Efficient Thinking: Measuring What Capability Costs

Efficient Thinking VIII: Experience Priors

Efficient Thinking VIII: Experience Priors

The constraint and the carrier — how verified history moves a frozen model's quality–compute frontier

> ***STATE 2026-09-20: MEASURED, closed.*** *Every result below is on disk under* `experience/results/` *with its command; the pre-registered readings are scored in §10; the closing runs landed 2026-09-19 to 09-20 and nothing further enters this paper. First paper of the VIII line (working label 8a); VIII-b accumulation, VIII-c carrier studies and VIII-d instruments follow (* `et8-series-plan.md` *).*

Louay Alsakka · September 20, 2026 · *v1.0*

Abstract

An experience prior is a component that makes a frozen model's search cheaper without making the model less capable. We define it by that constraint rather than by any implementation: minimise search cost subject to capability held within a pre-registered margin, measured by an external verifier, paired per task slice. Under that definition we tested six carriers of experience in two fields with no code in common — a text memory injected into the prompt, a steering vector, an additive logit bias, two forms of a hand-written localisation rule, and a learned move prior over a chess search — and none lowered cost with capability held. One mechanism did: a read-only linear head that reads the frozen model's own hidden state after one observation and chooses among the model's own candidate actions. On a task set where each episode is a distinct problem it solves 11 to 15 more problems in a hundred, on independent problems, twice, against a bar fixed before the head was fitted, with 95% intervals that exclude zero. Its cost is stated three ways and the pre-registered one named: per episode in actions, the pre-registered measure, it is cheaper; per episode in tokens with the controller's

own inference counted, it is 34% dearer; per problem solved it is 21% cheaper, a metric chosen after the fact and labelled so. Whether the gain is the experience or the extra compute is decided by a matched control: the frozen model alone, given the head's compute to within half a percent, reaches 22.3% where the head reaches 31.7% (+9.3, interval [+3.7, +15.0]). At equal compute the head still wins, so under the general criterion of an experience prior — improving the model's quality–cost frontier — it is one: the same frozen intelligence, having learned from its own history, thinks more effectively per unit of computation. On the efficiency limb the head is compute-neutral at budget 8 and solves 8.3 more problems in a hundred (interval [+3.0, +14.0]); both limbs hold on this task set. As a curve, base at six budgets and head at four across 4.4 to 13.9 thousand tokens, the head sits above the base throughout, and at equal quality — 24.0% both, paired, 33 discordant each way — the head spends 2.32× less. At the same decision over the same candidates, the model's own preference reaches 20.3% and a five-sample majority vote 12.3%, below the base itself; the head's 31.7% is the readout fitted on verified history, not the inference spent. On a second search structure — SQL query repair, clauses as regions, a result-set verifier, the head re-fitted from that environment's own history — the same mechanism gives +12.3 points on a disjoint 300 (interval [+6.7, +18.0]), so the claim is experience-directed search on two structures, not one grammar. The result is bounded by a fact about the task, not the mechanism: the prior helps only where the thing it improves — here, localisation — is what limits the agent; on a repair-bound set the same head does nothing. Two further findings shape what follows. The largest effect of the study was not a prior but a change to *when* the agent may act: requiring one observation before the first guess doubled problems solved and cut cost, with no memory, vector, head or training. And accumulation — the claim that experience compounds across generations — cannot be tested at a decision point whose inputs are fixed before the prior acts, because there the generations can differ only in their target; it needs a loop in which the prior's earlier choices shape the states it later reads, and that is the subject of Paper VIII-b.

Results at a glance

finding	measurement	where
an experience prior is a constraint, and the constraint is one limb of a frontier	cost down with capability held within a pre-registered ε, paired per slice; generally, $Q_E(C) > Q_0(C)$	§3
six carriers of experience fail the constraint	text memory, steering vector, logit bias, two localisation rules, a chess move prior — each fails on capability or on cost	§7.3
a read-only head over the frozen model's hidden state	+11.3, +13.0, +14.0, +13.3 problems in 100 across four n	§7.4

finding	measurement	where
passes	= 300 measurements, all $p < 10^{-4}$	
it replicates in a second model family	Llama-3.1-8B: +20.0 held-out, +14.3 [+7.7, +21.0] on a disjoint 300	§7.4
the gain is not bought with compute	the base given the head's compute: 22.3% against 31.7% (+9.3 [+3.7, +15.0], p 0.002)	§7.6
both limbs of the frontier hold	compute-neutral at budget 8: +8.3 [+3.0, +14.0]; equal quality (24.0% both, 33 discordant each way) at 2.32× less compute	§7.6
the head sits above the base across the measured range	six base and four head budgets over 4.4k–13.9k tokens; the base plateaus from 9.2k	§7.6
the strongest simple baselines at the same decision do not reach it	the model's own preference 20.3%; five-sample majority vote 12.3%, worse than the base; head 31.7%	§7.7
the mechanism transfers to a second search structure	SQL repair, head re-fitted: probe 89.3% vs 20.9% chance; +12.3 [+6.7, +18.0] on a disjoint 300, p 6×10^{-5}	§7.7
the bound	the prior helps only where what it improves — localisation — limits the agent; on a repair-bound set the same head does nothing	§7.5
accumulation cannot be tested at one decision	a second generation's training states were byte-identical to the first's; P9 fails by the letter	§9
two carrier studies	chess prior +20 ± 55 Elo against +182 for search; a poet's LoRA carries the style and not the person	§8

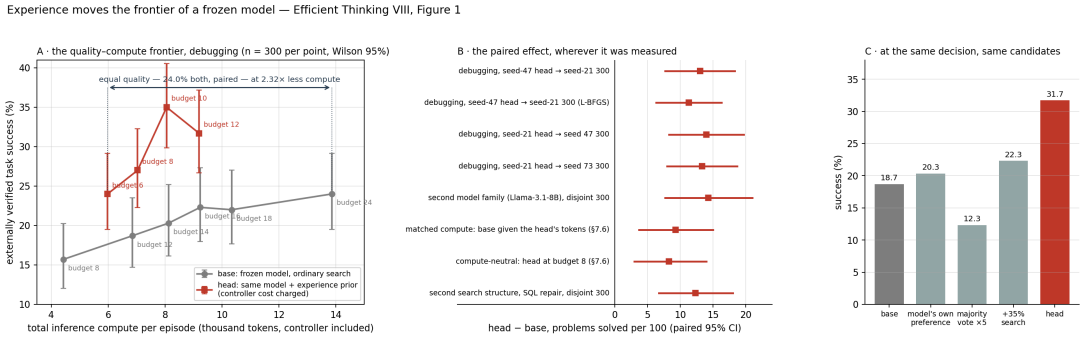

Figure 1

Figure 1. ***A.*** *The quality–compute frontier on the debugging family: the frozen model under ordinary search at six action budgets, and the same model with the read-only head at four, every head point charged the controller's measured cost; bars are 95% Wilson intervals per point, the paired tests are in §7.6. At equal quality (24.0% both) the head spends 2.32× less.* ***B.*** *The paired difference, head minus base, with its 95% interval wherever it was measured: four cross-run debugging measurements, a second model family, the two matched-compute limbs, and the second search structure with the head re-fitted.* ***C.*** *At the same decision over the same enumerated candidates: the model's own highest-probability candidate and a five-sample majority vote do not reach the head; the vote is worse than the base (§7.7). Source:* `experience/results/r5c.json`, `r1_intervals.json`, `r4b_disjoint.json`, `r7_disjoint_stats.json`, `r6.json`; *script* `scripts/fig1_frontier.py`.

1. Motivation

Two quantities are usually run together when an agent's competence is discussed: what the underlying model can do, and what it has learned from doing it. The first is intelligence in the sense of a frozen network's capability on a task it has never seen. The second is experience: the procedural knowledge that a given kind of problem is usually solved by looking in a particular place first, that a certain branch rarely pays, that one observation is worth several guesses. Continual learning proposals tend to change the first in order to obtain the second, and pay for it with the corruption they were trying to avoid: an adapter that fits the loss and loses the capability, a memory that fills the context and degrades the behaviour it was meant to guide.

This paper asks a narrower question than "can an agent learn from experience". It asks what a component would have to satisfy to count as experience at all, builds the instruments that can tell whether a candidate satisfies it, and reports what six candidates did under those instruments in two fields.

2. Intelligence and Experience Are Different Variables

Let π_0 be the search policy of a frozen model on a family of tasks, and let $Q(\pi)$ be the fraction of tasks it solves, judged by an external verifier — unit tests, a rule table, an engine — never by the training loss. Let $C(\pi)$ be the cost of the search: actions per episode, tokens, or simulations. Intelligence is $Q(\pi_0)$. Experience is anything that lowers C without lowering Q. The two are separable in principle because search cost is dominated by choices the model makes before it has committed to an answer — where to look, which branch to try, when to stop — and those choices can be reordered without touching the machinery that produces the answer once the place is found.

3. The Definition: an Experience Prior Is a Constraint

An experience prior is not defined by its implementation but by what it is permitted to do to the frozen model. With π_ϕ the same model with the prior applied:

$$\min_\phi C(\pi_\phi) \quad \text{subject to} \quad Q(\pi_\phi) \geq Q(\pi_0) - \epsilon$$

Three rules make this measurable rather than rhetorical, and every result below was read under them.

1. ϵ is declared before the run, not chosen after it.
2. C and Q are measured on the same held-out tasks, paired per slice, and the constraint must hold on the slices, not on a pooled mean.
3. The verifier ships with its reach: the fraction of positions or cases it could actually read. A verifier that silently skips what it cannot parse inflates Q. Verify the learner, and verify the verifier.

The consequence is sharp. A mechanism that lowers cost by lowering capability is not a weaker prior; it is not a prior. Every failure in §7 is a failure of the constraint, not of the objective. The verification gate of the lifecycle (§5) is where the constraint is enforced before anything consolidates, and the regulariser of the training objective is its relaxed form.

The general criterion, of which the above is one limb. The constraint as written asks for the same capability at lower cost. An experienced engineer shows experience in two equivalent ways: given the same hour, she solves a problem the novice cannot, or given the same problem, she solves it sooner. Both are one fact about the frontier of capability against cost. The general definition is therefore that an experience prior improves the frozen model's quality–cost frontier, $Q_E(C) > Q_0(C)$ over a meaningful range of C, with two manifestations:

$$\text{efficiency: } C_E(Q^*) < C_0(Q^*) \qquad \text{effectiveness: } Q_E(C^*) > Q_0(C^*)$$

Every experiment in this paper was pre-registered under the first limb and is reported exactly as it was run; the frontier definition is a revision of the theory made after those results, stated here so that §7's cost finding is read correctly. Under the first limb the read-only head of §7.4 fails on tokens: it solves more problems and spends 34% more compute per episode doing so. Under the frontier it is undecided, because the two points compared sit at different compute, and the control that decides it — the frozen model alone given the head's compute budget, and the head's minimum compute at the model's own success rate — is the pre-registered addendum in §7.6. This is also the series' common objective stated once: efficient thinking is more useful capability per unit of computation, and search (Efficient Thinking I) and experience (this paper) are two ways of moving along or moving the same curve.

4. Where a Prior Can Act

Before asking which carrier of experience satisfies §3, one has to ask where in an episode a prior can act at all. The answer is a structural fact that two of the study's task sets were needed to make visible.

At a decision point before any observation, a prior over the state is a prior over the prompt. The first decision of an episode is made on the symptom and nothing else; two tasks with the same symptom present byte-identical prompts, identical activations at every layer, and therefore identical output from anything that reads them. A head fitted at that point is a lookup table on the symptom string — exactly, not approximately — and text memory injected there is the same table in another form. On the study's second task set the 80 evaluation episodes carried six distinct first-decision prompts.

A prior over search can act only where the search has observed something, and whether it has can be told by counting, before anything is built (§6). The harness used for every positive result in this paper therefore requires one inspection before the first hypothesis, and the prior acts on the hypothesis that follows it.

Where the prior's inputs are fixed before it acts, generations cannot differ in input. If the observation that precedes the prior's decision is made by the frozen model on a prompt the prior does not touch, then a second generation trained on a better agent's episodes trains on the same states as the first; it can differ only in its target. This closes the accumulation question on the present harness (§9) and opens Paper VIII-b.

5. The Lifecycle, Compressed

Experience is detected in episodes, distilled into candidates, verified against held-out replay, and consolidated only if it passes. The one part of this lifecycle the study could test in isolation is the gate. Twelve lessons were distilled from a 7B model's own successful episodes — all of them true in the sense that the region they named was the region the fix landed in, at

confidences above 0.99 — and each was replayed on twenty held-out episodes in its own condition and in a control condition. Eleven of twelve were rejected: five because, replayed, the lesson did nothing (gain 0.0), two because the control gained exactly as much as the treatment, and the rest because they lost ground. The one admitted was the only lesson whose content was negative — avoid a region — where every rejected lesson was a positive "start here". Twelve is a small set and eleven of eleven in one direction is stated as a pattern, not a law; but the gate did the work assigned to it, and it rejected lessons that were true. Carrier harm, not wrong advice, is what it caught.

6. The Instruments

Every number in §7 and §8 passed through the following, in this order. Each was built because a result had already misled us without it (Appendix A).

- **The task gate.** Count the distinct problems in a task set as the agent sees them — program, tests, symptom, regions — excluding identifiers and seeds. Below nine-tenths of the file count, the set is a handful of problems wearing many names, and every rate on it is a weighted count of a few deterministic outcomes. The first task set read 12 problems in 2,000 files; the second, 11.
- **The prompt gate.** At the decision point the prior will act on, count distinct decision prompts against episodes. Below nine-tenths, the state is a table and no head is fitted. Necessary, not sufficient: prompts can differ by a counter and carry no signal.
- **The probe controls, three, always together.** A linear probe on the hidden state, held out by task, reported beside (i) the same fit with labels permuted, five times; (ii) the same fit after projection to eight dimensions; (iii) the same fit on a hundred examples, five draws. A learned readout degrades under (ii) and (iii); a key copied from the prompt does not. A probe that passes (i) alone proves nothing — the first task set's head scored 100% held out and passed its permutation control, and was a six-row table.
- **The fit has no free parameter.** Heads are fitted by a second-order method; a probe accuracy that moves with an optimizer's step size (56%, 59%, 24% on the same states) is not a measurement of the state.
- **Verify the verifier.** Every rate ships with the fraction of cases its instrument could read. A tone table that read the model's character set as unknown and skipped the rule inflated a form score by 18 points; a rhyme rule scored a famous ancient-style series as broken verse until two poets from two editions collapsed on one form, which accuses the instrument.
- **Verify the sample.** Every generated artefact is checked against the training set and a canon before a judge sees it. A fine-tune that regurgitates its training data fools a judge honestly (§8).
- **Persist the trajectories.** A run keeps the state and the action of every decision it may later be distilled from, or the claims downstream of it are marked unauditable. Three artefacts a later question needed were not the artefacts the runs had been written to save.

7. Results in the Debugging Field

The claims in this section form a ladder, each rung harder than the last, and the paper says which rungs it has climbed: the head works on one task set (§7.4); it replicates across independent problem sets (§7.4, four measurements at n = 300); it replicates across model families (the second model, below); it beats ordinary search at matched compute (§7.6, the frontier control); and it survives a different search structure (the last pre-registered run, §7.7). A dozen further seeds or checkpoints would add less than any one of the last two, so none are run. The failure reading of each rung was written before its run, and the fourth's is worth restating because it is the one that would have unmade the paper: had ordinary search at the head's compute done as well as the head, the head would have been another way of buying capability with compute, and not evidence that history moves the frontier.

7.1 The environment and its three versions

The agent is a frozen 7B model with a budget of twelve actions from a family of four — hypothesise a region, inspect it, patch, run the tests — on a pipeline program with one injected bug, judged by the program's own tests. Three task sets were built. The first two failed the task gate after the fact (12 and 11 distinct problems); their results are reported as directions only, with the number of problems printed, and no magnitude from them is carried. The third is generated from a program grammar with tests derived from the clean program's behaviour and bugs admitted only if the verifier distinguishes them; it reads 300 distinct problems in 300 files, a held-out lookup table on the symptom localises 6.7%, and 257 of 300 base episodes take distinct trajectories. It is the first set on which an episode is an observation, and every interval below is from it.

7.2 The base, and the loop

On the third set under the inspect-first loop the base solves 56 of 300 (18.7%, SE 2.2) in 10.63 actions, with the first hypothesis in the bug region 29.0% of the time. Two independent draws of 75 problems land on the same 17.3%, so the rate is a property of the generator and not of a seed.

The loop itself is a result. On the second task set, requiring one inspection before the first hypothesis took problems solved from 2 of 11 to 4 of 11 and mean actions from 10.2 to 8.8, with a mandatory extra action spent — because failures burn the whole budget and successes finish in three steps, solving more is what makes the average cheaper. All 80 episodes complied and in one slice nine of twenty chose to inspect a second time. That is the shape the constraint was written to detect, produced with no memory, no vector, no head and no training, by hand-writing the one rule a controller is meant to learn: look before you guess. It is evidence about the harness the priors were measured in, and every prior below was measured inside it.

7.3 Six carriers against the constraint

carrier	field	task set (n)	direction	constraint
A · text memory from the model's own successes, injected into the prompt	debugging	first (12)	success falls on every problem; input tokens ×3–4; the better memory falls further	fails, capability
C · steering vector along the productive-vs-wasted direction, layer 27, α = 4	debugging	first (12)	success falls; localisation flat	fails, capability
F · additive logit bias on emittable action tokens from productive/wasted frequencies	debugging	first (12)	capability held; cost not reduced	fails, cost
naive rule · always hypothesise the symptom region (an oracle on region and bug class)	debugging	second (11), third (300)	on the second set, aim up and outcome down; on the third, +10 points of localisation and one episode	fails, cost
naive rule with the model's own words for the forced region	debugging	second (11)	lands exactly on the base	fails, cost
learned move prior over MCTS, β chosen by rule	chess	400 games	+0.03 ± 0.08 win rate at every simulation count; search itself worth +0.13–0.15 per doubling	fails, cost

Two lines from this table are the sentences the study keeps. **The damage a naive prior does scales with how good the chooser under it is**: on the blind loop, forcing the correct region cost about one episode, because the agent was guessing and overriding a guess

costs little; on the observing loop the same rule, on the same set, solved the same problems less reliably at higher cost while localising better. And **localisation is never a bar**: every arm that aimed better and did worse would have passed one.

7.4 The mechanism that passes: a read-only controller head

The steering result separates two things: the frozen model's hidden state at a decision point carries the signal that separates productive from wasted search, and writing along that direction into the residual stream lowers success. The head uses the signal without writing. At the post-inspect decision it enumerates candidate actions — the emittable family crossed with every region visible, each teacher-forced as a prefix, the model's greedy choice always among them — reads the hidden state at each candidate's last token (layer 18), scores it with a linear head trained on the verifier's labels from the base agent's own episodes, and takes the argmax. The model's parameters and residual stream are never touched; the model writes every action it takes; the only thing the head can get wrong is the choice.

Candidates are enumerated rather than sampled because sampling offered one region across eight draws at 40% of decision points and raised its ceiling to 94–96% at most; enumeration's ceiling is 100% by construction.

The probe, with its controls. On 75 decisions held out by task: head 66.7%, the agent's own pick 33.3%, chance 20.4%; permuted labels 10.7–24.0%; eight dimensions 26.7%; a hundred examples 37.9%. On the second seed's pool: 42.7% against 20.0% and 20.1%; permuted 13.3–24.0%; 33.3%; 31.2%. The signal degrades when dimensions and examples are removed, which is what a learned readout looks like.

The bar, fixed before the head was fitted: success at least 16.5% on the four slices, and either at least 23.1% at no higher cost or the same number solved at least one action cheaper.

The result.

head → evaluation set	base	G	delta
first seed's head → first seed's held-out slice (same generator run)	17.3%, 10.71	38.7%, 9.17	+21.4
first seed's head → second seed, independent (75)	17.3%, 10.68	29.3%, 9.81	+12.0
second seed's head → first seed's 300	18.7%, 10.63	31.7%, 9.85	+13.0

head → evaluation set	base	G	delta
second seed's converged head → first seed's 300	18.7%, 10.63	30.0%, 9.92	+11.3
first seed's converged head → second seed (75)	17.3%, 10.68	32.0%, 9.76	+14.7

With paired per-problem bootstrap intervals (95%) and McNemar on success:

pair	n	success	actions	McNemar
first seed's head → first seed's held-out slice (same run)	75	+21.3 [+9.3, +33.3]	−1.53 [−2.43, −0.63]	0.0025
first seed's head → second seed	75	+12.0 [+1.3, +22.7]	−0.87 [−1.63, −0.15]	0.064
second seed's head → first seed's 300	300	+13.0 [+7.7, +18.3]	−0.78 [−1.15, −0.43]	7×10^{-6}
second seed's converged head → first seed's 300	300	+11.3 [+6.3, +16.3]	−0.71 [−1.06, −0.37]	3×10^{-5}
first seed's converged head → second seed	75	+14.7 [+4.0, +25.3]	−0.92 [−1.63, −0.24]	0.013

Four cross-run measurements span +11.3 to +14.7 points; the direction of transfer is symmetric. The one same-run measurement is what training on adjacent tasks buys and is never averaged in. One 75-problem pair has a bootstrap interval excluding zero and a McNemar p of 0.064 on nineteen discordant pairs; it reads as consistent with the others and not independently significant, and the 300-problem pairs carry the result. Six pairs, no multiplicity adjustment; the 300s would survive one and the 75s would not all.

The larger independent evaluation then ran: the same head, never trained on either set, on two further sets of 300 sharing no signature with its training set or with each other.

set (300 each)	base	G	success	actions	McNemar
second seed	16.7%, 10.86	30.7%, 9.91	+14.0 [+8.3, +19.7]	−0.95 [−1.32, −0.58]	2×10^{-6}
third seed	18.0%, 10.65	31.3%, 9.91	+13.3 [+8.0, +18.7]	−0.74 [−1.08, −0.41]	2×10^{-6}

Four measurements at n = 300 — +11.3, +13.0, +14.0, +13.3 — across disjoint sets and both transfer directions, all with p below 10^{-4}: not a seed artefact, not a same-run artefact, not a direction artefact. The three bases are their own result: three 300-problem draws from one grammar land within 2.0 points and 0.23 actions of each other (18.7%, 16.7%, 18.0%), so the base rate is a property of the generator, which the first two task sets could never have shown. None of this touches the cost sentence above: the action figures are the pre-registered metric, which does not count the controller's own inference, and in tokens G costs 34% more, with the batched scoring above failing its bar by 26 tokens and the shared-cache accounting still to run.

A second frozen model. The same environment, harness and gates were run on a 4-bit 3B instruct model of a different family, with a floor pre-registered before the run: no probe is fitted unless the base solves at least 5% of problems, because a null from a probe with no room would read as "no signal" when it means "no room". The base solved 4 of 262, 1.5%, at 11.95 actions. It is not a format failure: the model emits well-formed actions 94% of the time, and its dominant behaviour is submitting a patch byte-identical to the code already there, 31% of steps — it understands the protocol and cannot repair. The third task set is calibrated to the 7B and out of range for a 4-bit 3B, and a reader reaching for a smaller model should know it. The run also returned something it was not asked for: the smaller model produced a truncated escape inside a patch that the 7B never had in three thousand episodes, and the harness crashed the run rather than the step; a malformed action is now scored as invalid, which the loop already knew how to do. One further model then ran under the same floor, a different family at comparable size, pre-registered as the last: a 4-bit 8B instruct model cleared it at 41 of 300, 13.7%, in 11.24 actions, and is a capable model rather than a lucky one — its dominant action is inspection, invalid actions are 0.1% of steps, and the byte-identical resubmission that dominated the 3B is 16%. The probe on its own post-inspect states, held out by task with every draw printed: head 56.0%, the model's own pick 16.0%, chance 20.4%; permuted labels 9.3 to 26.7%; eight dimensions 33.3%; a hundred examples 34.4%. Layer 27 scored 1.3 points higher and layer 18 was wired, because that is the layer the whole package used and switching on the answer would make every earlier number non-comparable. G, on the held-out quarter the fit never saw, paired per problem, n = 75: success 16.0% to 36.0%, +20.0 with interval [+8.0, +33.3]; actions 11.12 to 9.48, −1.64 with interval [−2.53, −0.77]; McNemar p = 0.0059. **It is consistent with the 7B, not bigger than it**: the interval contains all four of the 7B's measurements, and no difference between families is claimed or can be claimed at this n. On

both frozen models the head adds about twenty points to the share of hypotheses that name the bug's region — 18.2% to 38.1% on the 8B, 33.3% to 55.3% on the 7B — a different quantity from the head's accuracy at the one decision it touches, since every later hypothesis is the model's own. What each model then does with a better hypothesis differs: the 7B stops submitting patches identical to the code in front of it (84% of its saving) and inspects slightly more; the 8B stops re-inspecting. The totals are close; the routes are not. Descriptive, no test, n = 75 per arm. The disjoint-set leg then ran on a second seed's 300 problems the head had seen nothing of, base first and then G, both arms complete: success 18.3% to 32.7%, +14.3 with interval [+7.7, +21.0]; actions 10.98 to 9.62, −1.36 with interval [−1.85, −0.88]; McNemar $p = 4.7\times10^{-5}$. Two frozen models from different families, transferring to a disjoint problem set, agree to within a point (the 7B's four: +11.3, +13.0, +14.0, +13.3), and the mechanism replicates (region-hit 21.2% to 38.2%). The held-out +20.0 and this +14.3 are the same result, not two — the smaller interval contains the larger's estimate — and the 300-problem figure is the one quoted. The reviewer's condition is met at the level it was asked: the phenomenon is not peculiar to one checkpoint of one family. On the 300, the head agreed with the agent's own pick on 121 decisions and changed 179: on the 179 the base solved 12 and G solved 52; on the 121, 44 against 43. The effect is entirely in the decisions the head changed, and the residual confound from the harness's exploration temperature is bounded at about a third of an episode per hundred.

What fit quality buys. A head that is better by every probe criterion solved five fewer problems in three hundred, inside noise, in one direction on one set and the other on the other. Two heads with identical probe accuracy differed in weights and in how many picks they changed, and the one that intervened more landed lower. A head good enough to beat the agent's own pick is good enough; probe accuracy predicted the episode difference in neither direction, four times.

What it costs. The mean-actions column does not count the controller's own inference: at each decision the head scores 4.9 candidate prefixes on average, each a 480-token prompt, 2,353 tokens per episode. Per episode the base spends 6,854 tokens and G 9,170, +34%, and the wall clock had said so — G's episodes take 0.61 s longer despite fewer actions. Three cost measures are therefore printed and the pre-registered one is named. Per episode in actions, the pre-registered measure: G is cheaper, −0.71 [−1.06, −0.37]. Per episode in tokens with the controller counted: G is 34% dearer. Per problem solved, in tokens: the base spends about 36,650 per solution and G about 28,930, 21% less — a metric chosen after the fact, and labelled so. The honest sentence is that G buys problems solved at a 34% compute premium per episode and does not, as measured, buy them cheaply; under the definition of §3 with cost in tokens it is not yet an experience prior but a better allocation of more compute. The candidate prompts differ only in their final region token, so a batched pass over a shared cached prefix would remove most of the 2,353 tokens; that is an engineering change to a system that has not been measured, it is pre-registered with the bar that the overhead fall below 502 tokens per episode — the 0.71 actions saved at the measured marginal cost of an action, 707 tokens by the slope of tokens on actions — with the budget-based and mean-based

readings (406 and 458) printed beside it. Built and measured, the batched system costs 528.4 tokens per episode: the shared prefix once, 492.3, plus 36.1 of continuations, against 2,487.9 for the separate passes and against the bar of 502. It fails, by 26 tokens, on all three bases, and the reading pre-written for that case applies: as measured, G buys success at a compute premium. A first implementation had cleared the bar at 498.7 and was wrong twice — it ignored its layer argument and read one position early — and was found by isolation rather than by a fourth guess: a correct cached split agreed with it to zero, so the fault was in what was read, not how. The structural fact under the number is that the decision prompt alone is 492 tokens, 98% of the bar, so no batching scheme clears it. That prompt is one the agent prefills anyway in order to act; a controller that reads its state from the agent's own pass pays only the continuations. That is the accounting the measured system got wrong rather than a rescue, and it is the last pre-registered item of the package: the head over the agent's own cache, counting only tokens the agent would not otherwise pay, bar unchanged, with the expectation written first that it costs 36 to 180 tokens and passes — or, if the state cannot be shared with the agent's pass, fails, and the premium stands. Built, its self-test was written before the run as two identities — the candidate states against the measured path, and the agent's generated text against an uncached generation — and the second failed: the states agree to the bf16 floor, but the text differs on 7 of 40 decisions. Measured rather than assumed, the divergence is a rewording of the free-text rationale after the action; the parsed action, region and bug class are identical on 40 of 40, because greedy decoding is chaotic under bf16 noise and one near-tie flips the prose. The gate as written is not relaxed after the fact: that run is void and the premium is the measured system's number. A new gate is pre-registered instead, with a control that says whether the old one could ever have passed: uncached against uncached on the same hardware, repeated, for raw-text identity, which if it also reads near 80% shows text identity is a property of the decoding and not of the cache; parsed-action identity at 300; and paired outcomes, cached against uncached on the same held-out 75, identical on success and actions, the level every number in this paper is scored at. The calibration went against the reading it might have licensed: uncached against uncached, same inputs, reads 100% text identity on 40, so identity is achievable and the 77% agreement under the shared cache is the cache, not the decoding. At 300 the parsed action agrees on 295 — the five mismatches differ only in the bug class named, the region is identical on 300 of 300 — and the gate, set as parsed-action identity including the class, fails. The third leg is moot. So the cost limb closes where it started, four attempts and one sentence: **as measured, G buys success at a compute premium.** Per episode in actions, the pre-registered measure, it is cheaper; per episode in tokens with its own inference counted, 34% dearer; per problem solved, 21% cheaper, post hoc and labelled. The benefit limb is untouched throughout. A controller whose state is read from a pass the agent makes anyway would change the accounting, and this paper does not have one that leaves the agent's decisions unchanged.

7.5 The bound

The third set's failures are localisation-bound: doubling the budget converts 6 of 65 failures, and 22.7 points of localisation produced 21.4 points of problems solved, close to one for one. The second set's failures were repair-bound: forcing correct localisation converted at 12.5% on the margin, because the agent's own correct localisations were the easy ones and the imposed ones were not. The same head is expected to do nothing on a repair-bound set, and that is what was measured. **A prior helps only where the thing it improves is what limits the agent.** The result is not that experience priors work. It is that this prior works where localisation is what limits the agent, and the six negatives of §7.3 are the other half of the same sentence.

7.6 Addendum, pre-registered: the frontier control

The cost finding compares two points at different compute: the base at (6,854 tokens, 18.7%) and the head at (9,170 tokens, about 31%). Whether the head improves the frontier or merely buys success with compute is decided by two runs written before either is made, on the same 300 problems with the same verifier.

- **Effectiveness at equal compute.** The frozen model alone, with no head, given the head's compute: its action budget raised until its mean tokens per episode is nearest 9,170, the budget chosen by that rule and not by the result, from a small ladder (14, 16, 18) measured on one slice first. Because the budget is in the prompt, this is a different agent and is reported as one. Reading: if the equally funded base solves fewer problems than the head by more than the paired interval, the head has moved the frontier — same intelligence, same compute, more solved; if it solves as many, the head bought its success with compute and the frontier is unmoved.
- **Efficiency at equal quality.** The head at reduced budgets (10, 8, 6): the smallest budget at which it still solves at least the base's 18.7%, and its tokens per episode there against the base's 6,854. Reading: fewer tokens at the base's success is the efficiency limb met; not fewer is the second half of the same answer.
- **Pre-written expectation, from one data point already held.** On 80 problems of this set, doubling the base's budget from 12 to 24 raised its success from 18.8% to 22.5% — 3.7 points for about twice the compute — while the head's 34% more compute raised it 11 to 14 points. If that holds at 300 the frontier is moved, and the sentence the paper may then use is the one it has so far refused: the same frozen intelligence, having learned from its own history, thinks more effectively per unit of computation. If it does not hold, the head is a better allocation of more compute, as §7.4 says today.

Result, effectiveness at equal compute: the frontier moved. The ladder read 8,325, 9,417 and 10,506 tokens per episode at budgets 14, 16 and 18, and the rule chose 16 as nearest the target — which matters, because budget 18 had the best first-slice success of the three, and

a rule chosen after the ladder would have been pulled toward it and turned a compute match into a search for the base's best showing. On the full 300, compute-matched to within 0.44% with the base handed slightly more:

arm	tokens/episode, all in	success	actions
base, budget 16	9,234	22.3%	13.75
G, budget 12	9,194 (6,706 model + 2,488 controller)	31.7%	9.85

G leads by +9.3 points, interval [+3.7, +15.0], McNemar p = 0.002; actions −3.90 [−4.36, −3.43]. The reading written before the run applies as written: the base at the head's compute does not reach the head, so **the head is not buying success with compute; at equal compute it wins.** The expectation written from the 80-problem probe held almost exactly — more budget bought the base +3.6 points for 35% more compute, the head bought +13.0 for 34% — and the mechanism holds a third time: even with 35% more compute the base localises worse than G (hypotheses naming the bug's region 29.0% against 47.4%); the extra budget buys more attempts, not better aim. Bounds: one point of a frontier, not a curve; each budget is a different agent, as the rule requires; the controller's cost charged to G is the measured 2,488 tokens of the unbatched system, so a cheaper controller would only widen the lead — the failures of R2b through R2d bound this result conservatively rather than weaken it. Under the general criterion of §3, then, the head is an experience prior on the effectiveness limb: the same frozen intelligence, having learned from its own history, solves more at the same computation. The efficiency limb — the head's minimum compute at the base's success — is the last run.

Result, efficiency at equal quality: the limb passes, at the row the statistics support. The head at reduced budgets, every row charged the measured controller cost, paired against the same base on the same 300:

arm	tokens/episode, all in	success	vs base
base, budget 12	6,854	18.7%	—
G, budget 6	5,979 (−12.8%)	24.0%	+5.3 [+0.3, +10.3], p 0.052
G, budget 8	7,034 (+2.6%)	27.0%	+8.3 [+3.0, +14.0], p 0.005
G, budget 10	8,054 (+17.5%)	35.0%	+16.3 [+11.0, +22.0], p 4×10^{-8}
G, budget 12	9,194 (+34%)	31.7%	+13.0

The pre-registered rule — the smallest budget at which the head still solves at least the base's 18.7% — selects budget 6, at 12.8% less compute than the base, and that is a real sentence; but its interval barely excludes zero and McNemar reads 0.052, so clearing the bar and demonstrating a difference come apart at exactly that row, and the claim is made at the next one: at budget 8 the head is compute-neutral to within 2.6% and solves 8.3 more problems in a hundred, p = 0.005. Both are printed, the cheaper point with its borderline flagged. One ordering is not printed as an ordering: budget 10 scores above the measured system's budget 12, and the paired test says the two are not separated (+3.3, interval [−1.7, +8.3]); success is flat across budgets 10 to 12, and there is no optimum to report. Both limbs of the general criterion now read on this task set: compute-matched with the base handed more, +9.3; compute-neutral, +8.3; and the same conservative bound applies to every head row, since the controller is charged at its measured, unbatched cost.

Result, the frontier as a curve. Base at six budgets and head at four, the same 300, every head row charged the measured controller cost (`experience/results/r5c.json` ; three readings written before the run, all three scored):

base budget	tokens/episode	success		head budget	tokens/episode, all in	success
8	4,424	15.7%		6	5,979	24.0%
12	6,854	18.7%		8	7,034	27.0%
14	8,121	20.3%		10	8,054	35.0%
16	9,234	22.3%		12	9,194	31.7%
18	10,339	22.0%				
24	13,858	24.0%				

The reading that applies is the first: the head sits above the base across the measured range — the four nearest-compute pairings read +5.3 [+0.3, +10.3], +8.3 [+3.0, +14.0], +14.7 [+9.3, +20.0] and +9.3 [+3.7, +15.0]. The two that did not fire are printed with it. The curves did not cross at low budget: the head's cheapest point still beats the base at budget 12 by 5.3; whether they cross below 5,979 tokens is unmeasured, and this paper does not say. The curves did not merge at high budget: the base's top, 24.0% at 13,858 tokens, equals the head's cheapest point, not its best, and the base's own last step (18 to 24, 34% more compute) reads +2.0 [−1.7, +5.7], p 0.38 — a plateau from 9.2 to 13.9 thousand tokens. The cleanest sentence the curve allows is an equal-quality one: base at budget 24 and head at budget 6 both solve 24.0% — paired on the same 300, +0.0 [−5.3, +5.3], 33 discordant pairs each way — and the head does so for 2.32× less compute (5,979 against 13,858 tokens) and 3.6× fewer actions (5.53 against 19.81). That is a ratio of two measured points, not a fitted curve; nothing here is interpolated. Scope:

one seed, one model, one task set; base budgets 8 to 24 and head budgets 6 to 12; below 5,979 tokens unmeasured. Within it, the frontier is moved over the range rather than at a point, and the base does not reach the head's best at 1.7× the head's spend.

7.7 Pre-registered: the strongest simple baselines, and a second search structure

Two runs remain before this paper stops, both written before they are made. The first answers the reviewer a matched-compute result invites — why an experience controller rather than the model's own preference at the same cost: at the same decision point, over the same enumerated candidates, the candidate the frozen model itself assigns the highest log-probability, charged the same passes the head pays; and a majority vote over five sampled hypotheses, charged its five. If either matches the head, the learned readout adds nothing over the model's own preference and the claim narrows to "a controller at the decision point"; if both fall short by more than the paired interval, experience — a head trained on verified history — is responsible, not the extra inference. The expectation is that the model's own preference lands between the base and the head, since it is what the head learned to overrule on 179 of 300 decisions. The second changes the search structure rather than the bugs: a second environment under the same harness, verifier and gates, in which the regions are the clauses of a query against a fixed schema and the verifier is the expected result set, with the head re-fitted from that environment's own base episodes so that what transfers is the mechanism and not the weights. If the head clears the same bar there, the claim is experience-directed search; if not, the paper says the mechanism is bound to the first environment's grammar and hands the question forward.

Result, the strongest simple baselines: neither reaches the head, and one is worse than nothing. Both baselines were wired through the agent's own head plumbing with an identity readout, so they take exactly the decision the head takes over exactly the candidates it sees; the agent was not edited (`experience/results/r6.json` ; the sampled vote's temperature 0.8 and five samples were pre-registered, not tuned):

arm at the same decision	success	vs base, paired	vs head, paired
base, budget 12	18.7%	—	—
(a) the model's own highest-log-probability candidate	20.3%	+1.7 [−1.7, +5.3], p 0.46	head +11.3 [+6.0, +16.7], p 8×10^{-5}
(b) majority vote over five sampled hypotheses	12.3%	−6.3 [−10.7, −2.0], p 0.008	head +19.3 [+13.7, +25.0], p 1×10^{-10}

arm at the same decision	success	vs base, paired	vs head, paired
(c) base with 35% more search (§7.6, budget 16)	22.3%	+3.6	head +9.3 [+3.7, +15.0], p 0.002
head, budget 12	31.7%	+13.0	—

The second reading fires and the first does not: both baselines spend inference at the same decision, over the same enumerated candidates, and neither reaches the head, so what the head has that they lack is the readout fitted on verified history — experience, not the extra inference. The expectation held: the model's own preference lands between the base and the head and is not separated from the base, which is what a head that overruled it on 179 of 300 decisions would predict. The surprise is printed at the same prominence: majority vote is significantly worse than the greedy base it was meant to improve, 33 problems solved by the base alone against 14 by the vote alone. Sampling at the decision admits candidates the greedy path would never take, and five votes can carry one; a cheap and common baseline actively hurts at this decision. Bounds: charges are analytic as for the head, (a) paying the same passes and (b) five short generations, not read from the episodes' token counts; one seed, one model, 300 per arm.

Result, the second search structure: the mechanism transfers. The environment is SQL query repair against a fixed schema — regions are the clauses select, where, join, group and order; the verifier is the expected result set, so any query returning the right rows is a repair; every task carries a sibling clause so the symptom cannot pin its own region — under the same harness, gates and budget accounting (`docs/r7-sql-repair-scope.md` , committed before the generator; `experience/results/r7_env.json` , `r7_probe.json` , `r7.json`). Gates on the training set: distinct problems 300 of 300, distinct decision prompts 300 of 300, the symptom-only lookup table at 57.3% under the 60% cap. Three readings were written before the head was fitted: the probe at chance means the state does not carry the clause and there is nothing to transfer; the probe above chance with the head no better than the base is the localisation-versus-repair bound of §7.5 in a second grammar; the probe above chance and the head above the base by more than the paired interval is transfer. The head was re-fitted from this environment's own base episodes, so what is tested is the mechanism and not the weights.

The probe reads the faulty clause from the post-inspect state at 89.3% on held-out decisions against a per-task chance of 20.9% and the agent's own pick of 30.7%; permuted labels 16 to 28%, eight principal components 29.3%, one hundred examples 54.1%, and a fourth control this environment demanded, since 89% invites the suspicion of leakage: a table that reads only the symptom string reaches 57.3%, the gate's own statistic, so the head's 32-point margin over it is the number that says it reads structure. The head on a second seed's 300 problems, generated with zero signature overlap with the training seed:

arm, second seed, n = 300	success	actions	paired vs base
base	11.3%	11.74	—
head, re-fitted	23.7%	11.09	+12.3 [+6.7, +18.0], p 6×10^{-5}; actions −0.65 [−0.89, −0.43]

The third reading fires. Two things the record keeps beside it. The held-out quarter of the training seed, 75 problems, read +10.7 with an interval [−2.7, +24.0], the same direction and size and not a result at that n; it was reported as such at the time and the disjoint 300 is what decides. And the first disjoint seed failed the lookup-table gate at 61.3% and its run was stopped with nothing kept: across eight seeds the table reads 56 to 69%, so this grammar sits on the gate's boundary and a set is admitted by measurement, not by seed; the generator now prints the gate on the set it writes. Bounds: one model; two synthetic environments that share a harness and a verifier-decides-the-bug discipline whose blind spots would be invisible to both, a sentence written before the generator existed and not retired by the result; the mechanism transfers, the weights never did. With this the claim of §7 is experience-directed search on two search structures, and the paper stops.

8. Two Carrier Studies, in Brief

Full treatment in Paper VIII-c. Both are studies of what happens to a capability when experience is pushed into it by a weight update, and both are read under §3.

Chess. A frozen 3.45M evaluator under MCTS, an external oracle, and a learned prior over move types by position class fitted to the frozen net's own games, its strength chosen by a rule before the sweep so that an inert prior could not produce a null that reads as a finding. At equal search the prior wins 0.544 (+30 Elo, constraint passes); at half and a quarter of the search it wins 0.350 and 0.275 against an un-priored net at 0.350 and 0.225. Paired differences +0.00 to +0.05 with a standard error of 0.08. A ladder run puts the base at 1983 ± 73: adding search to the raw policy is worth +182 Elo, the prior +20 ± 55. The mechanism already in the system dominates the prior added to it.

A poet's voice. A rank-8 adapter on 63 regulated poems reduces the form-verifier pass rate from 32.7% to 17.3% at three epochs while its validation loss keeps falling, and at three epochs it regurgitates its training poems (9 of 34 outputs contain a whole training line). A blind judge, with memorisation probes run first and every generated poem checked against the training split, separates every arm from the poet at 92 to 100% on original poems; the arm that seemed closest to the poet's voice was the one copying him. Under the search reading — a regulated

quatrain is a search over characters under a rhyme table and a tonal pattern — the cost of a passing poem rises from 3.1 attempts for the base to 6.0 for the adapter. Loss and capability diverge; the verifier is the only measurement of the constraint.

9. What Accumulation Needs

P9, the claim that experience compounds — work, consolidate, improve, again — was pre-registered for the third set and could not be run as designed, for the reason in §4: the second generation's training states were byte-identical to the first's, because everything that determines the post-inspect state is fixed before the head fires. The only second generation available on this harness trains on the same states with the verifier's labels weighted by what the first generation achieved. It moved the probe (66.7% to 70.7%) and changed the argmax on 3.3% of training tasks; on the independent set it changed 4.0% of outcomes, three episodes of 75, a null by size and, by the letter of the pre-registration, a fail. At a single fixed decision point nothing accumulates but the target. Accumulation lives in a loop where the prior's earlier choices shape the states it later reads — more than one hypothesis per episode, the head touching each — and that loop, with the prompt gate applied at every decision the head touches and a base of its own, is Paper VIII-b.

Related work

Inference-time compute is now studied as a budget to allocate rather than a fixed cost [Snell et al. 2024; Brown et al. 2024], and Efficient Thinking I–III measured what that allocation buys: search extracts what the evaluator already contains, and only external information raises the ceiling. This paper adds the experience axis to that picture and keeps its rules — a pre-registered reading, a paired test, a stated bound. Self-consistency [Wang et al. 2022] and process reward models [Lightman et al. 2023] select among candidates the model has already produced; the head here acts earlier, choosing where the search looks next, and §7.7 measures both selection baselines at the same decision. Agent memories that store lessons as text [Shinn et al. 2023; Wang et al. 2023] are the carrier this paper calls A, and it is the one that fell furthest under the constraint. Activation steering [Turner et al. 2023; Li et al. 2023] is carrier C; the result that the hidden state carries the decision while an additive vector cannot act on it is the observation the read-only head was built from. Linear probes on hidden states [Alain & Bengio 2016; Kadavath et al. 2022; Burns et al. 2022] are the instrument; what is new is their use as a controller under a cost constraint, with the three probe controls and the task and prompt gates that separate a readout from a lookup table. Low-rank adaptation [Hu et al. 2021; Dettmers et al. 2023] is the weight-update carrier studied in §8's poet study, not as a baseline but as a case of what a weight update carries. The chess anchor stands on AlphaZero-style self-play [Silver et al. 2017] with a frozen evaluator, and the debugging environment is deliberately synthetic where SWE-bench [Jimenez et al. 2023] is real, so that the verifier decides every bug and the gates can be enforced at generation.

10. Registered predictions, scored

The proposal (`efficient-thinking-8-proposal.md` , registered before any run) carried ten predictions. Each is scored against the record; a miss is printed at the same size as a hit.

prediction (registered before the run)	outcome
P0 — the external verifier rejects ≥ 20% of candidate lessons	**Hit.** Eleven of twelve rejected, all of them positive "start here" lessons that were true (§6)
P1 — the best injected prior reduces actions-to-green on held-out same-family tasks by ≥ 25% at equal success	**Hit, by a mechanism the proposal did not list.** No injected prior did; the read-only head at equal success (24.0% both) takes 5.53 actions against the base's 19.81 (§7.6)
P2 — that prior beats distilled text memory at ≤ 0.5× total tokens	**Hit, vacuously.** Text memory fell below the base on every problem at 3–4× input tokens; the head beats it at +34% (§7.3, §7.4)
P3 — mid-depth injection beats early and late	**Not scored as registered.** No early/mid/late comparison of an injected prior was run: the steering vector was measured at layer 27 and failed the constraint; the head was wired at layer 18, where its probe and the layer-27 probe were within 1.3 points (§7.3, §7.4)
P4 — gradient-free steering vectors recover ≥ 50% of the trained adapter's gain	**Not scored.** The adapter arm was not run in the debugging field; there was no gain to recover
P5 — out-of-domain regression ≤ 2 points for C, D, F	**Not scored for C and F**, which failed the constraint before the check applied; **holds by construction for the head**, which changes no weight
P6 — an invalidated lesson decays within three rounds	**Not run.** The lifecycle's decay arm belongs to Paper VIII-b (§5)
P7 — lesson text transfers to 7B, vectors do not	**Miss on the first half.** Text memory on 7B was destructive, not transferable; what transferred was the head, to a second family (§7.4)
P8 — a learned move prior cuts simulations to 2600 by ≥ 30%	**Miss.** +20 ± 55 Elo at equal search, no saving at any simulation count (§7.3, §8)
P9 — efficiency improves monotonically over three rounds, then plateaus	**Fail by the letter.** The second generation's training states were byte-identical to the first's; the only available second generation changed 4.0% of outcomes, a null (§9)

Pre-registered readings written during the study — the frontier control's two limbs (§7.6), the curve's three readings (§7.6), the two baseline readings (§7.7) and the second structure's three (§7.7) — are scored in place, with the readings that did not fire printed beside the ones that did.

11. Limitations and future work

One task family per search structure, both synthetic, both sharing a harness and a verifier-decides-the-bug discipline whose blind spots would be invisible to both. Two frozen models of 7–8B parameters; a 3B model floored and is not a counterexample. One seed for the frontier curve and for the second structure. The controller is charged at its measured, unbatched cost, and the simple baselines are charged analytically, not from their episodes' token counts. The head reads one decision; nothing here accumulates, and §9 says why it could not on this harness. The SQL grammar sits on the lookup-table gate's boundary, so a set there is admitted by measurement rather than by seed. Nothing in this paper uses production traffic or any data about a person.

What follows is set by these bounds. Paper VIII-b builds the loop in which the head's earlier choices shape the states it later reads and asks whether a second generation beats its parent — and, with a head fitted on the agent's own unverified success claims, whether the verifier's bits are the gain. Paper VIII-c takes the two carrier studies of §8 to full length. Paper VIII-d is the instruments: the gates, the controls and the persistence rule as a checklist for any experience-learning claim. A single-pass head that scores candidate families without enumerating them would remove the enumeration premium entirely and is a new mechanism with its own ladder.

12. Conclusion

An experience prior is a constraint, not a component. Under it, six ways of carrying experience into a frozen model's search failed, in two fields, and the pattern of their failures says why: each touched the competence the search depends on, or acted at a point where there was nothing yet to read. One carrier passed, twice, on independent problems: a read-only head over the model's own hidden state after one observation, choosing among the model's own candidates. Its effect is real, bounded to the kind of failure it can reach, and no larger for a better fit. The largest effect in the study was a hand-written rule about when to act, which is the rule a controller is supposed to learn, and the question this paper leaves is whether one can be learned that compounds.

The series reads as one argument from here. Efficient Thinking I asked what happens when a fixed intelligence is given more thinking, and found that search substitutes for size: $I + C \uparrow \Rightarrow Q\uparrow$. Papers II to VII found the limits of that substitution — the evaluator, the verifier, the supervision, and the information already in the system. This paper asks what happens after

the intelligence has experience of how to think, and finds that historical computation can teach a frozen intelligence where to direct future computation: $Q_E(C) > Q_0(C)$ over the measured compute range, on one task family, and not by the inference spent — the model's own preference at the same decision does not reach it, and on a second search structure with the head re-fitted from that structure's own history the same mechanism moves the frontier again (§7.7). Search moves a system along its frontier; experience moves the frontier, and it is a fact about experience-directed search rather than about one grammar. This paper stops there. Paper VIII-b asks whether it moves again — $Q_2(C) > Q_1(C) > Q_0(C)$ with the intelligence frozen throughout — which would close the loop this series opened: intelligence, search, outcome, verification, experience, better search.

Reproducibility

Every number in this paper is in the repository under `experience/`: environments (`et8_env_v3.py`, `et8_sql_env.py`), the agent (`et8_agent.py`, unchanged from the first measured run; the SQL agent is a separate file), the head and its controls (`et8_head_v3.py`, `et8_sql_head.py`), the gates (`et8_task_gate.py`, `et8_prompt_gate.py`), the paired statistics (`paired_stats.py`), and the results as JSON under `experience/results/` with the command that produced each table recorded beside it. Trajectories and decision states are persisted per run. The package's measured constants live in one registry (`experience/results/et8_constants.json`) with a checker that fails on drift (`check_constants.py`); the inputs the repository does not track are listed with counts and checksums in `docs/inputs-manifest.md`. The chronology, including every withdrawn claim, is Appendix A.

References

- Alain, G. & Bengio, Y. (2016). *Understanding intermediate layers using linear classifier probes*. arXiv:1610.01644.
- Alsakka, L. (2026). *Efficient Thinking I: Measuring What Capability Costs*. This series.
- Alsakka, L. (2026). *Efficient Thinking II: Where Search Pays and Where It Can't*. This series.
- Alsakka, L. (2026). *Efficient Thinking III: Efficient Judging*. This series.
- Brown, B. et al. (2024). *Large Language Monkeys: Scaling Inference Compute with Repeated Sampling*. arXiv:2407.21787.
- Burns, C. et al. (2022). *Discovering Latent Knowledge in Language Models Without Supervision*. arXiv:2212.03827.
- Dettmers, T. et al. (2023). *QLoRA: Efficient Finetuning of Quantized LLMs*. arXiv:2305.14314.

- Hu, E. J. et al. (2021). *LoRA: Low-Rank Adaptation of Large Language Models.* arXiv:2106.09685.
- Jimenez, C. E. et al. (2023). *SWE-bench: Can Language Models Resolve Real-World GitHub Issues?* arXiv:2310.06770.
- Kadavath, S. et al. (2022). *Language Models (Mostly) Know What They Know.* arXiv:2207.05221.
- Li, K. et al. (2023). *Inference-Time Intervention: Eliciting Truthful Answers from a Language Model.* NeurIPS.
- Lightman, H. et al. (2023). *Let's Verify Step by Step*. arXiv:2305.20050.
- McNemar, Q. (1947). *Note on the sampling error of the difference between correlated proportions or percentages.* Psychometrika 12.
- Shinn, N. et al. (2023). *Reflexion: Language Agents with Verbal Reinforcement Learning*. NeurIPS.
- Silver, D. et al. (2017). *Mastering the game of Go without human knowledge*. Nature 550.
- Snell, C. et al. (2024). *Scaling LLM Test-Time Compute Optimally can be More Effective than Scaling Model Parameters*. arXiv:2408.03314.
- Turner, A. M. et al. (2023). *Activation Addition: Steering Language Models Without Optimization*. arXiv:2308.10248.
- Wang, G. et al. (2023). *Voyager: An Open-Ended Embodied Agent with Large Language Models*. arXiv:2305.16291.
- Wang, X. et al. (2022). *Self-Consistency Improves Chain of Thought Reasoning in Language Models*. arXiv:2203.11171.

Appendix A. How This Was Found

The order in which the results above were reached, with what each error cost and the rule it earned. Dates are 2026. The work was done by one experimenter (E), who built and ran everything, and one reviewer (R), who ruled on design and pre-registration; both are pseudonymous here and belong to no institution named in this series.

1. **09-05 to 09-14.** The proposal, the environment, a 3B and then a 7B base over 2,000 tasks, and the first three mechanisms. Two 7B seeds at 59.9% and 59.3% were read as reproducibility. They were the tell for A.5.
2. **09-16, morning.** Text memory from the 3B's weak episodes: 20.0% against 62.5%. The content hypothesis — the lessons were bad — was written down as the likely cause. *Rule: pre-register the fair test before the excuse.*
3. **09-16.** The fair test: the 7B's own lessons, 5.0%. The carrier is the harm. The gate was then built as a rejection test and rejected 11 of 12 true lessons.
4. **09-16.** Steering at $\alpha = 4$: a first slice raised localisation by two episodes and was published as "the clean separation of the two deficits"; it did not survive the other three. *Rule: no number from one slice.*

5. **09-16 to 09-17.** The head for G scored 100% held out, passed its permutation control, survived projection to eight dimensions and training on 100 examples. That pattern is a copied key, not a readout: the task set carried ten symptom strings, each pinning one region. *Rule: three probe controls, always; the prompt gate.*
6. **09-17.** The second task set was built to fix the symptom axis and inherited the defect beneath it: 2,000 files, 11 problems. Every magnitude of the week was withdrawn as a size and kept as a direction; "no slice reverses" was withdrawn by name. *Rule: count the problems before counting anything else — the task gate.*
7. **09-17.** The naive rule was pre-registered to reach the ceiling and refuted by a three-episode smoke test before it cost a run; it destroys the group where symptom and bug differ. *Rule: ask what a rule does to the other group before pre-registering it.* Then the canned control turned out to be an oracle on the bug class as well as the region, and matched the model's own words to within two episodes.
8. **09-17.** The inspect-first loop: 2 of 11 to 4 of 11. The largest effect of the week, and not a prior.
9. **09-17.** The prompt gate refused the second set's post-inspect state too (7 of 80); chasing why found the 11 problems of item 6. The third set was scoped before it was generated: a program grammar, tests from the clean program's behaviour, file count equal to signature count, a replication set under a second seed held unrun. Its first attempt reached the entropy bar by information starvation and was discarded.
10. **09-17.** The third set's gate read 1.000. The probe beat the agent and degraded under all three controls. G cleared both limbs on one set, then on the independent set. The effect sat in the decisions the head changed.
11. **09-17.** The hundred-example control scored above the full head on one fit; chasing the fix found the optimizer's step size moving the probe by 35 points, and the control itself had been a single draw. *Rules: a second-order fit; five draws.* Re-fitting bought five fewer episodes, inside noise, in one direction on one set and the other on the other.
12. **09-18.** P9's second generation reported the first's numbers exactly; the states were byte-identical (§4, §9). The outcome-weighted target moved the probe and not the decisions. 12a. **09-19.** With the matched-compute point read on both limbs (§7.6), R set the stop rule this paper adopts: the two §7.7 controls — the model's own preference at the same decision and cost, and a second search structure with the head re-fitted — close the empirical story, and further work belongs to the next papers rather than to a larger paper. R's one-line reading of the core result is the one §12 keeps: the same frozen model, informed by verified history, obtains more task success from the same inference computation. 12b. **09-19 to 09-20.** The three closing runs. The frontier curve read above the base across the measured range and equal quality at 2.32× less compute (§7.6). Neither simple baseline at the same decision reached the head, and a five-sample majority vote was worse than the greedy base (§7.7). The second search structure's first disjoint seed failed the lookup-table gate and its run was stopped unfinished; the ceiling check found a verifier that could never say yes (tuples against lists) and a positive control became structural; the mechanism transferred on the admitted seed (§7.7). Three results E had committed on E's box were found never pushed and were pushed the same day. *Rules: a gate is printed by the*

generator on the set it writes; a positive control runs before any rate; a sha is quoted only once it resolves on the shared remote. The paper closed on R7, as pre-registered, whichever way it fell.

13. **Throughout.** Two runs were lost to moving the working tree under a process that was writing to it; three artefacts a later question needed had not been the artefacts a run was written to save; one results table was typed from memory and corrected the same minute; one counting function assumed the single-outcome property its author had just written a caveat against. *Rules: never move a tree under a writer; persist state and action; every table carries its command; a counting function carries its own caveat*.