rather than pretraining scale, and that is the point: the central claim of a position paper about frontier training is falsifiable on a budget a single laboratory can raise. The efficiency workstream is load-bearing for the same reason. If mastery-model grading cannot be driven under 5% of policy rollout cost, the same experiment costs several times the above and the program is not runnable at a scale that would convince anyone. The target is set against current practice rather than in the abstract: the groupwise grader of LLM-Core Xiaomi [73] consumed about 12.7% of total reinforcement-learning cost for their larger model, and that run's entire post-training budget was roughly $2.6M, which also calibrates the scale of the month-12 experiment against a frontier run of the same kind.

Five workstreams: substrate manufacturing (Section 8); the first mastery model, beginning where the CodeInsight state space model left off and passing the two-level consistency check of Section 7.2 before use as a reward; mastery-optimized RL with the defenses of Section 9; efficiency, since dense-reward RL grades every rollout and the mastery model must cost under 5% of policy rollout compute per graded rollout to be runnable at scale; and mastery evaluations. Milestones: at month 3, consented capture live in two domains and mastery model v0 passing the two-level check with a first scaling-curve fit; at month 9, the model beating step-PRM and deep-knowledge-tracing baselines off code, cost per trace at environment-task parity, and evaluations published with stated validity; at month 12, the transfer proof; at month 24, a model with a measurable continual-improvement advantage in the first vertical; at month 36, a second vertical.

Which vertical goes first, reconsidered. Earlier drafts placed software agents first and embodied work second, on the grounds that code has a free verifier and an instrumented substrate. Section 4.3 inverts part of that reasoning. A free per-node verifier is exactly the condition under which exact replay is available, so code is the domain where the non-learned alternative is strongest and a learned value has least to add. The embodied setting has the opposite profile: attempts are expensive, counterfactual branches were never executed and therefore cannot be replayed, and the outcome of an unexplored grasp does not exist to be looked up. The case for ordering the verticals the other way is now stronger than the case that produced the current schedule, and Hypothesis 17 is what decides it. Two qualifications restrain the inference. The embodied setting is where our negative claim already fails [128], so a mastery model enters it as one component of a recipe rather than as a new object. And the binding operational constraint there is human attention: in the system of Dong and Finn [24] a person defines success, resets the scene and intervenes when the robot goes wrong, so scaling past one operator per robot is what a learned progress criterion would actually buy. That is a narrower and more defensible claim than displacing the control algorithm, and it is the form in which we would test it, preferably on a partner's hardware.

12 On Publication and Capture

A program whose unlock is a corpus has a known failure mode, and the two closest historical analogues both exhibit it. The corpus that made modern computer vision possible was published, and the value accrued to the group that trained on it. The architecture that made modern language modelling possible was published, and its authors dispersed. Neither outcome followed from a mistake; both followed from the structure of what was built.

This program is shaped like the first. The method is cheap and the corpus is expensive, the corpus takes years and consent rather than compute, and the result appears in someone else's model unless something prevents it. We therefore separate four layers and treat them differently.

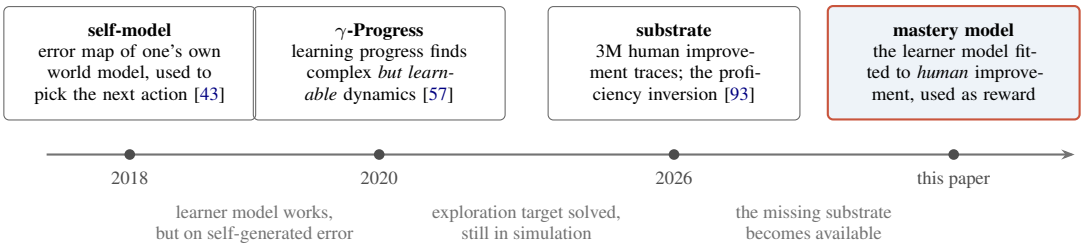


Figure 12: What is new and what is not. A learner-side model that selects the next attempt was built in 2018; a learning-progress signal that locates the complex-but-learnable band was built in 2020. Both fitted the learner model to an agent's own instantaneous prediction error in simulation. The contribution of this paper is not that a learner model is useful, but that the same machinery fitted to a manufactured record of *humans* improving becomes a reward that transfers to frontier post-training.

The **recipe** is published, including the full specification of Section 5.5, because it cannot be run without the corpus and publishing it makes that legible. The **evaluations** are published under a permissive licence, because the author of a measurement sets the axis on which others are scored, and we would rather this became a standard than a differentiator. The **corpus** is never published, in whole or in representative part, because consent, time and institutional relationships are the only components of this program that cannot be compressed by capital.

The **model** is neither published nor licensed, and it is also not shipped. It is served: a training-time interface returning a graded advantage per rollout, priced per grading, callable from inside another laboratory's training loop. This follows from the commercial principle rather than extending it, since serving returns the model's outputs while licensing would transfer the capability itself. We state it because it is the layer neither historical analogue had, and it is the reason the failure mode above is not forced.

13 Discussion

Why now. Four conditions became true between late 2025 and 2026. The coverage limit of outcome-verified RL was measured and named [127, 131, 132]. Trajectory degradation was measured at scale rather than argued [63, 81]. Capture of improvement at scale was demonstrated, and modeling it produced the proficiency inversion [93, 102]. And a laboratory built on a manufactured human substrate proved out as both a research institution and a company: Simile, founded by the group behind generative agents [90, 91], reached a multi-billion-dollar valuation within eighteen months of the founding paper on a foundation model of human behavior trained on interviews and large-scale studies (Tier 2 reporting, July 2026; see Appendix C). The relationship is complementary rather than competitive, and the distinction is the one drawn in Section 4: that program models the state of a population at a point in time, and a population simulation that could learn would require the object this paper defines.

What is new and what is not. Neither half of this proposal is unprecedented. A model of the learner used to select the next attempt was built and shown to generate structured behavior in 2018 [43]; a learning-progress signal that finds the complex-but-learnable band was built and shown to beat standard exploration baselines in 2020 [57]. What was missing in both was the substrate. Those systems fitted the learner model to an agent's own instantaneous prediction error in simulation, which is available but thin. This paper's claim is that the same machinery, fitted instead to a manufactured record of humans improving, becomes a reward that transfers to frontier post-training. The novelty is the substrate and the objective built on it, not the idea that a learner-side model is useful.

Implications for open-ended learning. If the learnability frontier can be computed from a model of the learner rather than discovered by trial, the open-endedness program [20, 49, 113, 121] gains an instrument it has lacked: a way to generate tasks at the edge of a *specific* learner's ability rather than from population statistics of an environment. This is also the sharpest point of contact with Silver and Sutton [110]: self-generated experience needs a criterion for which experience is worth generating, and that criterion is a mastery model.

Regimes rather than architectures, revisited. Section 1 read the period since 2017 as three regimes over one architecture, each unlocked by a corpus rather than a model, and placed this program as a bid on the fourth signal. Two consequences of that reading bear on everything above. It predicts where the difficulty sits, which is in manufacturing rather than in method, and the structure of this program reflects that: a published recipe and an unpublished corpus. It also predicts the shape of the competitive risk. A regime defined by a corpus is not defended by secrecy about the method, and Hypothesis 4 is the test of whether it is defended at all.

13.1 The scope of the program

The implications collected below are usually presented as separate research agendas. They are not separate. Six of the hardest open problems in machine learning each require a judgment about whether a learner moved toward competence, and that judgment is the artifact this paper defines. We state the architecture explicitly because a list of ambitions cannot be falsified and a structure of dependencies can.

The result the program turns on. One result carries everything below, and we state it once here rather than qualifying each consequence separately. A value fitted to sub-frontier human improvement must guide a solver past the competence present in its training records. We expect that it does, for the reason the program exists: recovering from a wrong turn, abandoning a line that is not working, and recognizing progress before the outcome confirms it are properties of improvement rather than properties of any particular improver. A model fitted to ten thousand people recognizing a dead end has learned to recognize a dead end, including one no solver in the corpus ever reached. Hypothesis 15 is the test, it is specified at three starting competence levels with the mastery model held fixed, and it resolves at month 12. If the gain vanishes as starting competence rises, this is a method for teaching students faster and not a method for extending capability, and the rest of this subsection does not follow.

The complement. A world model predicts how an environment changes under an action; a mastery model predicts how a solver changes under an attempt. Neither contains the other, and an agent that must both act well now and learn efficiently needs both halves of Figure 3. The field has built the first at scale. The position we take is integration rather than competition: constructing a rival environmental model would consume the program without advancing its distinctive claim, and the test that matters is whether modeling the learner improves retained capability beyond an equally resourced model of the environment alone.

The six consequences. Each follows from the same object rather than requiring a separate one.

1. **Continual development.** Forgetting is a negative change in retained competence and loss of plasticity is a negative change in the rate at which competence changes [23]. Both are quantities a model of learner transition predicts by construction, and neither is visible to outcome supervision until the capability is already gone. What the mastery model does not supply is the remedy: memory, rehearsal and consolidation mechanisms remain to be built on top of the detection.
2. **Abstraction acquisition.** Internalizing a reusable rule is the developmental event in its purest form, and it is precisely the event outcome supervision cannot see, because before and after the acquisition the artifact can look identical while the learner does not. This is a candidate explanation for why abstraction has resisted outcome supervision rather than yielding to scale.
3. **A transferable learning procedure.** A value that guides learners it was never fitted on, across architectures, sizes and competence levels, is a transferable learning procedure by definition. There is no second object to invent here, which makes this the consequence most tightly coupled to the central result and the first we attempt.
4. **Resource efficiency.** Intervention selection that causes learning reaches a given competence in fewer attempts, so sample efficiency is what the central claim means rather than a separate objective. The conversion into compute efficiency is not automatic and runs through the cost of grading: the target of Section 11, holding the mastery model under 5% of policy rollout cost, is what separates an efficiency claim from an efficiency hope.
5. **Experiment selection in discovery.** Choosing the experiment that best discriminates between competing mechanisms is intervention selection under a different objective over the same object. The remaining apparatus, domain knowledge, hypothesis generation, execution and external validation, is substantial and already partly built elsewhere [86]; the contribution would be which experiment gets run.
6. **The evaluation step in self-improvement.** Treated at length below.

The order the architecture dictates. The structure is the research sequence and requires no separate roadmap. The carrying result first, at month 12. Transfer next, because it is the same object on one more axis. Continual development and abstraction after that, each gated on retention measured with the guide withdrawn. Efficiency is reported at every stage rather than scheduled as one. The complement and discovery enter through partners. Self-improvement is last and is conditional on every result before it.

Hypothesis 23 (Structural dependence)**.** For each consequence above, an arm with the mastery model outperforms a matched-resource arm without one. *Fails, for any given consequence, if the capability is no easier to reach with the model than without it*, in which case that consequence is a separate program and will be stated as one rather than carried by association. This hypothesis tests the architecture of this subsection rather than any result within it.

Why the confidence is placed where it is. The structural claim does not wait on an experiment. That six capabilities each require a judgment about whether a learner moved toward competence is an argument about what those problems are, and any reader can evaluate it on the page. What the month-12 experiment settles is how well this artifact supplies that judgment, not whether the judgment is the bottleneck. The program is therefore unusually exposed and unusually specific at the same time, and for the same reason: one result carries it, and it is named, dated and costed in Section 11.

The other rows. Section 2 noted that a missing transition model implies more than one artifact, and the RSI roadmap literature has since named them as levels: experience acquisition at L3, environment adaptation at L4, and revision of the improver or verifier itself at L5 [25]. Their own assessment is that only the first three levels have working instances. Each of those levels requires a model of how the learner transitions, and none of the cited systems supplies one. Three follow directly and we name them without pursuing them here. A curriculum that selects what a learner should encounter next requires a model of how that learner transitions, which is what the learning-progress and open-endedness literatures approximate by search [49, 88, 121]. A population of simulated agents that could develop rather than merely persist requires the same object [50, 90]. And self-improvement requires it most directly of all.

The gap, named by the laboratory nearest to closing the loop. Favaro and Clark [27] report that more than 80% of the code merged into their codebase is now written by their models, and that the models match or exceed skilled humans at executing a well-specified experiment. What persists, in their account, is judgment: choosing which problems matter, "which results to trust, and when an approach is a dead end." They describe that as the gap between current systems and one that could design its own successor. Two of those three are the value and persistence heads of Definition 3. The first, choosing which problems matter, is goal selection, and we do not claim it.

Implications for self-improvement. A mastery model is an improvement criterion that operates where outcomes are flat, which makes it the missing evaluation function for any system attempting to improve itself. Generating a candidate modification is not the hard part and is getting easier; deciding whether a modification was an improvement is, and the current answer is a benchmark score, which is outcome supervision and inherits the coverage failure of Section 3. On a problem where every variant fails, the score is identical and the loop has no signal about which variant was closer. Buehler [18] is the closest published system to such a loop, and it scales by spending more agent-hours rather than by choosing better ones, which is what an improvement criterion would change. Zheng et al. [136] close the loop more completely and at the layer this paper cares about: the object being improved is the exploration policy rather than the weights, and it is improved from the system's own recorded history. Their criterion is exact replay, which is the right instrument where history is replayable and is unavailable everywhere else (Section 4.3). That two independent groups arrived at recorded improvement history as the substrate for self-improvement, from different directions, is the strongest available evidence that the substrate is the right object.

An earlier version of this paper noted the implication and declined to pursue it. We replace that with a conditional objective, because declining to state a goal is not the same as not having one, and because the sequence in Section 13.1 makes the condition precise. We pursue a contribution to self-improvement by testing whether developmental models improve the effectiveness of successor development, in this order: externally evaluated intervention selection first, then discovery of better learning procedures, and only then successive improvers measured against fixed-guide and fixed-agent controls at matched research budgets. Each stage is gated on the one before it.

The relevant observation is a safety one and it is why the ordering is not arbitrary. The failure mode of self-improvement is a system optimizing a criterion it also controls. The re-anchoring architecture of Section 9 exists precisely to prevent that: the value head is periodically re-fitted to held-out human trajectories with realized outcomes, and drift between predicted and realized success is a stopping condition. A self-improvement loop evaluated by a criterion repeatedly re-grounded in an external human record is a different object from one evaluated by a criterion that drifts with the policy, and we regard that distinction as the substantive contribution here rather than the loop itself. Hypothesis 14 makes the claim falsifiable.

Two conditions bound what such a result would establish. The number of generations, budgets, human interventions, evaluation tasks and stopping rules are fixed in advance, and final evaluation stays outside the optimizing system's control. A recursive contribution is established only when *updating the improver* causes the additional gains, since accumulating compute or data explains stronger successors without it. A finite sequence of improvements is not indefinite acceleration and we do not read it as one.

Implications for simulation. A population of generative agents equipped with mastery models is a population of learners. Classrooms, firms, and scientific communities are learning systems, and a simulation of one that cannot learn is a snapshot.

Implications for embodied systems. A robot policy that plans against a world model and is trained against a mastery model optimizes both halves of Figure 3, and the human corrections it receives become substrate rather than waste.

The civilizational implication. The record culture discarded can be rebuilt. If the program succeeds, an archive of how humans got better at things will exist for the first time, under consent, as a training substrate. That archive is worth constructing independently of any single model trained on it.

14 Limitations

This is a position paper. It reports no new experimental result of its own, and what follows are the precise limits of what it does claim, each of which the program in Section 11 is built to resolve. The proficiency inversion is established in one domain and on three of four courses. The exploration wall in its strong form is refuted by Liu et al. [71], and we rely only on its operative form. Process supervision is contested, not settled [135]. The idea of training on human edits is not novel [74, 96, 100]; the contribution is the value-function framing rather than the data type, and that boundary is narrow. The negative claim in Section 2 is false without its domain and semantic qualifiers, and Xiao et al. [128] satisfy all three of its conjuncts in robot manipulation; the surviving claim is restricted to language, reasoning and symbolic domains and to a value over the solver's competence trajectory rather than over task progress. The dense signed policy-invariant shaping mechanism is likewise already published with an invariance proof [117], so the contribution is the conditioning rather than the mechanism. Of the five identification problems in Section 5.6, the sparsity of abandonment labels is the one we consider most likely to bind, and it constrains the head that most distinguishes this work. Automating exploration from recorded improvement history is likewise no longer an open position. Zheng et al. [136] do it, at frontier scale, with released code, and where the recorded history is exact and replayable their approach is better than a learned value and should be used instead. This paper's claim is scoped accordingly, and Section 4.3 states the boundary rather than contesting it. Substrate-invariance is a conjecture supported by two results and one analogy. Equation 10 is a measurement plan, not a measurement, and no published scaling law separates labeler diversity from label count. The mastery advantage is potential-based and therefore optimum-preserving. The paper's claim is correspondingly about reachability under finite compute rather than about asymptotic capability, and Section 1 states it that way. Readers who require a claim about the ceiling will not find one here, and we regard a defensible claim about what training runs actually reach as the more useful object. Covariate shift is present from the first gradient step and re-anchoring bounds it without eliminating it. The example trace in Section 1 is a schematic composite and will be replaced with a real anonymized trajectory when one exists under the consent architecture. The scope argument of Section 13.1 is an argument and not a result: five of its six consequences are unattempted, none of the four causal hypotheses has been run, and the claim that these capabilities share one bottleneck is offered for evaluation on the page rather than demonstrated. Hypothesis 23 is how it would be shown false. We also note that the framework as formalized is observational, fitted to recorded histories and therefore identifying prediction rather than intervention effects; Hypothesis 18 is the first point at which that changes, and until it resolves the control language in Section 13.1 describes an objective rather than a demonstrated capability.

15 Ethics and Societal Considerations

The substrate is a record of people learning, which is a record of their failures. The consent architecture of Section 8 is the primary mitigation, and its purpose-limitation clause matters most: traces train models of how competence develops and never evaluate the individuals who produced them. A model of persistence could be used to manipulate persistence; c_θ is a component of a training reward and is not exposed as a product for acting on people. Misidentification of the program as education technology would invite deployment on learners before the science supports it. Against these, the benefits: models that improve rather than degrade over long interactions, which is the failure Laban et al. [63] and Moore et al. [81] measured, and an archive of human improvement that did not previously exist.

16 Conclusion

The frontier trains on what intelligence produced. Developmental intelligence trains on how intelligence develops. The omission is older than machine learning: every recording technology that expanded what a civilization could transmit captured a state and discarded the process, and the corpus these models are trained on is the output of that selection. The first technology able to record process at scale is the one currently trained on its absence. We have named the form, defined the artifact class, specified the substrate, formalized the objective and confronted the strongest formal objection to it, grounded it in six sciences that already treat trajectory and error as the object of study, and stated how it dies across twenty-three dated kill tests, five of them lethal or near-lethal. The central claim resolves within twelve months

with one matched-compute experiment. If it is wrong, the program will say so. If it is right, the field has been modeling one half of the agent loop for a decade, and the other half is still unbuilt.

References

[1] Deep knowledge tracing and dropout prediction with self-attentive models. *arXiv preprint arXiv:2210.09012*, 2022.

[2] JE-IRT: Joint estimation of item response theory parameters for language model evaluation. *arXiv preprint arXiv:2509.22888*, 2025.

[3] Self-play with variational problem synthesis sustains RLVR. *arXiv preprint arXiv:2508.14029*, 2025.

[4] How to utilize failure demonstration data in imitation learning. *arXiv preprint arXiv:2605.07560*, 2026.

[5] Pedagogical safety in educational reinforcement learning. *arXiv preprint arXiv:2604.04237*, 2026.

[6] Rubric-ARM: Learning to generate rubrics as latent actions. *arXiv preprint arXiv:2602.01511*, 2026.

[7] RLR-cubed: Criterion-level routing for rubric rewards. *arXiv preprint arXiv:2605.30244*, 2026.

[8] Ahmed M. Alaa and Mihaela van der Schaar. Attentive state-space modeling of disease progression. In *Advances in Neural Information Processing Systems (NeurIPS)*, 2019.

[9] Dario Amodei, Chris Olah, Jacob Steinhardt, Paul Christiano, John Schulman, and Dan Mané. Concrete problems in AI safety. *arXiv preprint arXiv:1606.06565*, 2016.

[10] John R. Anderson. Acquisition of cognitive skill. *Psychological Review*, 89(4):369–406, 1982.

[11] Paniz Behboudian, Yash Satsangi, Matthew E. Taylor, Anna Harutyunyan, and Michael Bowling. Policy invariant explicit shaping: An efficient alternative to reward shaping. *Neural Computing and Applications*, 2022.

[12] Yoshua Bengio, Jérôme Louradour, Ronan Collobert, and Jason Weston. Curriculum learning. In *Proceedings of the 26th International Conference on Machine Learning (ICML)*, 2009.

[13] Ioana Bica, Ahmed M. Alaa, James Jordon, and Mihaela van der Schaar. Estimating counterfactual treatment outcomes over time through adversarially balanced representations. In *International Conference on Learning Representations (ICLR)*, 2020.

[14] Marcel Binz et al. Centaur: A foundation model of human cognition. *arXiv preprint arXiv:2410.20268*, 2024.

[15] Robert A. Bjork. Memory and metamemory considerations in the training of human beings. In Janet Metcalfe and Arthur P. Shimamura, editors, *Metacognition: Knowing About Knowing*, pages 185–205. MIT Press, 1994.

[16] Benjamin S. Bloom. Learning for mastery. *Evaluation Comment*, 1(2), 1968.

[17] Benjamin S. Bloom. The 2 sigma problem: The search for methods of group instruction as effective as one-to-one tutoring. *Educational Researcher*, 13(6):4–16, 1984.

[18] Markus J. Buehler. Artificial intelligence agents autonomously build computational laboratories that reveal design principles of hierarchical metamaterial failure. *arXiv preprint arXiv:2608.26081*, 2026. Preprint.

[19] Kathryn Chaloner and Isabella Verdinelli. Bayesian experimental design: A review. *Statistical Science*, 10(3): 273–304, 1995.

[20] Jeff Clune. AI-GAs: AI-generating algorithms, an alternate paradigm for producing general artificial intelligence. *arXiv preprint arXiv:1905.10985*, 2019.

[21] Albert T. Corbett and John R. Anderson. Knowledge tracing: Modeling the acquisition of procedural knowledge. *User Modeling and User-Adapted Interaction*, 4(4):253–278, 1994.

[22] DeepSeek-AI. DeepSeek-R1: Incentivizing reasoning capability in LLMs via reinforcement learning. *arXiv preprint arXiv:2501.12948*, 2025.

[23] Shibhansh Dohare, J. Fernando Hernandez-Garcia, Qingfeng Lan, Parash Rahman, A. Rupam Mahmood, and Richard S. Sutton. Loss of plasticity in deep continual learning. *Nature*, 632:768–774, 2024.

[24] Perry Dong and Chelsea Finn. Towards universal post-training for robotics. Blog post, September 2026. pd-perry.github.io/posts/post-training.html.

[25] Yi Duan, Ying Liu, Zirui Tang, Haodong Chen, Jun Zhou, Xuanhe Zhou, Fan Wu, et al. The last AI built by humans: Toward genuine recursive self-improvement. *arXiv preprint arXiv:2609.11873*, 2026. v2, 15 September 2026. 35 authors across Shanghai Jiao Tong, Theseus Labs, Tsinghua, ByteDance, ModelBest, Xiaohongshu, Shanghai AI Lab and others.

[26] K. Anders Ericsson, Ralf T. Krampe, and Clemens Tesch-Römer. The role of deliberate practice in the acquisition of expert performance. *Psychological Review*, 100(3):363–406, 1993.

[27] Marina Favaro and Jack Clark. When AI builds itself. The Anthropic Institute, September 2026. anthropic.com/institute/recursive-self-improvement.

[28] Daniel Fein, Max Lamparth, Violet Xiang, Mykel Kochenderfer, and Nick Haber. One bias after another: Mechanistic reward shaping and persistent biases in language reward models. In *Proceedings of the 43rd International Conference on Machine Learning (ICML)*, volume 306 of *Proceedings of Machine Learning Research*, pages 29862–29896, 2026.

[29] Chelsea Finn, Pieter Abbeel, and Sergey Levine. Model-agnostic meta-learning for fast adaptation of deep networks. In *Proceedings of the 34th International Conference on Machine Learning (ICML)*, 2017.

[30] Paul M. Fitts and Michael I. Posner. *Human Performance*. Brooks/Cole, 1967.

[31] David J. Foster and Matthew A. Wilson. Reverse replay of behavioural sequences in hippocampal place cells during the awake state. *Nature*, 440:680–683, 2006.

[32] Xinghong Fu, Aravinth Kulanthaivelu, and Yutaro Yamada. Self improvement via fast tree-search. *arXiv preprint arXiv:2609.19526*, 2026. MIT and Sakana AI.

[33] Kanishk Gandhi, Ayush Chakravarthy, Anikait Singh, Nathan Lile, and Noah D. Goodman. Cognitive behaviors that enable self-improving reasoners, or, four habits of highly effective STaRs. In *Conference on Language Modeling (COLM)*, 2025. arXiv:2503.01307. Stanford University and SynthLabs. Code at github.com/kanishkg/cognitive-behaviors.

[34] Leo Gao, John Schulman, and Jacob Hilton. Scaling laws for reward model overoptimization. *Proceedings of the 40th International Conference on Machine Learning (ICML)*, 2023.

[35] Zhilong Ge, Yuting Shao, Yutao Yang, Yuxuan Cai, Jie Zhou, Kai Chen, Xin Li, Bo Zhang, Qin Chen, and Liang He. SkillGym: Internalizing large-scale human skills into LLMs for real-world problem solving. *arXiv preprint arXiv:2609.27717*, 2026. East China Normal University and Shanghai AI Laboratory.

[36] William J. Gehring, Brian Goss, Michael G. H. Coles, David E. Meyer, and Emanuel Donchin. A neural system for error detection and compensation. *Psychological Science*, 4(6):385–390, 1993.

[37] Ziqin Gong, Ning Li, and Huaikang Zhou. Machines acquire scientific taste from institutional traces. *arXiv preprint arXiv:2603.16659*, 2026. Tsinghua University.

[38] Alex Graves, Marc G. Bellemare, Jacob Menick, Rémi Munos, and Koray Kavukcuoglu. Automated curriculum learning for neural networks. In *Proceedings of the 34th International Conference on Machine Learning (ICML)*, 2017.

[39] Matthias J. Gruber, Bernard D. Gelman, and Charan Ranganath. States of curiosity modulate hippocampus-dependent learning via the dopaminergic circuit. *Neuron*, 84(2):486–496, 2014.

[40] Etash Guha et al. OpenThoughts: Data recipes for reasoning models. *arXiv preprint arXiv:2506.04178*, 2025.

[41] Anisha Gunjal, Anthony Wang, Elaine Lau, Vaskar Nath, Bing Liu, and Sean Hendryx. Rubrics as rewards: Reinforcement learning beyond verifiable domains. In *International Conference on Learning Representations (ICLR)*, 2026. arXiv:2507.17746.

[42] David Ha and Jürgen Schmidhuber. World models. *arXiv preprint arXiv:1803.10122*, 2018.

[43] Nick Haber, Damian Mrowca, Stephanie Wang, Li Fei-Fei, and Daniel L. K. Yamins. Learning to play with intrinsically-motivated self-aware agents. In *Advances in Neural Information Processing Systems (NeurIPS) 31*, pages 8388–8399, 2018. arXiv:1802.07442.

[44] Danijar Hafner, Timothy Lillicrap, Ian Fischer, Ruben Villegas, David Ha, Honglak Lee, and James Davidson. Learning latent dynamics for planning from pixels. In *Proceedings of the 36th International Conference on Machine Learning (ICML)*, 2019.

[45] John Hattie and Helen Timperley. The power of feedback. *Review of Educational Research*, 77(1):81–112, 2007.

[46] Andrew Heathcote, Scott Brown, and D. J. K. Mewhort. The power law repealed: The case for an exponential law of practice. *Psychonomic Bulletin & Review*, 7(2):185–207, 2000. doi:10.3758/BF03212979.

[47] Jordan Hoffmann et al. Training compute-optimal large language models. *arXiv preprint arXiv:2203.15556*, 2022.

[48] Clay B. Holroyd and Michael G. H. Coles. The neural basis of human error processing: Reinforcement learning, dopamine, and the error-related negativity. *Psychological Review*, 109(4):679–709, 2002.

[49] Edward Hughes, Michael Dennis, Jack Parker-Holder, Feryal Behbahani, Aditi Mavalankar, Yuge Shi, Tom Schaul, and Tim Rocktäschel. Open-endedness is essential for artificial superhuman intelligence. *arXiv preprint arXiv:2406.04268; ICML 2024*, 2024.

[50] humans&. Introducing Persimmon: A user model for simulating people in open-ended interactions, September 2026. Research preview. persimmon.humansand.ai.

[51] Minqi Jiang, Edward Grefenstette, and Tim Rocktäschel. Prioritized level replay. In *Proceedings of the 38th International Conference on Machine Learning (ICML)*, 2021.

[52] Haoxiang Kang and Ming Wen. SkillLift: Learning dense rubrics from sparse oracles for efficient skill evolution. *arXiv preprint arXiv:2609.15396*, 2026. Fudan University.

[53] Jared Kaplan, Sam McCandlish, Tom Henighan, Tom B. Brown, Benjamin Chess, Rewon Child, Scott Gray, Alec Radford, Jeffrey Wu, and Dario Amodei. Scaling laws for neural language models. *arXiv preprint arXiv:2001.08361*, 2020.

[54] Manu Kapur. Productive failure. *Cognition and Instruction*, 26(3):379–424, 2008.

[55] Michael Kelly, Chelsea Sidrane, Katherine Driggs-Campbell, and Mykel J. Kochenderfer. HG-DAgger: Interactive imitation learning with human experts. In *IEEE International Conference on Robotics and Automation (ICRA)*, 2019.

[56] Mohammad Khajah, Robert V. Lindsey, and Michael C. Mozer. How deep is knowledge tracing? *arXiv preprint arXiv:1604.02416; Educational Data Mining*, 2016.

[57] Kuno Kim, Megumi Sano, Julian De Freitas, Nick Haber, and Daniel Yamins. Active world model learning with progress curiosity. In *Proceedings of the 37th International Conference on Machine Learning (ICML)*, volume 119, pages 5306–5315, 2020. arXiv:2007.07853.

[58] James Kirkpatrick et al. Overcoming catastrophic forgetting in neural networks. *Proceedings of the National Academy of Sciences*, 114(13):3521–3526, 2017.

[59] Akaash Kolluri, Shengguang Wu, Joon Sung Park, and Michael S. Bernstein. Finetuning LLMs for human behavior prediction in social science experiments. In *Proceedings of the 2025 Conference on Empirical Methods in Natural Language Processing (EMNLP)*, 2025.

[60] Rahul G. Krishnan, Uri Shalit, and David Sontag. Deep Kalman filters. *arXiv preprint arXiv:1511.05121*, 2015.

[61] Rahul G. Krishnan, Uri Shalit, and David Sontag. Structured inference networks for nonlinear state space models. In *Proceedings of the AAAI Conference on Artificial Intelligence*, 2017.

[62] Thomas S. Kuhn. *The Structure of Scientific Revolutions*. University of Chicago Press, 1962.

[63] Philippe Laban, Hiroaki Hayashi, Yingbo Zhou, and Jennifer Neville. LLMs get lost in multi-turn conversation. *arXiv preprint arXiv:2505.06120; ICLR 2026*, 2025.

[64] Jean Lave and Etienne Wenger. *Situated Learning: Legitimate Peripheral Participation*. Cambridge University Press, 1991.

[65] Yann LeCun. A path towards autonomous machine intelligence. *Open Review*, 2022.

[66] Mina Lee, Megha Srivastava, Amelia Hardy, John Thickstun, Esin Durmus, Ashwin Paranjape, Ines Gerard-Ursin, Xiang Lisa Li, Faisal Ladhak, Frieda Rong, Rose E. Wang, Minae Kwon, Joon Sung Park, Hancheng Cao, Tony Lee, Rishi Bommasani, Michael Bernstein, and Percy Liang. Evaluating human-language model interaction. *Transactions on Machine Learning Research*, 2023.

[67] Enrico Bottazzi Leku and Pia Park. Security awareness in LLM agents: The NDAI zone case. *arXiv preprint arXiv:2603.19011*, 2026.

[68] Hunter Lightman, Vineet Kosaraju, Yura Burda, Harri Edwards, Bowen Baker, Teddy Lee, Jan Leike, John Schulman, Ilya Sutskever, and Karl Cobbe. Let's verify step by step. *arXiv preprint arXiv:2305.20050*, 2023.

[69] Dennis V. Lindley. On a measure of the information provided by an experiment. *The Annals of Mathematical Statistics*, 27(4):986–1005, 1956.

[70] Haozhe Liu, Tian Ye, Sensen Gao, Qihang Cao, Yitong Li, Mingchen Zhuge, Enze Xie, Song Han, et al. SoL-Pi: Recursively scaling auto-research loops for efficient agent harness. *arXiv preprint arXiv:2609.20519*, 2026. NVIDIA, NTU and MIT.

[71] Mingjie Liu, Shizhe Diao, Ximing Lu, Jian Hu, Xin Dong, Yejin Choi, Jan Kautz, and Yi Dong. ProRL: Prolonged reinforcement learning expands reasoning boundaries in large language models. *arXiv preprint arXiv:2505.24864; NeurIPS 2025*, 2025.

[72] Nelson F. Liu, Kevin Lin, John Hewitt, Ashwin Paranjape, Michele Bevilacqua, Fabio Petroni, and Percy Liang. Lost in the middle: How language models use long contexts. *Transactions of the Association for Computational Linguistics*, 12:157–173, 2024. doi:10.1162/tacl_a_00638.

[73] LLM-Core Xiaomi. MiMo-V2.6: Scaling reinforcement learning towards self-improvement. Technical report, September 2026. alphaxiv.org/abs/2609.mimo-scaling-reinforcement-learning.

[74] Ruofan Lu, Yintong Huo, Meng Zhang, Yichen Li, and Michael R. Lyu. Next edit prediction: Learning to predict code edits from context and interaction history. *arXiv preprint arXiv:2508.10074*, 2025.

[75] Xing Han Lù and Siva Reddy. Structured distillation of web agent capabilities enables generalization. *arXiv preprint arXiv:2604.07776*, 2026.

[76] Yuxing Lu, Yicheng Chen, Shanchan Wu, and Sercan Ö. Arık. Procedural graphs: Self-evolving execution structures for LLM agents. *arXiv preprint arXiv:2609.09153*, 2026. Google, Georgia Tech and Peking University.

[77] Brooke N. Macnamara, David Z. Hambrick, and Frederick L. Oswald. Deliberate practice and performance in music, games, sports, education, and professions: A meta-analysis. *Psychological Science*, 25(8):1608–1618, 2014.

[78] Dakota Mahan, Duy Van Phung, Rafael Rafailov, Chase Blagden, Nathan Lile, Louis Castricato, Jan-Philipp Fränken, Chelsea Finn, and Alon Albalak. Generative reward models. *arXiv preprint arXiv:2410.12832*, 2024.

[79] Niloofar Mireshghallah, Hyunwoo Kim, Xuhui Zhou, Yulia Tsvetkov, Maarten Sap, Reza Shokri, and Yejin Choi. Can LLMs keep a secret? testing the limits of privacy reasoning via contextual integrity theory. *arXiv preprint arXiv:2310.17884*, 2024. ICLR.

[80] Jared Moore, Declan Grabb, William Agnew, Kevin Klyman, Stevie Chancellor, Desmond C. Ong, and Nick Haber. Expressing stigma and inappropriate responses prevents LLMs from safely replacing mental health providers. In *Proceedings of the 2025 ACM Conference on Fairness, Accountability, and Transparency (FAccT)*, pages 599–627, 2025. doi:10.1145/3715275.3732039.

[81] Jared Moore et al. DelusionEval: Measuring delusion-linked behaviors in AI chatbots. *arXiv preprint arXiv:2608.05004*, 2026.

[82] Jaehyun Nam, Jinsung Yoon, Yanzhou Pan, Yubo Wang, Rui Meng, Parthasarathy Ranganathan, and Tomas Pfister. ScientistTwo: Pioneering the human knowledge frontier with autonomous AI. *arXiv preprint arXiv:2609.19644*, 2026. Google Cloud AI Research.

[83] Allen Newell and Paul S. Rosenbloom. Mechanisms of skill acquisition and the law of practice. In John R. Anderson, editor, *Cognitive Skills and Their Acquisition*, pages 1–55. Erlbaum, 1981.

[84] Andrew Y. Ng, Daishi Harada, and Stuart Russell. Policy invariance under reward transformations: Theory and application to reward shaping. In *Proceedings of the 16th International Conference on Machine Learning (ICML)*, pages 278–287, 1999.

[85] Aarati Andrea Noronha, Kavya Ravikumar, and Carly Xiaoyu Lin. AutoGym: Blueprint-first generation of verifiable agent gyms. *arXiv preprint arXiv:2609.22592*, 2026. Amazon AGI.

[86] Alexander Novikov, Ngân Vũ, Marvin Eisenberger, Emilien Dupont, Po-Sen Huang, Adam Zsolt Wagner, Sergey Shirobokov, Borislav Kozlovskii, Francisco J. R. Ruiz, Abbas Mehrabian, M. Pawan Kumar, Abigail See, Swarat Chaudhuri, George Holland, Alex Davies, Sebastian Nowozin, Pushmeet Kohli, and Matej Balog. AlphaEvolve: A coding agent for scientific and algorithmic discovery. *arXiv preprint arXiv:2506.13131*, 2025.

[87] Open X-Embodiment Collaboration. Open X-Embodiment: Robotic learning datasets and RT-X models. *arXiv preprint arXiv:2310.08864*, 2023.

[88] Pierre-Yves Oudeyer, Frederic Kaplan, and Verena V. Hafner. Intrinsic motivation systems for autonomous mental development. *IEEE Transactions on Evolutionary Computation*, 11(2):265–286, 2007.

[89] Daphna Oyserman and Andrew Dawson. Successful learning environments support and harness students' identity-based motivation: A primer. *The Journal of Experimental Education*, 89(3), 2021. doi:10.1080/00220973.2021.1873091.

[90] Joon Sung Park, Joseph C. O'Brien, Carrie J. Cai, Meredith Ringel Morris, Percy Liang, and Michael S. Bernstein. Generative agents: Interactive simulacra of human behavior. In *Proceedings of the 36th Annual ACM Symposium on User Interface Software and Technology (UIST)*, 2023.

[91] Joon Sung Park, Carolyn Q. Zou, Aaron Shaw, Benjamin Mako Hill, Carrie J. Cai, Meredith Ringel Morris, Robb Willer, Percy Liang, and Michael S. Bernstein. Generative agent simulations of 1,000 people. *arXiv preprint arXiv:2411.10109*, 2024.

[92] Fagun Patel, Duc Q. Nguyen, Sang T. Truong, Jody Vaynshtok, Sanmi Koyejo, and Nick Haber. The sound of syntax: Finetuning and comprehensive evaluation of language models for speech pathology. In *Proceedings of the 2025 Conference on Empirical Methods in Natural Language Processing (EMNLP)*, pages 34907–34925, 2025. arXiv:2509.16765.

[93] Fagun Patel, Sang T. Truong, Duc Q. Nguyen, Kazunori Fukuhara, Benjamin Domingue, Sanmi Koyejo, and Nick Haber. A dataset for modeling iterative problem-solving. *arXiv preprint arXiv:2609.00940; Findings of EMNLP 2026*, 2026.

[94] Deepak Pathak, Pulkit Agrawal, Alexei A. Efros, and Trevor Darrell. Curiosity-driven exploration by self-supervised prediction. In *Proceedings of the 34th International Conference on Machine Learning (ICML)*, 2017.

[95] Chris Piech, Jonathan Bassen, Jonathan Huang, Surya Ganguli, Mehran Sahami, Leonidas Guibas, and Jascha Sohl-Dickstein. Deep knowledge tracing. In *Advances in Neural Information Processing Systems (NeurIPS)*, 2015.

[96] Ulyana Piterbarg, Lerrel Pinto, and Rob Fergus. Training language models on synthetic edit sequences improves code synthesis. *arXiv preprint arXiv:2410.02749*, 2024.

[97] Michael Polanyi. *The Tacit Dimension*. University of Chicago Press, 1966.

[98] Rémy Portelas, Cédric Colas, Katja Hofmann, and Pierre-Yves Oudeyer. Teacher algorithms for curriculum learning of deep RL in continuously parameterized environments. In *Conference on Robot Learning (CoRL)*, 2020.

[99] Nasim Rahaman, Martin Weiss, Manuel Wüthrich, Yoshua Bengio, Li Erran Li, Chris Pal, and Bernhard Schölkopf. Language models can reduce asymmetry in information markets. *arXiv preprint arXiv:2403.14443*, 2024.

[100] Vipul Raheja, Dhruv Kumar, Ryan Koo, and Dongyeop Kang. CoEdIT: Text editing by task-specific instruction tuning. *arXiv preprint arXiv:2305.09857; Findings of EMNLP 2023*, 2023.

[101] Hans Reichenbach. *Experience and Prediction: An Analysis of the Foundations and the Structure of Knowledge*. University of Chicago Press, 1938.

[102] Alexis Ross et al. Modeling student learning with 3.8 million program traces. *arXiv preprint arXiv:2510.05056; Artificial Intelligence in Education (AIED)*, 2025.

[103] Stéphane Ross, Geoffrey J. Gordon, and J. Andrew Bagnell. A reduction of imitation learning and structured prediction to no-regret online learning. In *Proceedings of the 14th International Conference on Artificial Intelligence and Statistics (AISTATS)*, 2011.

[104] Megumi Sano, Julian De Freitas, Nick Haber, and Daniel Yamins. Learning in social environments with curious neural agents. In *Proceedings of the 42nd Annual Conference of the Cognitive Science Society*, 2020.

[105] Jürgen Schmidhuber. Formal theory of creativity, fun, and intrinsic motivation (1990–2010). *IEEE Transactions on Autonomous Mental Development*, 2(3):230–247, 2010.

[106] Wolfram Schultz, Peter Dayan, and P. Read Montague. A neural substrate of prediction and reward. *Science*, 275(5306):1593–1599, 1997.

[107] Burr Settles. Active learning literature survey. Technical Report 1648, University of Wisconsin–Madison, 2009.

[108] Omar Shaikh, Shardul Sapkota, Shan Rizvi, Eric Horvitz, Joon Sung Park, Diyi Yang, and Michael S. Bernstein. Creating general user models from computer use. *arXiv preprint arXiv:2505.10831*, 2025.

[109] Zhihong Shao, Peiyi Wang, Qihao Zhu, Runxin Xu, Junxiao Song, Xiao Bi, Haowei Zhang, Mingchuan Zhang, Y. K. Li, Y. Wu, and Daya Guo. DeepSeekMath: Pushing the limits of mathematical reasoning in open language models. *arXiv preprint arXiv:2402.03300*, 2024.

[110] David Silver and Richard S. Sutton. Welcome to the era of experience. *In: Designing an Intelligence, MIT Press*, 2025.

[111] Tanmay Sinha and Manu Kapur. When problem solving followed by instruction works: Evidence for productive failure. *Review of Educational Research*, 91(5):761–798, 2021.

[112] Joar Skalse, Lucy Farnik, Sumeet Ramesh Motwani, Erik Jenner, Adam Gleave, and Alessandro Abate. Invariance in policy optimisation and partial identifiability in reward learning. *arXiv preprint arXiv:2203.07475*, 2022.

[113] Kenneth O. Stanley and Joel Lehman. *Why Greatness Cannot Be Planned: The Myth of the Objective*. Springer, 2015.

[114] Matt Stephenson, Andrew Miller, Xyn Sun, Bhargav Annem, and Rohan Parikh. NDAI: Disclosure without expropriation in trusted execution environments. *arXiv preprint arXiv:2502.07924*, 2025.

[115] Richard S. Sutton and Andrew G. Barto. *Reinforcement Learning: An Introduction*. MIT Press, 2nd edition, 2018.

[116] Michael Tomasello. *The Cultural Origins of Human Cognition*. Harvard University Press, 1999.

[117] Raymond Tsao, Andrew Wagenmaker, and Sergey Levine. Learning process rewards via success visitation matching for efficient RL. *arXiv preprint arXiv:2606.23640*, 2026. UC Berkeley.

[118] TypeSafe AI. Introducing system one models and Jev. https://typesafe.ai/blog/introducing-system-one-models-and-jev, 2026. Company disclosure, September 2026. Architecture and training data undisclosed; benchmark figures self-reported on the vendor's own workflows.

[119] Jonathan Uesato, Nate Kushman, Ramana Kumar, Francis Song, Noah Siegel, Lisa Wang, Antonia Creswell, Geoffrey Irving, and Irina Higgins. Solving math word problems with process- and outcome-based feedback. *arXiv preprint arXiv:2211.14275*, 2022.

[120] Lev S. Vygotsky. *Mind in Society: The Development of Higher Psychological Processes*. Harvard University Press, 1978.

[121] Rui Wang, Joel Lehman, Jeff Clune, and Kenneth O. Stanley. Paired open-ended trailblazer (POET): Endlessly generating increasingly complex and diverse learning environments and their solutions. *arXiv preprint arXiv:1901.01753*, 2019.

[122] Eamon Worden, Cristina Heffernan, Neil Heffernan, and Shashank Sonkar. FoundationalASSIST: An educational dataset for foundational knowledge tracing and pedagogical grounding of LLMs. *arXiv preprint arXiv:2602.00070*, 2026.

[123] Fang Wu et al. The invisible leash: Why RLVR may not escape its origin. *arXiv preprint arXiv:2507.14843*, 2025.

[124] Siwei Wu, Jincheng Ren, Yizhi Li, Haau-Sing Li, Chengran Yang, Yuxuan Zhang, Weicheng Gu, Jian Yang, Riza Batista-Navarro, Chuanyi Zhang, Xianglong Liu, Ming Zhou, Bryan Dai, and Chenghua Lin. ModularRSI: Modular and generalizable recursive harness self-improvement. *arXiv preprint arXiv:2609.14857*, 2026. Beihang, Manchester, IQuest Research, M-A-P, Langboat, Hohai.

[125] Violet Xiang, Chase Blagden, Rafael Rafailov, Nathan Lile, Sang Truong, Chelsea Finn, and Nick Haber. Just enough thinking: Efficient reasoning with adaptive length penalties reinforcement learning. *arXiv preprint arXiv:2506.05256*, 2025.

[126] Violet Xiang, Charlie Snell, Kanishk Gandhi, Alon Albalak, Anikait Singh, Chase Blagden, Duy Phung, Rafael Rafailov, Nathan Lile, Dakota Mahan, Louis Castricato, Jan-Philipp Fränken, Nick Haber, and Chelsea Finn. Towards system 2 reasoning in LLMs: Learning how to think with meta chain-of-thought. *arXiv preprint arXiv:2501.04682*, 2025.

[127] Violet Xiang, Amrith Setlur, Chase Blagden, Nick Haber, and Aviral Kumar. ExpRL: Exploratory RL for LLM mid-training. *arXiv preprint arXiv:2606.17024*, 2026.

[128] Wei Xiao, Weiliang Tang, Yuying Ge, Hui Zhou, Yao Mu, Li Zhang, and Yixiao Ge. ROVE: Unlocking human interventions for humanoid manipulation via reinforcement learning. *arXiv preprint arXiv:2606.17011*, 2026. XPENG Robotics, Fudan, CUHK, Shanghai Jiao Tong.

[129] Haoyang Yan, Min-Le Su, Hangfan Zhang, Zhanhao Li, Chen Zhang, Shao Zhang, Yang Chen, Lei Bai, and Shuyue Hu. Harness-of-harness: Multi-day autonomous software development with continual improvement. *arXiv preprint arXiv:2609.01481*, 2026. Shanghai Artificial Intelligence Laboratory.

[130] Jiarui Yan, Weiwei Sun, Sijie Li, Wenhan Li, and Yiming Yang. TraceML: What auto-research agents miss in long-horizon ML development. In *Advances in Neural Information Processing Systems 40 (NeurIPS 2026), Datasets and Benchmarks Track*, 2026. Carnegie Mellon University. Corpus at huggingface.co/datasets/jerryyan/TraceML, CC BY 4.0.

[131] Qiying Yu et al. DAPO: An open-source LLM reinforcement learning system at scale. *arXiv preprint arXiv:2503.14476*, 2025.

[132] Yang Yue, Zhiqi Chen, Rui Lu, Andrew Zhao, Zhaokai Wang, Yang Yue, Shiji Song, and Gao Huang. Does reinforcement learning really incentivize reasoning capacity in LLMs beyond the base model? *arXiv preprint arXiv:2504.13837; NeurIPS 2025*, 2025.

[133] Eric Zelikman, Yuhuai Wu, Jesse Mu, and Noah D. Goodman. STaR: Bootstrapping reasoning with reasoning. In *Advances in Neural Information Processing Systems (NeurIPS) 35*, 2022. arXiv:2203.14465.

[134] Eric Zelikman, Georges Harik, Yijia Shao, Varuna Jayasiri, Nick Haber, and Noah D. Goodman. Quiet-STaR: Language models can teach themselves to think before speaking. *arXiv preprint arXiv:2403.09629*, 2024.

[135] Zhenru Zhang, Chujie Zheng, Yangzhen Wu, Beichen Zhang, Runji Lin, Bowen Yu, Dayiheng Liu, Jingren Zhou, and Junyang Lin. The lessons of developing process reward models in mathematical reasoning. *arXiv preprint arXiv:2501.07301; Findings of ACL 2025*, 2025.

[136] Tong Zheng, Xidong Wu, Zheng Zhang, Zhankui He, Chaoyi Zhang, Benjamin Coleman, Ruoqiao Wei, Di Bai, Haolin Liu, Rui Liu, Xue Wang, Yue Zhuan, Wang-Cheng Kang, Renkai Xiang, Heng Huang, Xinwu Cheng, and Yunsong Guo. Dream-RSI: Recursive self-improvement through evolving worlds. *Technical report, Google and Google DeepMind*, 2026. dream-rsi.com.

A Vocabulary

Developmental intelligence
: The form. The frontier trains on what intelligence produced; developmental intelligence trains on how intelligence develops. Pretraining is phylogenetic; this is ontogenetic.

Mastery model
: The category. A learned model of how competence develops, the learner-side counterpart to a world model. Formally the triple $(V_\theta, \pi^\star_\theta, c_\theta)$ of Section 5.

Improvement trace
: The substrate. Every wrong version, revision, failure, and correction, timestamped, across many solvers and domains. Manufactured under consent, not scraped.

Mastery advantage
: $A_\theta(h_t, a_t) = V_\theta(h_{t+1}) - V_\theta(h_t)$. Dense, signed, reference-free, and potential-based.

Learnability frontier
: The band of tasks where an attempt carries maximal information about the solver's competence; the zone of proximal development in formal terms, and the complement of the zero-gradient regions of group-relative RL.

Mastery evaluations
: Trajectory-level measures of a learner: improvement rate, retention, transfer, persistence, and long-horizon degradation, validated against human learning outcomes.

B The Organization

The program is pursued by a developmental intelligence laboratory whose artifact is the mastery model. Five seats. Two are filled: a chief executive owning the thesis, the science of learning, and the evaluations, and a chief technology officer owning ML systems and the mastery model inside the training loop. Three are open: a chief scientist for post-training owning the model and mastery-optimized RL, a head of substrate owning manufacturing, and a chief systems officer owning substrate manufacturing at scale, the embodied channel, deployment, and operations.

The chief scientist for post-training is the critical path, because that seat owns all four lethal hypotheses and Hypothesis 11. The non-obvious requirement for it is production experience of a learned reward model being gamed, not only publications about it. Hypothesis 4 states that no frontier laboratory in-houses this within twelve months, and that hypothesis depends on moving faster than a better-resourced competitor, so staffing is the rate limit and we state it rather than leave it inferred. Commercially, the mastery model, the objective, and the RL recipe are not licensed; products built on the model's outputs, trace corpora, derived environments, capture tooling, and the evaluations may be offered on non-exclusive terms. Licensing the model would transfer the one capability the thesis says the frontier lacks.

C Source Tiers

Claims are drawn from three kinds of source. Tier 1: primary (paper, dataset, company disclosure). Tier 2: credible reporting or named research groups. Tier 3: aggregation or inference. Figures on frontier laboratory environment spending, per-task compute cost, and vendor pricing are Tier 2 reporting and vendor models rather than measured spend, and are cited as orders of magnitude.