

Efficient Thinking II: Where Search Pays and Where It Can't

Testing the evaluator $\times$ search decomposition in a solved game, language reasoning, and sequential control

Louay Alsakka · July 17, 2026 · v1.0.2

Abstract

Efficient Thinking I proposed a working decomposition from chess: strength = evaluator $\times$ search. A single-pass evaluator sets a base level of capability, inference-time search multiplies that base and then saturates against the evaluator's ceiling, and self-improvement stalls because nothing inside a closed system raises the ceiling. This paper tests whether that decomposition is a chess artifact. We take it to three deliberately different settings: Connect-4, where a perfect solver makes every quantity exactly measurable; LLM mathematical reasoning, a non-game domain with a natural verifier; and a gridworld MDP, where value iteration supplies the exact optimal value function.

The central result is the size-vs-search crossover, located and measured. On a 0.5B-to-72B model ladder (500-problem, 32-sample caches), model size dominates search below roughly 3B: a base that rarely finds the answer leaves search nothing to select. Above roughly 7B the relationship inverts, and how sharply it inverts depends on the benchmark. On saturated GSM8K, 7B with self-consistency pulls level with 14B decoding greedily, and the paired McNemar scores the strict win as a tie ($p = 0.44$), so we report parity there. On unsaturated MATH, a pre-registered rerun at $n = 500$ with true-greedy comparators makes the same flip significant: 7B+search 72.8% against 14B-greedy 68.2%, $p = 0.0027$, surviving Bonferroni correction across all tested flips. Between the two regimes the search lift collapses monotonically, from +16 points to +0.7, as the base grows more competent. The threshold sits at roughly 7B on both benchmarks: its location is benchmark-robust, and only its sharpness is difficulty-dependent — a significant efficiency win where the benchmark has headroom, a statistical tie where it is saturated. A five-line gridworld model reproduces the mechanism against an exact oracle: lookahead recovers a mildly noisy evaluator from 48% to 96% of optimal and cannot rescue a badly degraded one.

Every remaining experiment locates the binding constraint at the evaluator. In reasoning, consensus saturates while a perfect verifier over the same samples keeps climbing (+14.2 points); capability rises smoothly with verifier accuracy, with no threshold; and search

improves a policy several times more than it improves a judge, because repeated judgment reproduces the same systematic error. A two-floor fine-tuning sweep shows training data setting the destination (~26% on the full GSM8K train set) regardless of whether the model climbs from a 1.3% floor or erodes down from a 35.3% one. And across five self-improvement experiments in two games, self-generated targets never break the plateau, while changing only the value target to an external oracle breaks it immediately. The one-sentence summary of the paper: search extracts what the evaluator already contains, the efficient lever flips at a measurable competence threshold, and only external information raises the ceiling. All experiments run on two Apple-Silicon machines; claims are scoped to these domains and budgets as recurring patterns, not laws.

Results at a glance

finding	measurement	where
the size-vs-search crossover, located	size wins below ~3B; at 7–14B a smaller model + search matches the next size up — significant on MATH ($p = 0.0027$), a tie on saturated GSM8K ($p = 0.44$); lift collapses $+16 \rightarrow +0.7$	§4
the threshold's location is benchmark-robust	crossover at ~7B on both benchmarks; significance splits by benchmark saturation, not by threshold location	§4
the mechanism, in five lines of NumPy	lookahead recovers $48\% \rightarrow 96\%$ at $\sigma = 0.25$; caps ~46% at $\sigma = 1$	§3
the evaluator is the ceiling	+14.2 verifier gap; smooth $75 \rightarrow 88\%$ in verifier accuracy; a judge barely improves with search	§5
data sets the destination, not the path	two fine-tuning floors (1.3% and 35.3%) converge at ~26%; in Connect-4, data binds and capacity is slack	§6–§7
only external information raises the ceiling	five self-play plateaus; an oracle value target breaks one ($+400 \rightarrow +719$), nothing else changed	§8

1. Introduction

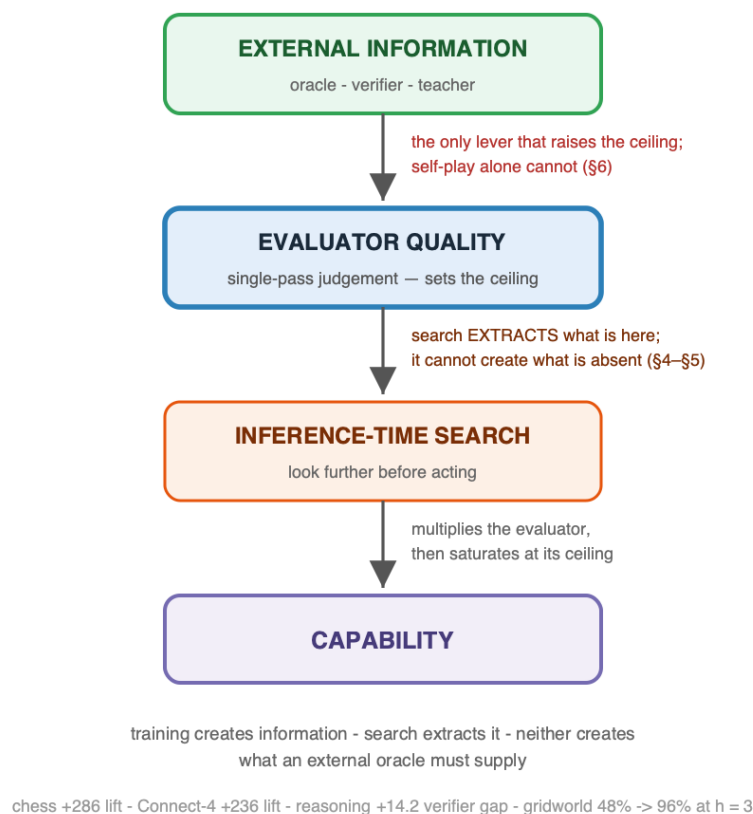
Modern game-playing systems and reasoning systems both buy most of their capability with two levers that are easy to conflate. The first is a learned evaluator: a network's single-pass judgement of a position, an answer, or a state. The second is search: spending inference-time compute to look further before committing. Efficient Thinking I separated the two in chess and found that a 3.45M-parameter network reaching roughly 2150 Elo open-loop reaches roughly 2800 with MCTS, that the search gain scales log-linearly in simulations, and that self-

play stalls because the evaluator, not the parameter count or the search budget, sets the ceiling.

If that structure is real rather than an artifact of chess, it should reappear in settings that share nothing with chess but the two levers. This paper runs that test three ways. Connect-4 is solved, so a perfect oracle and a graded opponent ladder let us measure every quantity exactly. LLM mathematical reasoning is not a game at all, but it has a natural verifier in the checkable final answer. And a gridworld MDP is a sequential-control setting where value iteration hands us the exact optimal value function, so the evaluator can be degraded by a known amount and nothing else.

Figure 1 — one causal chain, four instantiations. The whole paper is a single chain: external information is the only lever that raises evaluator quality; inference-time search extracts, and never creates, what the evaluator already contains; and the two compose into capability. Each arrow carries a load-bearing result of this work. Figure 1 is the whole paper.

Figure 1 — one causal chain, four instantiations



Read top to bottom, the figure is a causal chain. Read left to right across the table below, it is the claim that one mechanism spans games, language, and control. The bottom row is the load-bearing number each column earns; every quantitative result in the paper measures one arrow in one column.

chain link	Chess (ET-I)	Connect-4 (§3)	LLM reasoning (§4)	Gridworld (§5)
external information	Stockfish labels	perfect solver	verifier / RLVR signal	value iteration (V^*)
→ evaluator quality	3.45M value net	conv value net	P(answer correct)	$\hat{V} = V^* + \text{noise}$
→ inference-time search	MCTS simulations	PUCT MCTS	best-of-N / self-consistency	h-step lookahead
→ capability	~2150 → 2800 Elo	GELO vs ladder	GSM8K / MATH accuracy	% of optimal
load-bearing number	+286 search lift	+236 search lift	+14.2 verifier gap	48% → 96% at $h = 3$

Two terms in that chain carry weight and need definitions, because "search extracts and never creates" invites a fair objection: stochastic search plainly discovers trajectories and answers that greedy decoding would never produce. It does — and the definitions accommodate that. For selection search (the best-of-N family), "the policy contains an answer" means the answer has support in the policy's sampling distribution at the operating temperature, and the measured proxy for that support is coverage: oracle-best-of-N, a column in every frontier table below. Sampling discovers new samples, but every sample is drawn from the policy, so discovery is the realization of existing support, and any selector's accuracy is bounded by coverage at every N. "Creates" would mean exceeding that bound, which no selector can. For lookahead search, "the evaluator contains" means the ranking information in the value function together with the environment's terminal signals; the terminal rules are themselves external information, which is why self-play works at all, and §8 states the resulting bound as the search-over-policy margin. In both senses, search converts contained information into decisions and cannot exceed the containment ceiling; raising the ceiling requires information from outside — more data, terminal outcomes, or an oracle.

Our contributions, in the order the paper develops them:

1. **The size-vs-search crossover, located and measured (§4):** on a 0.5B → 72B ladder, size dominates below ~3B and search wins above ~7B, with the search lift collapsing monotonically as the base grows competent. The replication on MATH was pre-registered and scored: the threshold's location is benchmark-robust; only its sharpness is difficulty-dependent.
2. **A minimal exact-oracle model of the mechanism (§3):** a gridworld sweep of evaluator noise against search horizon, averaged over eight grids, showing that search compensates evaluator variance and cannot compensate evaluator bias.
3. **The evaluator bottleneck as a gradient and an asymmetry, not just a gap (§5):** that a verifier beats consensus is established; we show capability is smooth in

verifier accuracy, that a real imperfect judge captures most of the oracle headroom, and that search improves a policy several times more than it improves the judge itself.

4. **The training-data lever, placed in both arms** (§6–§7): a Connect-4 data × capacity × search grid that separates the three levers cell by cell, and a two-floor language fine-tuning sweep in which the data determines where both curves land.
5. **An oracle-target positive control for self-improvement** (§8): five plateaued self-play loops and one minimal intervention that breaks the plateau, isolating the signal source as the causal variable.
6. **GELO, a measurement framework** (§2): a calibrate-first logistic scale that makes search-lift magnitudes comparable across domains in odds units, and that surfaced a finding of its own — the mid-ladder ordering of models is benchmark-dependent.

A note on statistical power, stated once rather than scattered. The powered results are the reasoning frontier ($n = 500$ on GSM8K, $n = 300$ – 500 on MATH with the 7B/14B pair extended, 32- and 16-sample caches), the gridworld (eight grids, mean ± sd), the Connect-4 capacity cells (three seeds), and the arena (150 problems). The §5 evaluator-ceiling gap, first measured at exploratory power (Qwen-4B, 120 problems, +14.2), is recomputed over the 500-problem caches and holds per model (+3.8 to +38.8); the graded-verifier gradient is likewise recomputed at $n = 500$ across the ladder (§5, with its anchor-definition change stated); only the judge-asymmetry and real-judge results remain at exploratory power pending recomputation, and we say so where they appear. The crossover's paired McNemar tests — the correct test, since the compared systems share problems — split by benchmark saturation: the strict flip is significant on unsaturated MATH and a tie on saturated GSM8K, with exact counts and p-values in §4 and the predictions scored in §10; the lift-collapse trend, which does not depend on any strict ordering, is the benchmark-independent evidence.

2. Measurement: one calibrated scale

Elo, Bradley–Terry, and one-parameter item-response theory are the same latent-ability logistic model, $P(\text{win or solve}) = 1 / (1 + 10^{-(\theta - d)/400})$. That equivalence is textbook, not a contribution. What we add is a protocol and a scoped use. The protocol: calibrate a reference ladder first, fit ratings from a full cross-table rather than from per-opponent win rates, gate on logistic goodness-of-fit before trusting any rating, and pin interpretable anchors. The scoped use: because we keep chess's constants (400 GELO is 10× the odds; 120 GELO is one doubling), a search lift measured in Connect-4 and one measured in chess are expressed in identical odds units and can be compared directly. That comparison is the only cross-domain claim GELO carries. A "2500" in reasoning and a "2500" in chess share a ruler by construction, not a difficulty; we compare curve shapes and lift magnitudes across domains, never absolute levels. An agent can be placed against a graded reference ladder or pairwise against other agents with one pinned as the reference. The full specification is in [docs/ge-lo.md](#).

The first calibration, on Connect-4, passes the gate cleanly: the mean absolute error between predicted and observed pairwise win rates is 0.058, so the logistic model genuinely fits the game and the ratings are earned rather than assumed. Reasoning lands on the same axis two ways. A difficulty-anchored fit against MATH tiers gives a monotonic ladder (L1 +1273 to L5 +1712, roughly +110 GELO per level) and places a model by which tier it half-solves. And a pairwise arena, with Kimi-K2.5 judging head-to-head answers, places models directly against each other.

The arena also produced this paper's first scored prediction, and the prediction missed. An early 50-problem arena placed Qwen-1.5B above Qwen-3B, the opposite of their GSM8K accuracies, and we flagged the ordering as an under-powered artifact, expecting a powered rerun to fix it. The rerun (150 MATH problems, a five-model ladder, resilient concurrent judging) raised judge–verifier agreement from 72% to 84% and confirmed the coarse ordering (0.5B far below the mid-ladder, which sits below 7B $\approx$ 14B). But the inversion did not resolve into the expected order: 1.5B and 3B are statistically tied on MATH (ground-truth GELO +2419 versus +2392; clean-cache pass@1 30.6 versus 32.9) even though 3B clearly beats 1.5B on GSM8K (69.5 versus 52.8 pass@1). The honest conclusion is that the mid-ladder ordering is benchmark-dependent, not a fixable small-sample inversion. A benchmark separates two models only where its difficulty distribution carries information in their ability range; MATH evidently carries little between these two. The judge, at 84% agreement with the ground-truth verifier, is itself evaluator-limited: a referee can only rank as well as it can reason, which is why, on checkable tasks, a verifier still beats an LLM judge. Every check that the scale is earned rather than assumed — the fit gates, the anchors, the arena's agreement and its measured limits — is maintained as a running validation ledger alongside this paper (docs/gelo_validation_ledger.md), where the held-out calibration and transitivity audits are registered as the remaining queue.

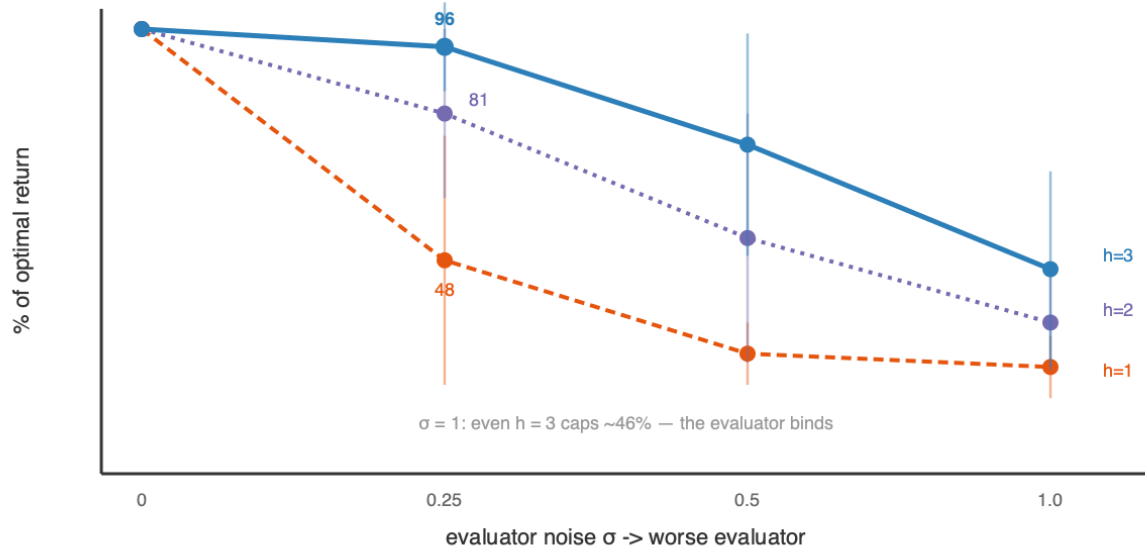
3. The mechanism in five lines (gridworld) — Anchor 1

Before any large experiment, the entire thesis can be reproduced in a setting with no confounds. We use a stochastic 8×8 gridworld with known dynamics, so value iteration gives the exact optimal value function V . *The controller receives a degraded evaluator*, V plus Gaussian noise of scale σ in units of the value spread, and plans with an MPC-style lookahead of horizon h , where $h = 1$ is greedy open-loop control. Return is normalized so that 0% is a random policy and 100% is optimal, and every cell is averaged over eight independent grid instances at 400 episodes each.

evaluator noise σ	open-loop ($h=1$)	$h=2$	$h=3$
0.0 (perfect)	100 $\pm$ 0	100 $\pm$ 0	100 $\pm$ 0
0.25	48 $\pm$ 28	81 $\pm$ 19	96 $\pm$ 10
0.5	27 $\pm$ 7	53 $\pm$ 28	74 $\pm$ 25

evaluator noise σ	open-loop ($h=1$)	$h=2$	$h=3$
1.0	24 ± 7	34 ± 11	46 ± 22

Anchor 1 — search rescues a noisy evaluator, monotonically in h (8-grid mean $\pm$ sd)



Three findings sit in this small table. With a perfect evaluator, search adds nothing: open-loop control is already optimal, and lookahead only earns its cost once the evaluator is imperfect. With a mildly noisy evaluator, search compensates dramatically and monotonically in horizon: at $\sigma = 0.25$ the greedy policy falls to 48% of optimal while three-step lookahead recovers 96%. And as the evaluator degrades further, search recovers less and less, capping near 46% at $\sigma = 1$ regardless of horizon. Past a point, the evaluator and not the search budget is the binding constraint. Search buys back evaluator variance; it cannot buy back evaluator bias. The single-grid version of this experiment initially showed $h = 3$ below $h = 2$ at $\sigma = 0.25$, and we hypothesized a mechanism for it; the eight-grid average shows the dip was noise (the per-cell standard deviations reach 28 points), and the effect is monotone in h at every imperfect σ . That correction is reported in §10 alongside the paper's other scored predictions.

Why this happens is not mysterious, and classical control theory predicts the table's shape. Bounds of the Singh–Yee family show that acting greedily on a value function with error ϵ costs at most on the order of $\gamma\epsilon/(1-\gamma)$ in return, and multi-step lookahead tightens the dependence geometrically in the horizon, to order $\gamma^h \epsilon$ (Singh & Yee 1994; Efroni et al. 2018). Three features of our table are exactly what those bounds predict: improvement that is monotone in h , a loss floor that remains proportional to ϵ so that no horizon rescues a sufficiently degraded evaluator, and zero benefit at $\epsilon = 0$. The gridworld is those bounds, measured.

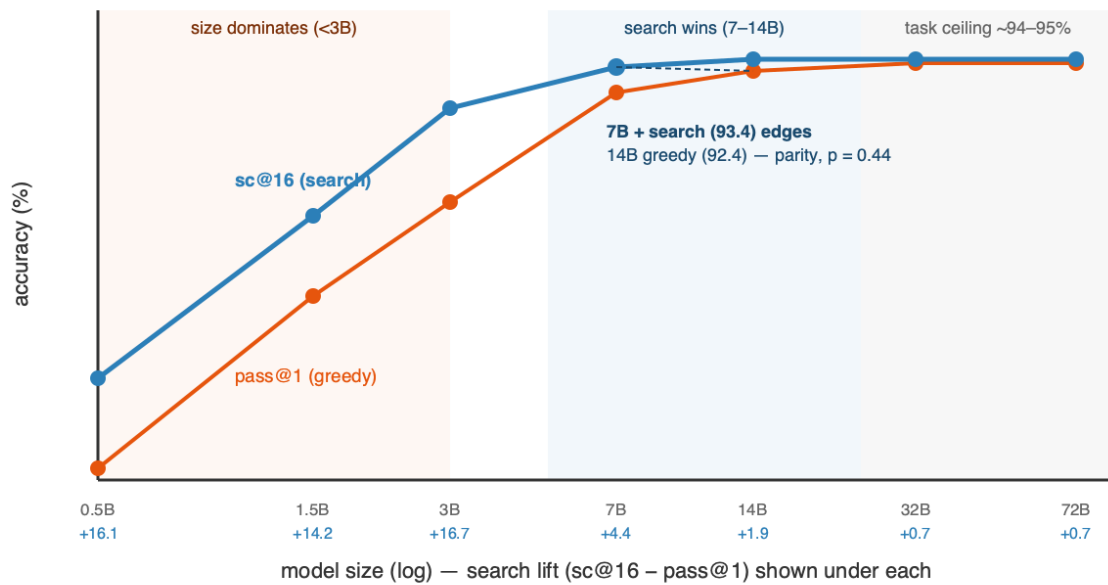
4. The crossover (LLM reasoning) — Anchor 2

The mapping from chess to language is direct. A position becomes a partial reasoning trace; the policy's move priors become the next-step distribution; the evaluation $P(\text{win})$ becomes $P(\text{this reasoning is correct})$; the terminal result becomes a verifier on the final answer, exact in mathematics; and MCTS becomes inference-time compute spent across reasoning paths. An LLM has no built-in correctness evaluator — its only native scorer is the next-token softmax, which judges plausibility, never correctness — so every search result below operates at whole-answer granularity with an external selector: sample N complete answers at temperature, then select by majority vote, by a checkable verifier, or by a real LLM judge. The two search knobs are N and the selector, both entirely external to the frozen weights.

The core efficiency question is then concrete. For a fixed compute budget, is it better to buy a bigger model or more search over a smaller one? We swept a clean size ladder, Qwen2.5 from 0.5B to 72B, against self-consistency N on GSM8K, with 500 problems and 32-sample caches ($\text{sc}@N$ is majority vote over N ; $\text{oracle}@32$ is a perfect verifier's coverage of the same 32 samples):

model	pass@1	sc@4	sc@16	sc@32	oracle@32	search lift
0.5B	22.1	28.8	38.2	40.8	79.6	+16.1
1.5B	52.8	60.8	67.0	66.6	93.8	+14.2
3B	69.5	79.8	86.2	87.2	96.8	+16.7
7B	89.0	91.2	93.4	93.6	97.4	+4.4
14B	92.7	93.0	94.6	94.4	97.4	+1.9
32B	94.1	94.2	94.8	95.0	98.2	+0.7
72B	94.1	95.0	94.8	95.0	97.8	+0.7

Anchor 2 — the size-vs-search crossover (GSM8K, n = 500, 32-sample caches)



Both regimes are visible on one sweep. Below the competence threshold, size dominates: the 3B model decoding greedily (69.5%) beats the 0.5B and 1.5B models at any N and any compute multiple, because a base that rarely finds the answer leaves search nothing to select. Above the threshold the relationship changes character: 7B with sc@16 (93.4%) reaches, and directionally exceeds, 14B decoding greedily (92.7%), and 14B with search does the same against 32B greedy. Between the two regimes the search lift collapses monotonically, from +16 points at 0.5B to +0.7 at 32B, where GSM8K's ceiling leaves nothing left to extract. The reconciliation with chess is the decomposition itself. The chess net was already competent at its task, so search extracted a great deal; the sub-3B models are not, so search extracts little; the 7–14B models are competent and unsaturated, which is exactly where search pays. Search complements a competent base and cannot substitute for one. The paired McNemar tests on the per-problem caches settle the strict ordering by benchmark. On saturated GSM8K they are ties — with true-greedy comparators, 7B-versus-14B has discordant counts 16/11 (exact $p = 0.44$) and 14B-versus-32B has 12/8 ($p = 0.50$) — so here the confirmed form is parity: above roughly 7B, search buys statistical indistinguishability from the next size class, the direction consistently favoring search. On unsaturated MATH the same 7B-versus-14B flip is significant (39/16, $p = 0.0027$; §4 below), so the strict win holds where the benchmark has headroom — and since three flips were tested, the significant one must survive multiple-comparison correction, which it does with room to spare (Bonferroni-adjusted across the three tests, $p = 0.0081$). No sub-threshold model reaches the next class at any N (0.5B at sc@32 scores 40.8 against 1.5B's 52.8 greedy). In deployment terms the parity point is a memory-for-latency trade: 7B with sc@16 uses roughly half the memory and about eight times the decoding compute of 14B greedy at the same accuracy, which is exactly the choice a memory-constrained system faces.

GSM8K saturates near 95%, which compresses the top of the ladder, so we pre-registered a prediction and re-ran the ladder on MATH-500, which does not saturate at these scales. The

prediction: the crossover region should widen, and the search lift should stay alive further up the ladder. On MATH (7B and 14B extended to $n = 500$, 3B and 32B at $n = 300$; 16-sample caches):

model	pass@1	sc@16	oracle@16	search lift
3B	32.9	51.7	75.0	+18.8
7B	65.5	74.0	85.6	+8.5
14B	69.9	76.2	86.0	+6.3
32B	75.0	81.0	89.3	+6.0
72B	75.4	81.3	89.0	+5.9

The prediction scored as a hit, and the strict question — flagged at $n = 300$ as a tie ($p = 0.14$) — resolved once the caches were extended. The hit: the margin is far clearer on MATH — 7B with search (74.0%) is ahead of 14B greedy (69.9% mean, 68.2% true-greedy) — and at $n = 500$ with true-greedy comparators the paired McNemar makes the flip significant ($b = 39$ problems where 7B+search is right and 14B-greedy wrong versus $c = 16$ the reverse, $p = 0.0027$), where the saturated GSM8K flips stay ties ($p = 0.44, 0.50$). The lift also stays alive far higher up the ladder: +6.0 at 32B against GSM8K's +0.7 at the same size, scoring the registered "+3–5 at 32B" prediction a hit on liveness, with the magnitude landing above the registered band. The crossover point itself does not move — it sits at roughly 7B on both benchmarks — so the competence threshold's location is benchmark-robust and only its sharpness is difficulty-dependent: the boundary between a significant efficiency win where the benchmark has headroom and a statistical tie where it saturates. The 72B point matches 32B (75.4 versus 75.0 pass@1), so the MATH ladder tops out too — but unlike GSM8K, the search lift stays alive all the way up: +5.9 at 72B against GSM8K's +0.7 at 32B, so the difficulty-dependent sharpness holds to the very top of the ladder.

One further search axis deserves a note. Measuring greedy accuracy against generation budget, the 1.5B model climbs from 7.5% at 128 tokens to 48.8% at 512 and is then flat through 2048, and the 3B model behaves the same way, flat past 512 at 67.5%. The gain is almost entirely a don't-truncate effect: a non-thinking base model finishes its chain of thought in about 512 tokens and stops, so additional budget buys nothing. Genuine serial scaling of the o1 and R1 kind requires a model trained to keep deliberating. For an untrained base model, the parallel axis — more samples and a better selector — is the live lever, which is one more instance of the same rule: search extracts only what the model already contains.

5. The evaluator is the ceiling (LLM reasoning)

The experiments in this section are the paper's oldest and remain at exploratory power (60–120 problems; roughly ± 8 points at 95% on a single proportion), pending recomputation from

the powered caches. Their pattern, however, is consistent across all of them and reproduces at $n = 500$ in the frontier table's oracle column, so we report them with that caveat stated here once.

Search scales accuracy and then saturates. Qwen-4B on GSM8K climbs from 66.7% greedy to 77.5% at $sc@16$ and stays there at $sc@32$. Majority vote is a verifier-free evaluator, and it stops helping. The saturation is an evaluator ceiling rather than a policy or search ceiling: over the very same samples, oracle-best-of- N , which is a perfect verifier, climbs to 91.7% and is still rising at $N = 32$. The right answer is in the sample set 91.7% of the time; consensus cannot select it. The gap is +14.2 points. That a verifier beats consensus is the founding observation of the verifier literature — it is why verifiers were trained in the first place [Cobbe et al. 2021; Lightman et al. 2023], and Brown et al. [2024] document the same coverage-versus-selection gap growing with N at frontier scale. What this paper adds is the placement of the gap inside the decomposition and the two results that follow.

First, the gradient. Varying a synthetic verifier's per-item accuracy q over the same samples, capability rises smoothly from 75.0% at $q = 0.5$, which is verifier-free consensus, to 88.3% at $q = 1.0$, passing 77.7, 80.6, 83.0, and 85.9 along the way — about +2.6 points per 0.1 of verifier accuracy, with no threshold anywhere. The powered caches confirm the gradient's shape across the whole ladder: recomputed at $n = 500$ for every model, capability is linear in q with no threshold anywhere (7B: 48.7% at $q = 0.5$ rising to 97.4% at $q = 1.0$; 0.5B: 39.8% to 79.6%), with one definition made explicit rather than silently changed — the powered sweep anchors $q = 0.5$ at random selection among the samples, where the exploratory version anchored it at consensus, so the two slopes are not comparable even though both establish the same finding: every increment of evaluator accuracy buys capability, smoothly. Every increment of a better evaluator buys capability. A real, imperfect judge confirms the deployable middle of that curve: selecting among Qwen-1.5B samples with Kimi-K2.5 as the scorer beats majority vote at every N and captures most of the consensus-to-oracle headroom (at $N = 8$: consensus 65.0%, judge 75.0%, oracle 83.3%). A stronger selector buys about ten points on identical samples, so more search pays far more when a better evaluator spends it.

Second, the asymmetry. Running the judge itself with self-consistency — majority over N repeated judgments — raises its agreement with the ground-truth verifier only from 72.5% to 75.0% as N goes from 1 to 5, a nudge of about +2.5 points, against the +8 to +11 points the same search buys a policy. The reason is instructive. A policy benefits from search because some among many attempts land on the answer and selection finds them; a judge that cannot reason a problem out reproduces the same systematic error on every repetition. Compute can partly buy back a weak policy. It cannot buy back a weak evaluator, and that is precisely why the evaluator's quality rather than its search budget binds.

The whole section follows from one short proposition. Fix a problem and let sampled answers be independent draws from the policy's answer distribution. Then as N grows, oracle-best-of- N converges to coverage — the probability that the correct answer has support at all — while majority vote converges to an indicator of whether the correct answer is the mode of the

distribution. Coverage always weakly exceeds mode-correctness, and the gap between them is precisely the mass of problems on which the model can produce the right answer but produces some wrong answer more often. The +14.2 gap is a measurement of that mass; every selector between consensus and oracle is an estimator of the mode-versus-support distinction with accuracy q ; and the judge asymmetry is the same statement applied to a judge, whose repeated judgments concentrate on its own mode, so that when its per-problem error is systematic, aggregation converges to the same error. The proposition also yields a registered, cache-computable prediction: per model, self-consistency at large N should sit near the fraction of problems whose correct answer is modal, a quantity directly computable from the stored 32-sample caches. Run over the caches, the check required its own correction first: at $N = 32$ over a 32-sample cache, self-consistency and fraction-modal-correct are the same computation, so their exact agreement ($\Delta = 0.0$ for every model) verifies only that the two code paths are consistent, not the proposition — a vacuous test we caught in our own scoring (§10). The informative form of the prediction is convergence, and it holds: finite- N majorities approach the modal fraction from below with shrinking gaps at every model size (0.5B: 28.8 $\rightarrow$ 38.2 $\rightarrow$ 40.8 across sc@4/16/32; 3B: 79.8 $\rightarrow$ 86.2 $\rightarrow$ 87.2; 7B: 91.2 $\rightarrow$ 93.4 $\rightarrow$ 93.6), while oracle-best-of- N keeps climbing, so the coverage-minus-mode gap — the +14.2 exploratory number — reproduces at powered $n = 500$ as +3.8 points at 7B and +9.6 at 3B, widening to +27.2 at 1.5B and +38.8 at 0.5B as the model weakens and its samples disagree more. The proper out-of-sample test — the mode of the first sixteen samples predicting the majority vote of the second sixteen — closes it: run per model over the 500-problem caches, the two agree within sampling noise everywhere ($\max |\Delta| = 1.0$ point: 0.5B -1.0 , 3B $+0.4$, 32B $+0.0$, the rest $+0.2$), so the modal answer of one half predicts the other half's vote, exactly as exchangeable draws from a stable answer distribution require — the informative form of Prop 2, confirmed on clean out-of-sample data rather than the $N = 32$ identity.

6. The levers separated in a solved game (Connect-4)

Connect-4 is solved, so the exact solver is a perfect oracle and a depth-limited alpha-beta gives a calibrated opponent ladder: random at 0, depth-1 at +804, and depths 2 through 6 spanning +954 to +1070. The first ply of tactics is the single biggest step on the ladder, larger than the next five plies combined, after which the heuristic ladder saturates — diminishing returns on search depth, quantified on a real axis. A small convolutional network trained on 12k oracle labels is the evaluator, and PUCT MCTS is the closed loop. The decomposition transfers to a second game: the raw net places at +644 GELO open-loop and +880 with MCTS-200, a search lift of about +236 GELO, roughly 4 $\times$ the odds per game and the same order as chess's +286 lift in identical units.

The evaluator-versus-search balance is data-dependent rather than intrinsic. With only 12k labels the raw net loses even to 1-ply search, which reads as search dominance. Tracing the open-loop ceiling against training data, however, the raw net climbs from +642 at 12k to +747 at 24k and +798 at 50k, reaching depth-1 parity. The apparent dominance of search was

mostly a starved evaluator. To separate the three levers directly, we swept training data against network capacity, placing every cell both open-loop and with MCTS-100:

data	small (~0.3M) open	large (~4M) open, 3-seed mean ± sd	search lift (small, +MCTS)
3k	+512	+278 ± 70	+147
12k	+673	+616 ± 92	+117
24k	+659	+665 ± 64	+210
50k	+801	+651 ± 266	+222

The table carries one reading per lever. Data binds: the open-loop ceiling climbs with labels on both nets, and the small net reaches depth-1 parity by 50k. Capacity is slack, and at low data it is harmful: the larger net's mean never beats the small net open-loop, and it is seed-unstable, with the 50k cells landing at +796, +879, and +278 across three seeds (mean +651 ± 266, median +796 — one collapsed seed drags the mean, and even the median sits just below the small net's +801). Extra parameters with too little data add noise, not skill. And search is a large multiplier that rescues most exactly where the evaluator is weakest, adding between +117 and +222 GELO on the small net with its largest rescues at the smallest data sizes. At this scale the binding constraint is data, capacity is slack, and search compensates — the same ordering chess found, here separated cell by cell.

7. The data lever in language (a two-floor fine-tuning sweep)

Size and search are two of the three levers; the third is training data, which Connect-4 carries above and which can also be placed directly in language. We LoRA-fine-tune two 0.5B models — the instruction-tuned Qwen2.5-0.5B at 4-bit and the non-instruct base at bf16, since at 4-bit the base fine-tune was unstable — on an increasing number of GSM8K examples at fixed epochs, so that more data does proportionally more training and the sweep isolates the data axis. Greedy accuracy on 150 held-out problems:

train examples	o (zero-shot)	64	256	1024	4096	7000 (full)
base (bf16) — from a low floor	1.3%	16.0	15.3	17.3	24.0	26.0%
instruct — from a competent floor	35.3%	19.3	20.0	22.7	24.0	26.7%

The two curves start from floors 34 points apart and converge. The base model climbs from a 1.3% floor to 26.0% at the full train set, a textbook data-scaling curve from nothing. The instruct model first falls from its 35.3% zero-shot floor to 19.3%, because a few thousand narrow examples trade away broad pretrained capability for GSM8K surface form, and then

buys the loss back gradually, reaching 26.7% at the full set — the same warm-start erosion Connect-4's self-play showed when a strong evaluator was pulled down toward a narrower signal. Both curves land at roughly 26%. The data determines the destination; the pretrained starting point determines only the direction of approach. At 150 problems the load-bearing observations are the two monotone orderings, the 1.3-versus-35.3 floor gap, and the joint convergence band; the 26.0-versus-26.7 endpoint difference is indistinguishable at this sample size, which is consistent with, though not yet proof of, a shared data-determined attractor. Stated plainly for what it does not show: this sweep demonstrates that data sets the destination, not that the data lever can climb past a solid pretrained baseline in language — neither curve ends above the instruct model's own zero-shot floor, and the larger-base, larger-data run that would test baseline-beating is registered future work.

8. Self-improvement, and the one intervention that breaks the plateau

The self-improvement question, stated value-first: fix the evaluator by distilling better-than-current value and policy targets back into the net, and strength should follow. We tested the loop five ways across two games, and it never broke the plateau. In Connect-4 from scratch, the evaluator genuinely improves (oracle value error falls from 0.48 to 0.355) and strength tracks it early, climbing to about +400 GELO, and then it stops; quadrupling the search budget per move does not raise it. Warm-starting from the supervised +644 net does not preserve the head start: the first iteration erodes it to +189, and it recovers only to roughly +480–500, well below the seed. In chess, three variants fail three ways — from scratch, open-loop strength bounces between 254 and 384 with no climb over 44 iterations; warm-started from a weak-policy seed it stalls at 300–326, a bootstrap barrier, since MCTS target quality depends on the policy prior; and started from a genuinely self-learned 1214-Elo plateau net it degrades to 833, because search over that value head does not play far enough above 1214 to generate improving targets.

The positive control isolates the cause. Changing only the value target, from the self-play outcome to an external depth-6 oracle, with the loop otherwise identical, Connect-4 self-training climbs past the +400 plateau to +719 by iteration 60, toward the oracle's own level, and the result is seed-independent: from the supervised +644 seed with oracle targets it reaches about +676. The target's information source, not the loop and not the starting point, is what matters.

Reported positive self-improvement results in the literature — ReST [Gulcehre et al. 2023], STaR [Zelikman et al. 2022], self-rewarding language models [Yuan et al. 2024] — are not counterexamples to this plateau; on inspection each injects external information implicitly, through a checkable verifier filtering the traces or human preference data seeding the reward model. What that literature lacks is the controlled isolation: our five loops and one positive control hold everything fixed except the signal source. The claim is therefore not that self-

improvement fails, but something sharper: self-improvement succeeds exactly in proportion to the external information its signal carries. The mechanism is simple enough to state in three sentences. A Monte-Carlo rollout gives the true value of a position under the current level of play, so training on those labels converges the evaluator to a perfectly calibrated evaluator of its own level and caps it there, since no move better than the current level ever appears in the data. The only way rollouts push past the current level is to play them above it, with search, and then the per-iteration gain equals the search-over-policy margin, which is itself bounded by evaluator quality. This is AlphaZero's engine, and precisely why it needs deep search multiplied by millions of games; at our scale the margin was too small.

9. Synthesis: what transfers, what shifts, what binds

Every experiment in the paper, including the failures, diagnoses one link in Figure 1. The map below lists each experiment by the lever it varied and the link that ended up binding:

experiment	domain	lever varied	what bound it	evidence
open- vs closed-loop	Connect-4	search (MCTS)	search <i>extracts</i>	+236 GELO search lift
ceiling vs data	Connect-4	evaluator data	evaluator	+642 → +798; depth-1 parity at 50k
data × capacity × search	Connect-4	all three	evaluator (data) ; capacity slack	small net ≥ large open-loop; open-loop ↑ with data
fine-tuning data sweep	reasoning	training-data size	evaluator (data) , unsaturated	19.3% → 26.7% monotonic in N
consensus vs oracle-best-of-N	reasoning	selector quality	evaluator	+14.2 (verifier ≫ consensus)
graded verifier	reasoning	evaluator accuracy q	evaluator	smooth 75% → 88%
Kimi-best-of-N	reasoning	a <i>real</i> selector	evaluator	65% → 75% at N=8
judge self-consistency	reasoning	search <i>on the evaluator</i>	evaluator (search can't fix it)	72.5% → 75% only
serial thinking length	reasoning	serial search	policy (untrained base)	flat past 512 tokens
size × search frontier	reasoning	size vs search	base competence (crossover)	size<3B, search>7B; lift +16→+0.7

experiment	domain	lever varied	what bound it	evidence
noisy-eval × lookahead	control	search horizon h	evaluator	$\sigma=0.25$: lookahead lifts 48%→96%; $\sigma\geq 1$ caps ~46%
self-play ×5	chess + C4	training signal	evaluator / search margin	plateau never breaks
oracle-value target	Connect-4	<i>external information</i>	— (positive control)	+400 → +719

Read down the "what bound it" column and the evaluator recurs; search binds only where the evaluator was starved into dominance or is competent but unextracted; and the one row that breaks a plateau is the one that injects external information. The same map, indexed by resource rather than by experiment, answers the practitioner's question of when each resource is worth spending on:

resource	when it binds	when it doesn't	evidence
training data	starved evaluator (early Connect-4; every LLM fine-tune)	no ceiling reached in our range	data sweeps, both arms (§3–§4)
evaluator quality	whenever search has saturated	never observed slack	+14.2 gap · graded 75→88 · judge asymmetry (§4)
search	competent-but-unextracted evaluator (chess 2448 base; gridworld $\sigma=0.25$)	below base competence (frontier); at perfection ($\sigma=0$)	+286 / +236 lifts · 48%→96% · size≫search (§4–§5)
capacity	never, in our regimes	everywhere tested — and <i>harmful</i> at low data	grid + seed collapses (§3); ET-I capacity sweep
external information	at every self-improvement plateau	— (always the ceiling-raiser)	oracle control +400→+719 · +14.2 verifier · Stockfish labels (§6)

Two rows carry the rollup. Capacity's honest cell is "never observed binding" — a resource whose answer here is never is what makes the table a measurement rather than a framework decoration. And external information is the row a three-lever menu would omit: the only entry that raises the evaluator's ceiling rather than spending against it, which is §8 in one line.

The synthesis in prose is short. The decomposition transferred: strength = evaluator × search held in two games, in language reasoning, and in sequential control, on one calibrated scale, as

the same saturating curve — Elo against simulations, accuracy against N, GELO against MCTS, return against horizon. The lever balance shifts with spend: whichever currency is starved dominates returns, a thin evaluator makes search look all-important, feeding it rebalances, and in reasoning the same logic produces the crossover, with size the efficient lever below the competence threshold and search the efficient lever above it. And the evaluator is consistently what binds, shown independently in every arm. Training creates information; search extracts it; neither creates what an external oracle must supply.

A footnote vignette: asked to play raw chess against Paper I's ladder, a frontier general LLM (Kimi-K2.5) beats a random mover barely more than half the time, scores 0% against Stockfish-1320, and often cannot emit a legal move, while Paper I's 3.45M-parameter specialist plays at the ~2800 ladder band with search. We draw no cross-domain GELO number from this; the qualitative point is enough. Task capability is bought by the right evaluator plus search over it, not by scale alone.

10. Registered predictions, scored

This research program's operating rule, inherited from Paper I's measurement-artifact autopsies, is that no delta is believed until it survives a low-variance re-measurement. Its prospective form is the registered prediction: state what the framework expects before the run, then score it. Four predictions have been scored so far.

prediction (registered before the run)	outcome
The single-grid $h = 3$ dip at $\sigma = 0.25$ is real (an optimizer's-curse mechanism)	Miss. The eight-grid average shows the dip was noise; search is monotone in h at every imperfect σ (§3)
A powered arena will restore the GSM8K ordering (3B above 1.5B)	Miss, and the more interesting outcome. At 84% judge agreement, 1.5B and 3B genuinely tie on MATH; the mid-ladder ordering is benchmark-dependent (§2)
MATH will widen the crossover region and keep the search lift alive further up the ladder	Hit. The flip widens and reaches significance at $n = 500$, and the lift stays alive to +6.0 at 32B against GSM8K's +0.7 at the same size (§4)
The crossover point will move up in model size on the harder benchmark	Miss, in the informative direction. The threshold sits at ~7B on both benchmarks; its location is benchmark-robust and only its sharpness is difficulty-dependent (§4)

prediction (registered before the run)	outcome
The strict crossover flips (smaller model + search beats the next size decoding greedily) are individually significant	Miss at n = 300–500 exploratory, then resolved (next row). Paired McNemar first returned all three flips as ties; parity was reported, the lift collapse untouched, and the deciding rerun registered (§4)
The larger MATH rerun will decide the strict question (registered when parity was scored)	Hit — decided below the predicted n. Significant already at n = 500 ($p = 0.0027$), where the 24-vs-14 $\rightarrow$ 39-vs-16 discordant ratio predicted $p \approx 0.03$ (§4)
A live search lift at 32B on MATH (+3–5 predicted vs GSM8K's +0.7)	Hit on liveness; magnitude above the registered band. MATH 32B lift = +6.0 against the predicted +3–5 (pass@1 75.0 $\rightarrow$ sc@16 81.0); search stays alive on the unsaturated benchmark where GSM8K's collapsed to +0.7. Overshoots are flagged with the same rigor as undershoots (§4)
§5's proposition: consensus saturates at the modal fraction while oracle-best-of-N tracks coverage	Hit, after correcting our own test. As registered (sc@32 vs modal fraction) the check is an identity at $N = 32$ and confirms nothing — flagged and rescored. The informative form holds: finite-N majorities converge to the modal fraction at every model size, oracle keeps climbing, and the coverage-minus-mode gap reproduces at n = 500, scaling with model weakness (+3.8 at 7B to +38.8 at 0.5B). The out-of-sample split-half test now scores it clean: mode(1–16) predicts vote(17–32) within noise for every model ($\max \Delta = 1.0$ pt) (§5)

Most of the scored predictions replaced a hypothesis with a cleaner claim, which is the point of scoring them. The most consequential is the crossover: the paper flagged its modest flip margins as its weakest joint, queued the exact paired test that would find the problem, and the test demoted the strict ordering to parity as soon as the caches landed; then the registered extension re-ran it at n = 500 with true-greedy comparators and *resolved* it — significant on unsaturated MATH, still a tie on saturated GSM8K. Registered, demoted, resolved, in the open, with a public timestamp on each step — which is the credibility mechanism, not a blemish on it. The remaining queue — a true n $\approx$ 1,000 MATH rerun for a tighter interval, the graded-verifier and judge-asymmetry §5 recomputes, and the GELO held-out/transitivity ledger — will be scored the same way.

Related work

The evaluator-plus-search structure is the AlphaZero family [Silver et al. 2016; 2017; 2018] built on MCTS [Coulom 2006; Kocsis & Szepesvári 2006; Browne et al. 2012] and PUCT

[Rosin 2011], with expert iteration [Anthony et al. 2017] as its self-play loop and KataGo [Wu 2019] as a strong open reference. The inference-time-compute view of reasoning — o1/R1 [OpenAI 2024; DeepSeek-AI 2025], self-consistency [Wang et al. 2023], tree-of-thought [Yao et al. 2023], STaR-style bootstrapping [Zelikman et al. 2022] — is the same lever at frontier scale; Jones [2021] documents test-time-compute scaling in board games.

Three lines are closest to our reasoning results, and we position *against* them rather than merely alongside — for each, what they establish and what the controlled/cross-domain version adds. **Compute-optimal test-time scaling** — Snell et al. [2024] (test-time compute can beat extra parameters, regime-dependently) and Brown et al. [2024] (coverage scales with repeated sampling over four orders of magnitude) — establishes the *above-threshold* half of our frontier and implies a competence boundary; we identify that boundary as base competence, derive it from the evaluator $\times$ search decomposition, and reproduce it under a *perfect oracle* in the gridworld where no LLM confound exists (§4–§5). Brown et al.'s "coverage scales, precision does not" is our +14.2 verifier gap at scale, so their large-n result agrees with our small-n one. **Verifier / process-reward supervision** — Cobbe et al. [2021], Lightman et al. [2023] — establishes that a verifier beats consensus; our contribution is the *gradient* (capability smooth in verifier accuracy q , no threshold) and the policy-vs-judge *asymmetry* (search helps a policy $\sim 4\times$ more than a judge) (§4). **Self-improving language models** — ReST [Gulcehre et al. 2023], Self-Rewarding LMs [Yuan et al. 2024] — report self-improvement that works; we contribute the *controlled isolation* showing the signal *source* is the operative variable, with an external-oracle positive control the at-scale results do not isolate (§6).

Our capability scale is the classical logistic latent-ability model shared by Elo, Bradley–Terry, and item-response theory (Rasch); pairwise LLM rating with a judge is the LMSYS-Arena approach [Zheng et al. 2023]. Benchmarks: GSM8K [Cobbe et al. 2021] and MATH [Hendrycks et al. 2021].

11. Limitations and future work

Sample sizes are stated per experiment, and the exploratory results of §5 remain the weakest numbers in the paper until the cache recomputation lands. GELO is a common scale, not a universal difficulty: the shared constants make curve shapes and lifts comparable across domains, and absolute cross-domain levels are never claimed — the benchmark-dependence result of §2 is a reminder of why. Reasoning search is whole-answer only; process-reward tree search over reasoning steps is the richest form and the natural next experiment. The language data lever never produced a fine-tune above the instruct model's zero-shot floor, and the bf16 base run, while clean, is one model at one size; a larger base and larger data are the untested routes to a from-floor curve that crosses the pretrained one. The gridworld is a minimal model by design. All policies here are non-reasoning base and instruct models; whether RLVR-trained reasoning models, which internalize the serial search axis into the policy, preserve the

decomposition is an open question on which the framework takes a position — an internalized searcher extracts more of its own evaluator, so the crossover should shift down in model size and the external-search lift should shrink further, a registered prediction for when such models fit our hardware. And four domains is not universality: Go, continuous and robotic control, and code are the obvious next tests, and the framework predicts both the same evaluator bound in each and where it should shift.

12. Conclusion

Across the four domains tested, capability decomposed the same way. A single-pass evaluator sets the level; search multiplies it and then saturates; the efficient lever flips at a measurable competence threshold whose location, in reasoning, is benchmark-robust; and only external information raises the ceiling. The compact statement remains Figure 1, and the compact evidence is the lever-limit map: thirteen experiments, including the failures, each diagnosing the same chain. Training creates information, search extracts it, and neither creates what an external oracle must supply. Within the domains and budgets studied here, the evaluator — not parameters, and not the search budget — is what binds.

Reproducibility

Code and data generators for all three arms are in the repo. Control (`control/gridworld.py`): the MDP, exact value iteration, the noisy-evaluator $\times$ lookahead sweep — pure NumPy, exact oracle. Games (`games/`): Connect-4 bitboard engine + solver, the depth-limited alpha-beta oracle, the conv net, MCTS, the GELO calibrator (`c4_calibrate.py` , which prints the goodness-of-fit gate), and the self-play / oracle-value loops. Reasoning (`reasoning/`): the accuracy-vs-N sweep, the evaluator-quality ablation and gradient, the MATH IRT fit, the pairwise arena + master-judge (Bedrock), the judge-scaling test, and the size $\times$ search frontier. Large datasets and model weights are regenerable and not committed.

Appendix A — Experimental setup, per result

So the "what we did" is explicit and reproducible, the exact knobs behind each headline number:

Arm A — Connect-4 / GELO (`games/`). - *Opponent ladder & calibration*

(`c4_calibrate.py` , `connect4_ab.py`): opponents are random plus depth- d alpha-beta ($d = 1\dots 6$) against the perfect solver's move ordering. Ratings are fit by logistic MLE from the full round-robin **cross-table** (not per-opponent win-rates); the goodness-of-fit gate is mean | predicted – observed | pairwise win-rate = **0.058**; anchor **random := 0**. - *Evaluator net* (`c4_net.py`): a small conv policy+value network trained on **oracle-solver labels** (value =

game-theoretic value, policy = solver move distribution). Data-size sweep uses 12k / 24k / 50k label subsets. - *Placing an agent* (`place_agent`): the agent plays **24–40 games per rung** vs each ladder rung; the win-rate vector is fit to the ladder's GELO by the same logistic model. Open-loop = greedy on the policy head; closed-loop = **PUCT MCTS, 200 simulations**. - *Which-lever diagnostic* (`c4_diagnostic.py`): a data (3k–50k) × capacity (small ≈0.3M / large ≈4M params) grid, each cell placed both open- and closed-loop, to read where data / capacity / search each bind.

Arm B — reasoning (`reasoning/`), all Qwen2.5-Instruct-4bit under MLX. - *Self-consistency & oracle-best-of-N* (`reason_math_sweep.py`): **Qwen-4B**, GSM8K test, **120 problems**, temperature **0.8**, **1024**-token budget, $N \in \{1,4,16,32\}$; final answer via `####<number>` regex; oracle-best-of-N = "is any of the N correct vs gold." - *Graded verifier* (same samples): a synthetic selector with per-item accuracy $q \in \{0.5...1.0\}$. - *Kimi-best-of-N* (`reason_bestofn.py`): **Qwen-1.5B**, **60 problems**, temp 0.8, $N \in \{2,4,8\}$; selector = **Kimi-K2.5** (Bedrock, temp 0) picking the correct candidate index — blinded to the gold. - *Judge scaling* (`reason_arena.py` judge, majority over repeats): Kimi judgments aggregated over $N \in \{1...5\}$; agreement measured against the ground-truth verifier on decisive pairs. - *Serial axis* (`reason_serial.py`): **greedy** (temp 0), max-tokens $\in \{128,256,512,1024,2048\}$, Qwen-1.5B and 3B. - *Size × search frontier*: Qwen **{0.5B,1.5B,3B}**, GSM8K, greedy + sc@{4,16}; compute proxy $\approx$ params × N. - *Pairwise reasoning-GELO* (`reason_arena.py`): **5 models on 50 MATH problems; every** ordered pair scored by the Kimi-K2.5 judge (**blinded, answer-order randomized**); ratings by Bradley–Terry MLE (`bt_elo`), anchor **Kimi := 2800**; the verifier cross-check gives the 72% judge-agreement figure. A powered rerun (`reason_arena.py` , resilient concurrent Bedrock judging: shared client with timeouts/retries, thread pool) over 150 MATH problems × the 5-model Qwen2.5 ladder gives **84%** agreement. - *Frontier at scale + MATH crossover* (`reason_cache.py` , `run_cache_ladder.sh`): sample-once-select-many caches — 32 completions/problem via `batch_generate` , then sc@{1,4,16,32}, oracle-best-of-N, and the graded-verifier curve computed post-hoc. GSM8K ladder 0.5B→72B at n=500; MATH ladder (`--math` , boxed extraction) 0.5B→14B at n=300. - *Difficulty-anchored GELO* (`reason_gelo_irt.py`): the same models vs MATH difficulty tiers L1–L5, Rasch (1-parameter IRT) fit. - *Training-data lever* (`reason_finetune_sweep.py`): LoRA fine-tune **two 0.5B models** — Qwen2.5-0.5B-Instruct (4-bit) and Qwen2.5-0.5B base (**bf16**; 4-bit was unstable) — (8 layers, lr 5e-5) on $N \in \{64,256,1024,4096,7000 \text{ (full)}\}$ GSM8K examples at **fixed 3 epochs** (so iters scale with N — isolates *data*, not optimization budget), calculator-annotation targets stripped; greedy accuracy on 150 held-out problems, N=0 = base model.

Arm C — control (`control/gridworld.py`). 8×8 gridworld, slip 0.1, $\gamma = 0.95$, scattered pits; exact V^* by value iteration; degraded evaluator = $V^* + \sigma \cdot (\text{value-spread}) \cdot \mathcal{N}(0,1)$; MPC lookahead $h \in \{1,2,3\}$ (h Bellman backups, then greedy). Return = mean discounted reward over **400 episodes per grid, averaged over 8 independent grid instances (mean ± sd)**, normalized to [random = 0, optimal = 100].

§6 self-improvement. Connect-4 (`c4_selfplay.py`): self-play games each iteration; value target = game **outcome** (baseline) vs **depth-6 oracle value** (`--oracle-value` , the positive control), loop otherwise identical; strength re-placed vs the ladder every iteration. Chess uses the analogous loop from three seeds (from-scratch / weak-policy warm-start / a self-learned +1214 net).

Infrastructure. Two **Apple M3 Ultra** machines (each 275 GB unified memory, ~819 GB/s): **MLX** / **mlx-lm** for all Qwen inference and net training — batched sampling (`batch_generate`) makes 32-completion caches and models up to **72B** tractable; **AWS Bedrock** (`moonshotai.kimi-k2.5` , us-east-1) for the master judge and the frontier-LLM chess vignette. Early exploratory sweeps used modest samples (50–120 problems); the powered reruns so far are §3 (multi-seed), §5 (eight-grid averaging), and the §4 size × search frontier (500-problem, 32-sample caches, 0.5B–72B), while the remaining §4 reasoning sweeps are still at exploratory power pending recompute from the caches — we flag exploratory-power results where they appear.

References

- Alsakka, L. (2026). *Efficient Thinking I: Measuring What Capability Costs*. This series.
- Anthony, T., Tian, Z. & Barber, D. (2017). *Thinking Fast and Slow with Deep Learning and Tree Search*. NeurIPS.
- Brown, B. et al. (2024). *Large Language Monkeys: Scaling Inference Compute with Repeated Sampling*. arXiv:2407.21787.
- Browne, C. et al. (2012). *A Survey of Monte Carlo Tree Search Methods*. IEEE TCIAIG.
- Cobbe, K. et al. (2021). *Training Verifiers to Solve Math Word Problems (GSM8K)*. arXiv:2110.14168.
- Coulom, R. (2006). *Efficient Selectivity and Backup Operators in Monte-Carlo Tree Search*. Computers and Games.
- DeepSeek-AI (2025). *DeepSeek-R1: Incentivizing Reasoning Capability in LLMs via RL*. arXiv:2501.12948.
- Efroni, Y., Dalal, G., Scherrer, B. & Mannor, S. (2018). *Beyond the One-Step Greedy Approach in Reinforcement Learning*. ICML.
- Gulcehre, C. et al. (2023). *Reinforced Self-Training (ReST) for Language Modeling*. arXiv:2308.08998.
- Hendrycks, D. et al. (2021). *Measuring Mathematical Problem Solving with the MATH Dataset*. NeurIPS.
- Jones, A. L. (2021). *Scaling Scaling Laws with Board Games*. arXiv:2104.03113.
- Kocsis, L. & Szepesvári, C. (2006). *Bandit Based Monte-Carlo Planning (UCT)*. ECML.
- Lightman, H. et al. (2023). *Let's Verify Step by Step*. arXiv:2305.20050 (ICLR 2024).
- OpenAI (2024). *Learning to Reason with LLMs (o1)*.
- Rosin, C. D. (2011). *Multi-Armed Bandits with Episode Context (PUCT)*. Ann. Math. AI.

- Singh, S. P. & Yee, R. C. (1994). *An Upper Bound on the Loss from Approximate Optimal-Value Functions*. Machine Learning.
- Silver, D. et al. (2016; 2017; 2018). *Mastering Go / Go without Human Knowledge / a General RL Algorithm* (AlphaGo, AlphaGo Zero, AlphaZero). Nature; Nature; Science.
- Snell, C., Lee, J., Xu, K. & Kumar, A. (2024). *Scaling LLM Test-Time Compute Optimally can be More Effective than Scaling Model Parameters*. arXiv:2408.03314.
- Wang, X. et al. (2023). *Self-Consistency Improves Chain-of-Thought Reasoning*. ICLR.
- Wu, D. J. (2019). *Accelerating Self-Play Learning in Go* (KataGo). arXiv:1902.10565.
- Yao, S. et al. (2023). *Tree of Thoughts*. NeurIPS.
- Yuan, W. et al. (2024). *Self-Rewarding Language Models*. arXiv:2401.10020.
- Zelikman, Y. et al. (2022). *STaR: Bootstrapping Reasoning with Reasoning*. NeurIPS.
- Zheng, L. et al. (2023). *Judging LLM-as-a-Judge with MT-Bench and Chatbot Arena*. NeurIPS.
- **Software/data:** Stockfish · MLX · mlx-lm · AWS Bedrock (Kimi-K2.5) · Lichess cloud evals · HuggingFaceH4/MATH-500.