SEEDS DO NOT REPLICATE RUNS: THE UNIT OF REPLICATION, THE ESTIMAND AND THE BUDGET RULE IN AGENT RL

Anonymous authors
Paper under double-blind review

ABSTRACT

Agent RL reports a credit-assignment comparison as "A beats B by X over three seeds". Three choices hide in that sentence, and on one pool of runs we measure what each decides. First, a seed argument is not a unit of replication. Whether re-running it reproduces the run is decided below the library: of the TRL and verl-agent configurations we census, only in-process generation replays to the bit, on one model and task, and our own loop replays only with its nondeterministic kernels off. Within one launch, our registered designs leave the seed's share of run-to-run variance unresolved, with upper ends near a quarter. Across launches, a re-executed seed argument moves the endpoint as far as changing it. On the released verl-agent stack, sixteen seed arguments put the seed's share at 0.1518, at most 0.470, at the final evaluation. Second, the size of a horizon effect is a property of the estimand and the budget rule. On KnobChain, a corridor whose length we set, 384 paired runs of Qwen2.5-7B-Instruct with a turn-level credit rule against an episode-level one show the endpoint gap shrinking nearly threefold with the horizon under a budget that grows with it. The shrinkage belongs to the reading: the method arm sits at the ceiling of the success scale in five of six depths; at a fixed update count the gap does not shrink; the rate at which runs first reach the success threshold shows no resolvable trend; and off the ceiling, at constant anchor coverage, the endpoint gap grows, factor 0.44 $[0.26, 0.70]$. On the field's own stack the endpoint and hazard readings differ in size, log-difference 0.7507 $[0.2259, 1.9358]$. That experiment reproduces the published growth under both estimands. The separation holds at the registered threshold and not below it. Third, three runs per arm resolve neither the effect nor the horizon trend, and the effect itself is not in question: under this dense shaped reward the method multiplies the takeoff hazard by two to three at every horizon. Report replications in runs rather than seed arguments, name the estimand and the budget rule, and fix the success threshold before training. The environment and the code accompany this submission; the run pool and the registration record will be released with the paper.

1 INTRODUCTION

A large and fast-growing literature compares credit-assignment schemes for reinforcement learning with language-model agents, and reports the comparison in a sentence of fixed shape: *method A beats baseline B by X, mean $\pm$ standard deviation over three "seeds"*. In a survey of 78 LLM RL post-training papers (Appendix K), 71.8% state no per-configuration replication count (double-coded by two language-model assistants, $\kappa = 0.86$); where one is stated the modal count is three ("Results are averaged over 3 random seeds", Feng et al. 2025, Table 1), at which an exact paired sign or permutation test cannot return a two-sided p below 0.25 (§6.1). That the same seed re-run on one GPU can scatter almost as widely as a change of seed was noted by Melis et al. (2018, §7.2); its share on an LLM-agent RL stack has not been measured.

The two published comparisons that stratify by task horizon report growth, and a survey argues for it on signal-to-noise grounds (Wang et al., 2025; Tang et al., 2026; Zhang, 2026). Our claim is narrower: it is about that sentence, not the method. What does that sentence measure? Our main finding is that its two hidden choices decide what a reported horizon effect is, and that on the stacks we census a

Reviewers: please read the Reviewer Guidelines (iclr.cc/Conferences/2027/ReviewerGuidelines) and the AI Policy for Reviewers (iclr.cc/Conferences/2027/AIPolicyForReviewers). If you used AI to expand, edit, or polish your review, please provide the input text to the LLM. Better still, consider skipping the LLM and submitting your original text: we, and the authors, are much more interested in your unedited thoughts than in what an LLM has to say. AI-assisted or not, you are putting your name and reputation behind your review: LLM-generated falsehoods, hallucinations or misrepresentations are subject to disciplinary action, which may include desk-rejecting all papers you have authored.

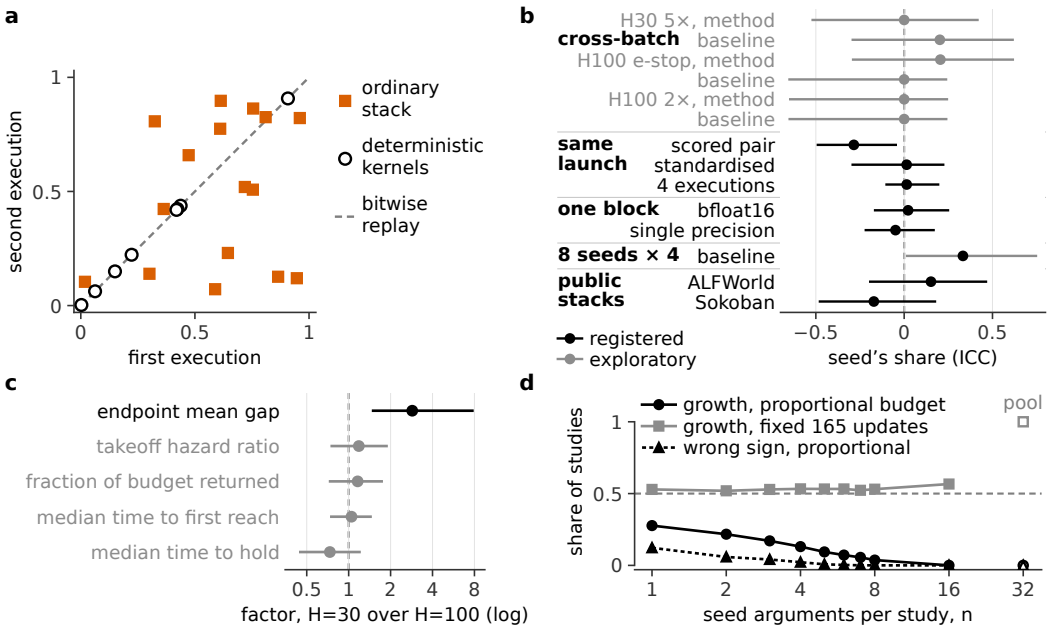


Figure 1: A seed argument is not a unit of replication, the size of the horizon effect is a property of the reading, and three runs resolve neither. (a) Two executions of one seed argument on the ordinary stack and under deterministic kernels; dashed: a bitwise replay. (b) Every printed interval on the seed's share of run-to-run variance, by design; black registered, grey exploratory. (c) The five horizon readings as factors, $H{=}30$ over $H{=}100$; only the endpoint reading excludes one. (d) The share of ten thousand simulated studies publishing growth, or the wrong sign, against seed arguments per arm.

seed argument replays a run in one configuration only. We call the two choices its joints, because the sentence's meaning turns on them without saying so. "Over three seeds" hides a unit of replication: what population the printed interval is over. "Beats B by X" hides an estimand and a budget rule: what X is a difference in, and at what training budget it is read. On one pool of runs the second joint decides whether the benefit of finer credit is read as shrinking with the horizon, holding or growing, and on which scale; the first decides what a seed buys (Figure 1).

What we ran. KnobChain is a corridor of rooms whose length we set; we train Qwen2.5-7B-Instruct (Qwen Team et al., 2024) on it with two credit rules and nothing else different, a GiGPO-style turn-level advantage against GRPO's episode-level advantage broadcast to every turn (Shao et al., 2024): 384 paired training runs, six horizons, thirty-two pairs per horizon. Throughout, a *seed argument* is the integer a run is launched with, an *execution* is one run of a fixed argument list, and a *launch batch* is a set of runs started together on one code revision. Three later launch batches re-execute runs of the pool with their arguments unchanged; one fork of the training loop switches the framework's nondeterministic kernels off; verl-agent on ALFWorld (Shridhar et al., 2021) and TRL (von Werra et al., 2020) on GSM8K (Cobbe et al., 2021) and MATH (Hendrycks et al., 2021) carry the same two-execution test.

The unit of replication. One clause of that sentence is "± standard deviation over three seeds". Its implicit promise is that fixing the seed fixes the run, so that a released seed reproduces it and pairing arms by seed, as the repeated-measures design of Patterson et al. (2024, §4.4) does, buys precision. Three runs with three seeds are three draws of the run whichever way that promise falls; what the promise decides is what a seed buys beyond that. Whether it holds is decided below the library: TRL with generation in the training process replays to the bit for one model and task and not for two others, and its served default and the released verl-agent stack do not (Table 2; Appendix B.6). On our own loop, in six cells (a *cell* is one horizon and one arm; Table 1), the same computation run twice scatters about as widely as different seed arguments do, so the share of apparent seed variance that is not the seed's is at or above 0.79 in every cell. With the framework's nondeterministic kernels

switched off the same-seed spread collapses to zero (§3). Those six cells compare an execution with a later launch batch; two designs registered to compare inside one launch do not resolve the seed's share, and their upper ends are near a quarter (§3). On the public stack, with sixteen seed arguments each run twice, the seed's share is 0.1518, at most 0.470, at the final evaluation. So a published seed argument does not reproduce a curve: "three seeds" samples three draws of the training noise on one configuration and three exact replays on another.

The estimand and the budget rule. The other clause is "beats B by X", which says neither what X is a difference in nor over what budget it is read. At the registered proportional budget (5.4–$5.5 \times H$ updates per cell), on a seed-argument bootstrap drawn jointly across arms and depths, the benefit from $H{=}30$ to $H{=}100$ shrinks by $2.86\times$ $[1.47, 7.90]$ when read as the endpoint mean difference the field reports, and by no resolvable factor when read as the takeoff hazard, the rate at which runs first reach a success level ($1.18\times$ $[0.74, 1.91]$), as the median time to first reach it ($1.05\times$ $[0.74, 1.47]$), as the fraction of the budget returned ($1.16\times$ $[0.72, 1.76]$), or $0.73\times$ $[0.44, 1.22]$ read as the median time to *hold* that level. Only the first interval excludes one, and the paired difference of their log-decays is 0.88, 95% $[0.25, 1.81]$ for the endpoint and hazard readings. That decay is a property of the reading, not of the method: the method arm sits at or above 0.90 success in five of six depths, so the decay is read against a ceiling (§4); at a fixed 165 updates the same runs show no decay in the difference of success rates (§4.1); and off the ceiling, with the corridor's pool of room cards held constant across depths, the endpoint gap grows (§4). The published growth comes from a mixed-horizon design read with the method arm at or above 0.90 in four of six task types (§7); our ALFWorld comparison reproduces it under both estimands (§5.3).

The effect and its reward regime. Neither joint is a claim that turn-level credit assignment does not work. Under this corridor's dense shaped reward the method multiplies the takeoff hazard by 2.29 to 3.25 at every horizon and returns 18.5% to 26.0% of the training steps. Under a purely terminal reward the sign can be read only at the top of the scale (§5).

Scope. Both joints are measured in full on one training stack, one model, one environment family and one hardware generation, against the GRPO credit rule rather than a published method's full recipe; the public stacks carry only the two-execution test and one horizon comparison (§5.3), and the replication reading, measured on two of the five further models we trained, reproduces on both (Appendix B.4). The pool carries no blind confirmatory test of the horizon joint: the prediction registered for it, on a second model family, checks the mechanism at the ceiling (Appendix J.2; Appendix N). Both joints are known in general (§7); what is new is their measurement on LLM-agent RL stacks and the census of what replays, and what transfers is the argument, not the numbers (§8).

2 Apparatus

KnobChain is a corridor of H rooms walked one turn at a time. At each turn the agent sees a *room card* and must pick the door that an episode-level *rule card* prescribes; its answer is a door number, two tokens. A correct choice advances one room; a wrong one ends the episode with probability 0.25 and otherwise pushes the agent back one room. Success is reaching room H, so a success takes at least H correct doors. Sixteen episodes make one update; an evaluation point is 256 fresh episodes at temperature 0.7. The pool of room cards a depth can show grows with depth at a rate set by λ (0.95 in the pool; no growth at $\lambda = 1.00$), so *anchor coverage*, the share of turns whose card recurs on another trajectory and can be grouped by a turn-level estimator, falls with depth at 0.95 and is constant at 1.00.

The pool spans six horizons, $H \in \{30, 40, 55, 70, 85, 100\}$, and two arms: a turn-level estimator that groups turns by exact observation hash, the GiGPO credit rule (Feng et al., 2025) in the same loop, not its published recipe, against the GRPO credit rule, whose turn-level credit content is zero by construction; thirty-two paired runs per arm, 384 training runs. The update budget is a fixed multiple c^* of the horizon, 5.4–$5.5 \times H$ updates in the pool, so runs get from 165 to 540 updates of sixteen episodes ($H{=}30$ to $H{=}100$) and the number of environment turns grows faster still (§4.1). The stack is one archived loop, a group-normalised token-level policy gradient with no ratio clip, no KL term and no critic, with a shaped reward, 1 per correct advance and $0.5H$ for the final room. Its weights are held in bfloat16 with no single-precision master copy, at a constant learning rate: the loop's own

default, a regime in which most updates round away and which the published stacks do not share, so every reading of this pool is conditional on it (Appendix A, §8). The two arms of a pair share the seed argument and the stream of task instances, which is all the pairing buys (Appendix A.5).

We read a run's outcome at its *endpoint window*, the three evaluation points in the last fifth of its updates, or at the final evaluation alone; on the public stack the window is the last six evaluation points. A run *takes off* at the first evaluation point at which its success rate reaches a threshold θ fixed before training (0.70 here), and a run that never reaches it is censored at its budget; the *hazard ratio*, method over baseline, compares how fast the two arms' runs take off (Appendix F). Where both arm means exceed 0.90 we do not read a gap: that *ceiling gate* was fixed before any row was scored, and it is not the scale's own ceiling of one.

Three questions about a seed argument are kept apart in what follows: whether re-executing it replays the run to the bit, how much of run-to-run variance it accounts for, and whether pairing the arms by it buys precision. We measure the *seed's share* of run-to-run variance as the intraclass correlation between two executions of one seed argument (Bouthillier et al., 2021, Eq. 7): fixing the seed does not remove its variance from an estimate, it correlates the runs. Its complement, the *nondeterminism share*, is the variance between two executions of one seed argument divided by the variance across seed arguments: 0 is a bitwise replay, 1 says re-running a seed argument moves the outcome as far as changing it, and above 1 it moves it further.

We fixed a matrix of 45 cells around the six horizons before any of its rows was scored. A *registered prediction* is a sentence with its threshold, filed to an append-only, hash-stamped register before the runs that score it exist; we score it as a hit, a miss, or as neither (Appendix M).

3 Whether the Seed Is a Unit of Replication

Three of our later launch batches re-execute runs of the pool with their arguments unchanged, and none is a bitwise replay: all 32 step-0 records match exactly, and every training-derived field then diverges from the first or second training step on (Appendix B). Each of these experiments compares an execution with one from a later launch batch on a later code revision, so the intraclass correlations of Table 1 are cross-batch readings (Figure 1a plots one of the cells beside the deterministic-kernel fork, and Figure 1b every interval on the seed's share this paper prints).

Table 1: Re-executing a seed argument scatters the endpoint at least as widely as changing it in four of the six cells. Two executions of the same argument list, read on the shared evaluation grid. σ_{rep} is the standard deviation of the paired difference between the two executions divided by $\sqrt{2}$; σ_{tot} is the standard deviation across seed arguments (the mean of the two batches', which overstates the nondeterminism share by up to 11% in our own favour); the nondeterminism share is their squared ratio, one minus the intraclass correlation before the clamp at zero; sign counts the pairs, of those with a non-zero arm contrast, in which the contrast keeps its sign. Experiments: $H{=}30$ and $H{=}100$ at five and at twice their budgets, and early stop (e-stop): $H{=}100$ read at 165 updates against its own 540-update runs at the same steps.

experiment	arm	σ_{rep}	σ_{tot}	ICC	nd. share	sign
$H30$, $5\times$	method	0.171	0.165	0.000	1.071	12/16
$H30$, $5\times$	baseline	0.258	0.289	0.202	0.798	12/16
$H100$, e-stop	method	0.301	0.337	0.205	0.795	12/13
$H100$, e-stop	baseline	0.275	0.244	0.000	1.266	12/13
$H100$, $2\times$	method	0.064	0.057	0.000	1.260	11/16
$H100$, $2\times$	baseline	0.227	0.201	0.000	1.265	11/16

We find that re-executing a seed argument scatters the endpoint at least nine tenths as widely, in standard deviation, as changing the seed argument in every cell, and at least as widely in four of the six (Table 1): the nondeterminism share at the endpoint window is at or above 0.79 in all six cells and at or above one in four. Pooled at that window, the four pooling rules put at most 0.17 to 0.30 of the endpoint variance on the seed argument (Appendix B). A batch or code-revision difference enters the within-seed term and lowers the correlation, so these shares are jointly about the launch batch and the stack (Appendix B; §8).

These are point estimates and not bounds: no cell's exact F interval excludes an intraclass correlation of 0.24, and the F test of no seed component rejects in none of the six (Appendix B.1). The arm contrast keeps its sign in 11 to 12 of the pairs on which it is non-zero (Table 1), no more often than two independent executions with the same sign rate would agree. What the seed joint costs is the curve and the pairing, not the sign.

These readings come from launch batches run for other reasons, so we registered a five-part programme in advance (Appendix B), with an intervention test: at $H{=}30$, switching framework-level nondeterminism off collapses the same-seed spread to zero and keeps the arm contrast's sign in eight of eight pairs. With the same kernels switched off on two different hosts the replay still holds to the bit in sixteen of sixteen pairs (Appendix B.3). The switch is not shown neutral (sixteen pairs resolve neither a shift of the baseline nor of the contrast; Appendix J.4), so what the test established is the sign. Two further models and a terminal-reward cell at $H{=}30$ reproduce the reading (Appendix B.4).

One of the programme's five scored predictions is scored against us: in the eight-seed four-execution experiment the intraclass-correlation form misses in the baseline arm, at 0.3327 against a registered 0.30 (a seed component, exact interval $[0.010, 0.753]$), and clears in the method arm at 0.0000, an arm at the ceiling where eight seed means have no room to differ.

Those six cells compare an execution with a later launch batch, so two designs were registered, before any of their runs existed, to make the comparison inside one launch (Figure 1b). Neither resolves the seed's share. The first re-executes sixteen seed arguments per arm on one code revision at $H{=}30$ and $H{=}100$, with four executions at $H{=}30$, meant to launch as one batch. Its scored reading, the pair of executions pooled over both arms and both depths, is the one scored interval in this paper that lies below zero, $[-0.4955, -0.0406]$ at the registered endpoint window ($[-0.3345, 0.1536]$ at the final evaluation). It lies below zero only on the equal-variance reference, which strata whose within-seed mean squares differ eighty-fourfold do not meet: the standardised pooling registered beside it contains zero at both readings, and the pooled estimate is carried by the $H{=}30$ baseline stratum, whose own interval contains zero, so no share is read from it. The four executions at $H{=}30$, read together and balanced, put the correlation near zero with an upper end of 0.20 at the window (registered, unscored). The second design re-executes the $H{=}30$ half as one launch block of three executions per seed argument and arm, interleaved at random; its interval, $[-0.1715, 0.2549]$, contains zero, in both arms. Read together, the registered same-launch designs put the upper end of the seed's share at the endpoint window between 0.20 and 0.26, and none resolves a positive share.

Two exploratory readings say why the first design is hard to read. At $H{=}30$ every run of its first execution was launched before any run of the other three; pooled over executions two to four the correlation is $+0.29$ $[+0.07, +0.52]$ at the window, against -0.17 $[-0.30, +0.03]$ over executions one to three, and a reading that models the launch block puts the seed's within-block share at 0.03 to 0.07 (Appendix C). The public stacks read the same quantity on sixteen seed arguments by two executions: ALFWorld and Sokoban put its upper end at 0.470 and 0.181, and only those upper ends are read as bounds (§5.3; Figure 4). The re-execution spread is not evaluation noise: the evaluation-sampling floor is below one per cent of the residual mean square in both launch blocks at $H{=}30$ and at most 8.89% in any stratum of the same-launch experiment (Appendix C.3, Appendix A.4).

Table 2 is the census of every configuration this project has run twice, the Sokoban recipe apart (§5.3), and our claim is scoped to it.

4 What the Endpoint Decay Is Made Of

We measure the endpoint gap at every depth. It falls from 0.318 at $H{=}30$ to 0.111 at $H{=}100$, a factor of $2.86\times$ $[1.47, 7.90]$. The six depth means show no resolvable rank trend, but the pair-level slope of the gap on $\ln H$ is negative (Appendix E.3), so the decay is real on this pool; what it is made of is the question. The method arm (Figure 2) sits at or above the 0.90 ceiling in five of six depths (the ceiling gate, fixed before any row was scored and applied to the pool after it had been read), while the baseline rises from 0.579 to 0.813 under a budget that grows with the horizon: the extra updates a deeper cell buys are spent by the baseline. A gap cannot exceed the room the baseline leaves below one, and an accounting identity on the relative failure reduction, the gap divided by the baseline's failure rate, puts 77.0% $[58.5\%, 113.9\%]$ of the log-decay in that shrinking room rather than in a change of efficacy.

Table 2: Of the four configurations, only the second library with generation in the training process replays to the bit; the shares say how far a re-execution moves the endpoint. Shares carry a 95% interval: an exact $F(7, 8)$ interval for the ALFWorld ANOVA share (Appendix D.1) and a bootstrap percentile band for the TRL rows (Appendix B.6). The public-stack row is read at its six-point window (0.5788 and 0.2761 at the final evaluation, Appendix D.1); designs are seed arguments $\times$ executions. Second-library rows are Qwen2.5-1.5B-Instruct at 100 updates unless named; their calibration detail is Table 19.

stack, configuration	replays?	share [95% interval] (design)	what restores a replay
this paper's pool: Qwen2.5-7B-Instruct on KnobChain	no	0.79 to 1.27 (six cells of 16×2; ICC upper ends 0.24 to 0.62); same-launch designs unresolved (§3)	—
the public stack as released: verl-agent, Qwen2.5-1.5B on ALFWorld	no	1.095 $[0.396, 1.691]$ / 0.861 $[0.267, 1.548]$ (8×2; the sixteen-seed experiment's values are Appendix D.2)	neither documented switch: 1.403 / 0.814
the second library, TRL, colocated generation in one process: Qwen2.5-1.5B on GSM8K	yes, to the bit	0 (8×2 per arm, 16 of 16 pairs)	— (Qwen2.5-0.5B and MATH do not replay: the full-determinism flag or eager attention, 2 of 2 each; Appendix B.6)
the same library's released default, a generation server in a second process	no	1.347 $[0.885, 1.943]$ / 0.490 $[0.087, 1.624]$ (8×3; shares read on two executions, 1.082 / 0.691 on three)	the engine kept in the serving process (35 of 35)

Off the ceiling and with anchor coverage held constant (λ=1.00, both arm means at or below 0.90 at both depths at c^*=2), the endpoint gap grows from H=30 to H=100, factor 0.44 $[0.26, 0.70]$ below one (Appendix E); the hazard reading there has fewer than four takeoff events in the method arm at H=30 (2 of 16) and is not evaluable. The headroom share is not scale-free: across the five efficacy scales of Table 21, listed before any was computed so that the reading is not a scale chosen after the result (Hofman et al., 2023; Schaeffer et al., 2023), it runs from 77% to -17%. The decay itself is also a property of the scale, the removable interaction of Loftus (1978) and Wagenmakers et al. (2012): on three of the five scales the endpoint decay is distinguishable from each of the other four readings and no two of those four are, and on the logit and the log failure ratio, where the two-point factor's interval touches or includes one, it is not (Table 29; Appendix F.2). That difference is a registered-budget result: off the registered multiple it narrows and then changes sign (Appendix F.2; the other two model families, Appendix J.3).

4.1 Both budget rules manufacture a trend on this pool

The pool's budget grows with the horizon (5.4–5.5 $\times$ H updates of sixteen episodes); the literature's rule, a fixed count, holds it fixed. Neither reads efficacy. Hold the budget at the shallowest cell's 165 updates at every depth and the baseline's rise with depth becomes a fall to 0.163, while the arm gap stops decaying in the difference of success rates: that "flat" sequence compares a mid-learning gap with an end-of-learning one, and at the same count the gap decays on the log failure ratio and grows on the log success ratio (Table 29; Appendix F). The decay of §4 belongs to the proportional budget we registered, and the standard report names neither rule.

4.2 Under the registered budget no success-based reading grows

Enumerating the conventions, the estimand family, the stopping rule (endpoint, peak, area under the curve or a fixed count) and the budget rule, gives twenty-four admissible specifications on the same pool, a specification curve in the sense of Simonsohn et al. (2020) and Steegen et al. (2016) (Appendix E.3); of the twenty that contrast the arms, only the four rooms-reached contrasts increase with the horizon at $\alpha = 0.05$ (Appendix E). That is the joint stated plainly: under the registered budget a horizon trend grows only if the rooms-reached count, whose ceiling is the horizon itself,

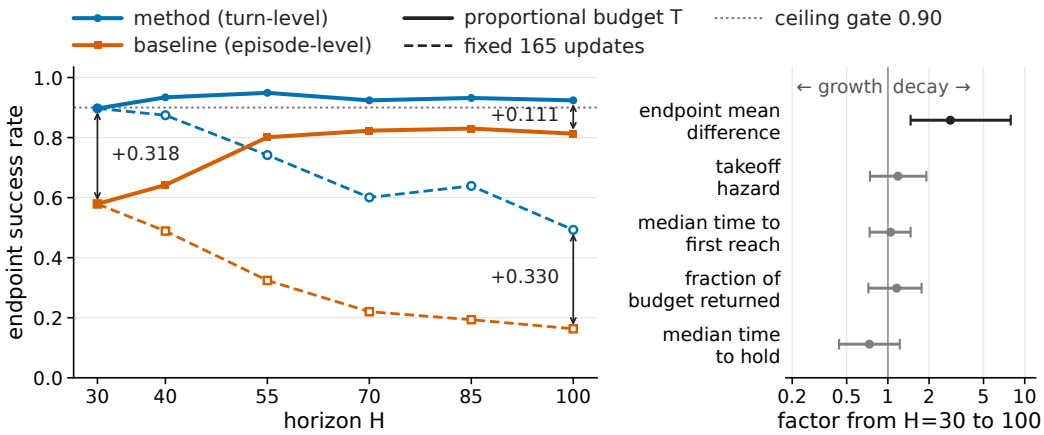


Figure 2: Under the proportional budget (solid) the method arm sits at or above the ceiling gate (dotted) at five of six depths while the baseline rises with the horizon; under a fixed count of 165 updates (dashed) both arms fall and the gap does not shrink (left). At the proportional budget, of five readings only the endpoint mean difference shrinks from $H{=}30$ to $H{=}100$ by a factor whose 95% interval excludes one (right; bootstrap of §1).

is called "the benefit". Remark 1 says why the endpoint curve cannot decide between speed and composition: *Every gap sequence in* $(0, 1)$ *compatible with two monotone outcome curves bounded in* $[0, 1]$*, and whose increases sum to at most* $\Delta(H_k)$, the gap at the deepest horizon, is reproduced exactly, under a proportional budget, both by a model in which the method only speeds learning up by a constant factor and by one in which the arms differ only in how many runs have taken off by the budget; neither uses an efficacy that depends on the horizon (Appendix G). The speed-up construction needs a rate factor of at least 3.33 on this pool and the pool's own fit gives 1.68 (Appendix G): the Remark says what an endpoint-only report cannot identify, not that the speed-up model holds.

5 What the Method Buys

5.1 On the event-time scale the effect is not resolvably different across depth

We find no resolvable change of the takeoff hazard across depth. That is a statement about what six cells can resolve, not a claim that the effect is constant: the method-over-baseline hazard ratio runs from 2.29 to 3.25 across the six depths ($\theta = 0.70$, right-censored at each run's budget), every 95% interval excluding one (Appendix F.1). At equal power the hazard reading needs thirteen paired seed arguments where the endpoint gap needs thirty-nine (Appendix H.2). The flatness holds at the pre-fixed threshold: at $\theta = 0.90$ the two-point factor reads 2.34 $[1.25, 5.56]$ (Appendix F.1).

5.2 What the method buys beyond speed

At five times the registered budget every run of both arms reaches $\theta = 0.90$ at least once, and the difference left is in holding it: all sixteen method runs stay above it, eight of sixteen baseline runs do (Appendix J.2). Under a terminal reward the anchored step-level advantage is still defined (Appendix O), so the terminal-reward cell sets a floor for the effect, not a boundary. At the registered budget neither arm had reached the takeoff threshold, a regime in which no sign can show; at three times that budget the cell reads 12 of 16 seed-paired gaps positive (exact binomial $p = 0.0768$; sixteen of sixteen seed pairs complete) with the method arm at the 0.90 ceiling, where a gap is compressed, and its prediction scored inconclusive (Appendix O). The public-stack comparison of §5.3 is itself an outcome-reward comparison (ten times the episode's success, paid when the task is won), and there the gap is positive at every task type (Appendix D.2).

Three registered perturbations change one thing at a time, a third estimator, a second model family (Llama-3.1-8B-Instruct; Grattafiori et al. 2024) and a further checkpoint: the first two read their

predictions as hits with the arm under test at or above 0.90 at both depths, so they check the mechanism at the ceiling, and the third was scored not evaluable at the ceiling gate (Appendix J; Appendix J.2).

5.3 On the field's own stack

We repeat both measurements on the field's own stack: verl-agent running GiGPO against GRPO on ALFWorld with Qwen2.5-1.5B, first with eight seed arguments and then with sixteen, each run twice. Its horizon reading is per task type, registered before any run completed, with the number of sub-goals a task type requires as the proxy for length. At eight seed arguments both readings grow with task length, the endpoint one significantly on a seed-argument bootstrap interval, and the experiment cannot tell the two readings apart (Appendix D.1). The sixteen-seed re-run, registered before any of its runs existed, can. At the recipe's own update count the endpoint gap's shortest-to-longest ratio is 0.4520 $[0.3401, 0.5906]$ and the takeoff hazard ratio's is 0.2134 $[0.0629, 0.3610]$; their log-difference, $D = 0.7507$, 95% $[0.2259, 1.9358]$ on a seed-argument bootstrap, excludes zero, so on a real benchmark the choice of estimand changes the size of the reported horizon effect (Appendix D.2). The separation holds at the registered threshold and not below it: at $\theta = 0.50$ to 0.65 the interval contains zero, and gated at the ceiling (no pick_and_place) it reads 0.40 $[-0.18, 1.60]$; both readings grow at every threshold, and no single baseline takeoff carries the separation.

The same runs read the seed joint at the final evaluation. The seed component is not separated from run-to-run scatter, ICC 0.1518, exact 95% $[-0.199, 0.470]$, and the design had one-sided power 0.89 against a true intraclass correlation of 0.48 and 0.53 against 0.30, so the reading bounds the component rather than merely failing to find it: at most 0.470 of run-to-run variance is controlled by the seed argument. Under the two other poolings the upper end reads 0.50 and 0.54 (Figure 1b; Appendix D.2). At the final evaluation, re-evaluating four saved checkpoints three times each does not resolve how much of the within-seed variance is evaluation sampling: 0.26 $[0.08, 1.08]$ in the method arm and 0.20 $[0.07, 0.83]$ in the baseline arm, and in the method arm the within-seed spread sits at the evaluation-sampling floor (Appendix D.2).

On the field's own second public environment, at the final evaluation, at least 0.819 of run-to-run variance is not the seed argument: the exact 95% interval on the seed component is $[-0.484, 0.181]$, whose upper end is below one half. The environment is Sokoban under the released recipe (Qwen2.5-VL-3B-Instruct, sixteen seed arguments by two executions per arm), and the horizon result is not tested there (Appendix D.4).

6 What Three Runs Cannot Show

6.1 A structural bound, and what each statistic costs

At three replications the printed spread is itself unstable: a three-draw standard deviation runs 11.4% low on average, and its interval for σ spans 0.52 to 6.29 times the printed value (Appendix H.1). A paired comparison scored by an exact sign or signed-rank permutation test has a smallest attainable two-sided p-value of 2^{1-n}, which is 0.25 at three, so under an exact test no three-replication study can be significant at $\alpha = 0.05$ (Colas et al., 2018; Jordan et al., 2020; Bowyer et al., 2025).

6.2 What the field reports, and ten thousand parallel literatures

We run the standard study ten thousand times on our pool at n runs per arm, and find that at $n = 3$ the outcome is a distribution over conclusions: 16.4% of simulated literatures publish the wrong horizon direction and 4.4% the wrong sign of the method (Appendix L.1). Under the fixed count, where the pool's own trend is flat, the published direction is a coin toss at every n up to 16; under the hazard reading 62% of the three-run studies that yield a factor publish a decay against the pool's factor of $1.18\times$ (Appendix L.2).

6.3 Three prescriptions

Four failures of the registry's own machinery are recorded in Appendix M, with the register's tally, a record of what was written down before the data and not a confirmatory weight: at three hundred and fifty register lines, of seventy-four scored predictions, thirty-nine hit, eleven missed, and twenty-four

returned an outcome that is neither. The prescriptions in full, with their costs, are Appendix I. **First**, count replications in runs rather than in seed arguments, power them for a stated detectable effect (Appendix H) and launch them in one batch on one stack: on this pool five runs miss five of the twenty-four largest comparisons, eight miss none of the sixteen matrix rows above a gap of 0.30 for about sixty per cent more compute, and a later run confounds the count with its launch batch. Report a mean-based and a rank-based test, and whatever the count, re-execute one configuration per arm twice, since only a third execution tells a low pair from a launch effect (§3). **Second**, fix the success threshold before training ($\theta = 0.70$ here) and print the sweep from 0.30 to 0.95. **Third**, write beside every registered assertion the cells and arms it covers.

7 Related Work

That implementation-level nondeterminism can rival seed randomness is not new (Summers & Dinneen, 2021; Zhuang et al., 2022; Pham et al., 2020), and on LLM stacks it is traced to batch-dependent kernels and removed (He & Thinking Machines Lab, 2025), with the served rollout engine's numerical gap to the trainer a related source (Yao et al., 2025; Qi et al., 2025; The vLLM and TorchTitan Teams, 2025); in deep RL, Nagarajan et al. (2018) find GPU nondeterminism comparable to the exploration and minibatch seeds and below the initialisation seed, and implementation choices decide results at scale (Engstrom et al., 2020; Andrychowicz et al., 2021). Our addition is that on an LLM-agent RL stack of this kind the re-execution spread reaches parity with the seed spread, so "mean $\pm$ std over 3 seeds" (e.g. Feng et al., 2025) quantifies a population it does not name; within one recent paper the same GRPO baseline appears at 72.8 ± 3.6 and at 60.9, two configurations under one name (Tang et al., 2026, Tables 1–3). More runs, interval reports and the reporting of what stands behind a number are argued for in general (Henderson et al., 2018; Colas et al., 2018; Lucic et al., 2018; Dodge et al., 2019; Agarwal et al., 2021; Bouthillier et al., 2021; Miller, 2024). The two published horizon-stratified comparisons (Wang et al., 2025; Tang et al., 2026) train one policy per arm for a fixed count on a mixed ALFWorld task set and stratify the evaluation afterwards, the method arm reading at least 0.90 in four of six task types (Tang et al. 2026, Table 1).

8 Limitations

Every training number of the horizon study comes from one synthetic environment family, on one core 7B model against a baseline estimator with zero turn-level credit content. The non-synthetic evidence is the public-stack comparisons of §5.3 and the released `stackcheck` tool (Appendix D.3). Three properties of the corridor bound the horizon reading: the dense reward; a method arm within 0.11 of a success rate of one, priced in §4 and following the method onto the public stack (Appendix N, Appendix D.2); and anchor coverage, which depth confounds at $\lambda{=}0.95$ (Appendix A) and the off-ceiling cell at $\lambda{=}1.00$ removes (§4, Appendix E). The registered proportional budget is not compute-matched (Appendix A). The cross-batch shares are jointly about the launch batch and the code revision: the registered pair of runs that would have told the two apart was never launched, the same-launch experiment fixes the code revision but not the launch, and its one-block re-run fixes both, at $H{=}30$ only (§3; Appendix N). The pool's loop holds its weights in bfloat16 with no single-precision master copy, and a first master-copy block at the pool's learning rate does not resolve whether the execution component depends on the master copy: the residual-variance ratio bfloat16 to single precision in the baseline arm is 2.6029, interval $[1.2550, 5.3982]$, at $H{=}30$, which excludes one but not the halving the registered prediction asks about, and the two blocks are read at different operating points (Appendix A.4).

A registered learning-rate pilot with a pre-fixed selection rule then found that the master-copy loop trains stably at 5×10^{-7} (four of four runs; Appendix N), and at that rate the execution component of the baseline arm does not halve: in one launch block of three executions the residual variance of the bfloat16 block is 0.7887 times that of the single-precision block, interval $[0.3803, 1.6357]$, at $H{=}30$, and within that block at most 0.1735 of the endpoint variance is the seed argument's and at least 0.8265 is not. The loss of sub-half-ulp updates is therefore not necessary for the execution-to-execution variance reported here. Single precision makes inference on these models reproducible (Yuan et al., 2025); with the framework's nondeterministic kernels on, it did not make training replay here. The second library's non-replay comes from the standard regime, single-precision master weights under bfloat16 autocast (Appendix A).

Reproducibility Statement

We will release, with the paper, the run pool, the instrument and the append-only registry in full, including the registered predictions that were scored against us, as one archive whose parts Appendix A.1 lists: the run records of every cell this paper reads, KnobChain and every training-stack revision it names, the register with the documents its lines pin, the scoring scripts and analysis scripts with the receipts they printed, and the record supplement. The supplementary material of this submission carries the archive's code; the run records, the register with its receipts and the record supplement, together the full record, will be released with the paper (Appendix A.1). Two verification programs in the supplementary code check every numeric claim of the main text against the receipt it comes from (Appendix M). We do not claim that releasing seed arguments reproduces our training curves; §3 is why not. Code identity is pinned by hash on every run.

Use of Generative AI

Generative AI tools, large language models used as coding and writing assistants, were used throughout this work, and we disclose their use by task. Among the uses for which disclosure is required: they helped draft the registration protocol of §2, including the pre-registered expectations; they implemented the analysis code, the scoring scripts and the programs that check this manuscript against its receipts, to the authors' specifications; they assisted in the derivations behind Remark 1 and its companion, Remark 3 (Appendix G), which the authors then checked by hand and, where a statement is constructive, against the numerical fits reported in the appendix; and they made both codings of the survey of §1, the second blind to the first, from the papers' full text, each row carrying the arXiv identifier and a verbatim evidence sentence. They were not used to generate data: every number in the paper is read from a training run or from a simulation whose code is released, from the survey of §1 or from the cited literature. The authors reviewed all AI-assisted work. In particular, every registered prediction was stamped before the runs that score it existed; two of them (Table 10) compare against anchor cells of the matrix fixed in advance that had already been read. We take responsibility for the final content of this work, including text, claims and artifacts produced with the aid of generative AI.

References

Rishabh Agarwal, Max Schwarzer, Pablo Samuel Castro, Aaron Courville, and Marc G. Bellemare. Deep reinforcement learning at the edge of the statistical precipice. In *Advances in Neural Information Processing Systems (NeurIPS)*, volume 34, pp. 29304–29320, 2021. Outstanding Paper Award.

Marcin Andrychowicz, Anton Raichuk, Piotr Stańczyk, Manu Orsini, Sertan Girgin, Raphaël Marinier, Léonard Hussenot, Matthieu Geist, Olivier Pietquin, Marcin Michalski, Sylvain Gelly, and Olivier Bachem. What matters for on-policy deep actor-critic methods? A large-scale study. In *International Conference on Learning Representations (ICLR)*, 2021. arXiv:2006.05990 (as "What Matters In On-Policy Reinforcement Learning? A Large-Scale Empirical Study").

Xavier Bouthillier, Pierre Delaunay, Mirko Bronzi, Assya Trofimov, Brennan Nichyporuk, Justin Szeto, Nazanin Mohammadi Sepahvand, Edward Raff, Kanika Madan, Vikram Voleti, Samira Ebrahimi Kahou, Vincent Michalski, Tal Arbel, Chris Pal, Gaël Varoquaux, and Pascal Vincent. Accounting for variance in machine learning benchmarks. In *Proceedings of Machine Learning and Systems (MLSys)*, volume 3, pp. 747–769, 2021.

Sam Bowyer, Laurence Aitchison, and Desi R. Ivanova. Position: Don't use the CLT in LLM evals with fewer than a few hundred datapoints. In *Proceedings of the 42nd International Conference on Machine Learning (ICML)*, volume 267 of *PMLR*, pp. 81143–81184, 2025. Spotlight position paper.

Karl Cobbe, Vineet Kosaraju, Mohammad Bavarian, Mark Chen, Heewoo Jun, Lukasz Kaiser, Matthias Plappert, Jerry Tworek, Jacob Hilton, Reiichiro Nakano, Christopher Hesse, and John Schulman. Training verifiers to solve math word problems. *arXiv preprint arXiv:2110.14168*, 2021.

Cédric Colas, Olivier Sigaud, and Pierre-Yves Oudeyer. How many random seeds? Statistical power analysis in deep reinforcement learning experiments. *arXiv preprint arXiv:1806.08295*, 2018.

Jesse Dodge, Suchin Gururangan, Dallas Card, Roy Schwartz, and Noah A. Smith. Show your work: Improved reporting of experimental results. In *Proceedings of the 2019 Conference on Empirical Methods in Natural Language Processing and the 9th International Joint Conference on Natural Language Processing (EMNLP-IJCNLP)*, pp. 2185–2194, Hong Kong, China, 2019. Association for Computational Linguistics. doi: 10.18653/v1/D19-1224. arXiv:1909.03004.

Logan Engstrom, Andrew Ilyas, Shibani Santurkar, Dimitris Tsipras, Firdaus Janoos, Larry Rudolph, and Aleksander Madry. Implementation matters in deep policy gradients: A case study on PPO and TRPO. In *International Conference on Learning Representations (ICLR)*, 2020. arXiv:2005.12729.

Lang Feng, Zhenghai Xue, Tingcong Liu, and Bo An. Group-in-group policy optimization for LLM agent training. In *Advances in Neural Information Processing Systems (NeurIPS)*, volume 38, pp. 51797–51830, 2025. doi: 10.52202/085713-1544.

Aaron Grattafiori et al. The Llama 3 herd of models. *arXiv preprint arXiv:2407.21783*, 2024.

Horace He and Thinking Machines Lab. Defeating nondeterminism in LLM inference. Thinking Machines Lab: Connectionism, blog post, https://thinkingmachines.ai/blog/defeating-nondeterminism-in-llm-inference/, September 2025. Published 2025-09-10.

Peter Henderson, Riashat Islam, Philip Bachman, Joelle Pineau, Doina Precup, and David Meger. Deep reinforcement learning that matters. In *Proceedings of the AAAI Conference on Artificial Intelligence (AAAI)*, volume 32, pp. 3207–3214, 2018. doi: 10.1609/aaai.v32i1.11694.

Dan Hendrycks, Collin Burns, Saurav Kadavath, Akul Arora, Steven Basart, Eric Tang, Dawn Song, and Jacob Steinhardt. Measuring mathematical problem solving with the MATH dataset. In *Proceedings of the Neural Information Processing Systems Track on Datasets and Benchmarks*, volume 1, 2021.

Andreas Hochlehnert, Hardik Bhatnagar, Vishaal Udandarao, Samuel Albanie, Ameya Prabhu, and Matthias Bethge. A sober look at progress in language model reasoning: Pitfalls and paths to reproducibility. In *Conference on Language Modeling (COLM)*, 2025.

Jake M. Hofman, Angelos Chatzimparmpas, Amit Sharma, Duncan J. Watts, and Jessica Hullman. Pre-registration for predictive modeling. *arXiv preprint arXiv:2311.18807v1*, 2023.

International Council for Harmonisation of Technical Requirements for Pharmaceuticals for Human Use. ICH harmonised guideline: Addendum on estimands and sensitivity analysis in clinical trials to the guideline on statistical principles for clinical trials E9(R1). ICH Harmonised Guideline, 2019. Final version, adopted on 20 November 2019 (Step 4).

Scott M. Jordan, Yash Chandak, Daniel Cohen, Mengxue Zhang, and Philip S. Thomas. Evaluating the performance of reinforcement learning algorithms. In *Proceedings of the 37th International Conference on Machine Learning (ICML)*, volume 119 of *PMLR*, pp. 4962–4973, 2020.

Philippe Laban, Hiroaki Hayashi, Yingbo Zhou, and Jennifer Neville. LLMs get lost in multi-turn conversation. In *International Conference on Learning Representations (ICLR)*, 2026.

Zichen Liu, Changyu Chen, Wenjun Li, Penghui Qi, Tianyu Pang, Chao Du, Wee Sun Lee, and Min Lin. Understanding R1-Zero-like training: A critical perspective. In *Conference on Language Modeling (COLM)*, 2025.

Geoffrey R. Loftus. On interpretation of interactions. *Memory & Cognition*, 6(3):312–319, 1978. doi: 10.3758/BF03197461.

Mario Lucic, Karol Kurach, Marcin Michalski, Olivier Bousquet, and Sylvain Gelly. Are GANs created equal? A large-scale study. In *Advances in Neural Information Processing Systems (NeurIPS)*, volume 31, 2018. arXiv:1711.10337.

Charles F. Manski. *Partial Identification of Probability Distributions*. Springer Series in Statistics. Springer, New York, 2003. ISBN 0-387-00454-8. doi: 10.1007/b97478.

Gábor Melis, Chris Dyer, and Phil Blunsom. On the state of the art of evaluation in neural language models. In *International Conference on Learning Representations (ICLR)*, 2018.

Evan Miller. Adding error bars to evals: A statistical approach to language model evaluations. *arXiv preprint arXiv:2411.00640v1*, 2024.

Prabhat Nagarajan, Garrett Warnell, and Peter Stone. Deterministic implementations for reproducibility in deep reinforcement learning. In *2nd Reproducibility in Machine Learning Workshop at ICML 2018*, Stockholm, Sweden, 2018. Also presented at the AAAI 2019 Workshop on Reproducible AI (RAI), Honolulu; see arXiv:1809.05676v4.

Andrew Patterson, Samuel Neumann, Martha White, and Adam White. Empirical design in reinforcement learning. *Journal of Machine Learning Research*, 25(318):1–63, 2024.

Hung Viet Pham, Shangshu Qian, Jiannan Wang, Thibaud Lutellier, Jonathan Rosenthal, Lin Tan, Yaoliang Yu, and Nachiappan Nagappan. Problems and opportunities in training deep learning software systems: An analysis of variance. In *Proceedings of the 35th IEEE/ACM International Conference on Automated Software Engineering (ASE)*, pp. 771–783, 2020. doi: 10.1145/3324884.3416545. ACM SIGSOFT Distinguished Paper Award.

Penghui Qi, Zichen Liu, Xiangxin Zhou, Tianyu Pang, Chao Du, Wee Sun Lee, and Min Lin. Defeating the training-inference mismatch via FP16. *arXiv preprint arXiv:2510.26788*, 2025. v1 2025-10-30.

Qwen Team, An Yang, et al. Qwen2.5 technical report. *arXiv preprint arXiv:2412.15115*, 2024.

Rylan Schaeffer, Brando Miranda, and Sanmi Koyejo. Are emergent abilities of large language models a mirage? In *Advances in Neural Information Processing Systems (NeurIPS)*, volume 36, pp. 55565–55581, 2023. doi: 10.52202/075280-2425. Outstanding Paper Award.

Zhihong Shao, Peiyi Wang, Qihao Zhu, Runxin Xu, Junxiao Song, Xiao Bi, Haowei Zhang, Mingchuan Zhang, Y. K. Li, Y. Wu, and Daya Guo. DeepSeekMath: Pushing the limits of mathematical reasoning in open language models. *arXiv preprint arXiv:2402.03300*, 2024. v3; Table 5 and Section 4.2.

Noam Shazeer and Mitchell Stern. Adafactor: Adaptive learning rates with sublinear memory cost. In *Proceedings of the 35th International Conference on Machine Learning (ICML)*, volume 80 of *PMLR*, pp. 4596–4604, 2018.

Guangming Sheng, Chi Zhang, Zilingfeng Ye, Xibin Wu, Wang Zhang, Ru Zhang, Yanghua Peng, Haibin Lin, and Chuan Wu. HybridFlow: A flexible and efficient RLHF framework. In *Proceedings of the Twentieth European Conference on Computer Systems (EuroSys)*, pp. 1279–1297, 2025. doi: 10.1145/3689031.3696075.

Mohit Shridhar, Xingdi Yuan, Marc-Alexandre Côté, Yonatan Bisk, Adam Trischler, and Matthew Hausknecht. ALFWorld: Aligning text and embodied environments for interactive learning. In *International Conference on Learning Representations*, 2021.

Uri Simonsohn, Joseph P. Simmons, and Leif D. Nelson. Specification curve analysis. *Nature Human Behaviour*, 4(11):1208–1214, 2020. doi: 10.1038/s41562-020-0912-z.

Sara Steegen, Francis Tuerlinckx, Andrew Gelman, and Wolf Vanpaemel. Increasing transparency through a multiverse analysis. *Perspectives on Psychological Science*, 11(5):702–712, 2016. doi: 10.1177/1745691616658637.

Cecilia Summers and Michael J. Dinneen. Nondeterminism and instability in neural network optimization. In *Proceedings of the 38th International Conference on Machine Learning (ICML)*, volume 139 of *PMLR*, pp. 9913–9922. PMLR, 2021.

Wenjie Tang, Minne Li, Sijie Huang, Liquan Xiao, and Yuan Zhou. Rewarding beliefs, not actions: Consistency-guided credit assignment for long-horizon agents. *arXiv preprint arXiv:2605.20061*, 2026.

The vLLM and TorchTitan Teams. No more train-inference mismatch: Bitwise consistent on-policy reinforcement learning with vLLM and TorchTitan. vLLM blog, https://vllm.ai/blog/2025-11-10-bitwise-consistent-train-inference, 2025. 2025-11-10; accessed 2026-09-06.

L. von Werra, Y. Belkada, L. Tunstall, E. Beeching, T. Thrush, N. Lambert, S. Huang, K. Rasul, and Q. Gallouédec. TRL: Transformer reinforcement learning. https://github.com/huggingface/trl, 2020.

Eric-Jan Wagenmakers, Angelos-Miltiadis Krypotos, Amy H. Criss, and Geoff Iverson. On the interpretation of removable interactions: A survey of the field 33 years after Loftus. *Memory & Cognition*, 40(2):145–160, 2012. doi: 10.3758/s13421-011-0158-0.

Hanlin Wang, Chak Tou Leong, Jiashuo Wang, Jian Wang, and Wenjie Li. SPA-RL: Reinforcing LLM agents via stepwise progress attribution. *arXiv preprint arXiv:2505.20732*, 2025.

Feng Yao, Liyuan Liu, Dinghuai Zhang, Chengyu Dong, Jingbo Shang, and Jianfeng Gao. Your efficient RL framework secretly brings you off-policy RL training. Blog post, Feng Yao's Notion, https://fengyao.notion.site/off-policy-rl, 2025. First published 2025-08-05, last updated 2025-10-13; accessed 2026-09-06.

Jiayi Yuan, Hao Li, Xinheng Ding, Wenya Xie, Yu-Jhe Li, Wentian Zhao, Kun Wan, Jing Shi, Xia Hu, and Zirui Liu. Understanding and mitigating numerical sources of nondeterminism in LLM inference. In *Advances in Neural Information Processing Systems (NeurIPS)*, volume 38, pp. 188045–188077, 2025. doi: 10.52202/085713-5653. Oral presentation.

Chenchen Zhang. From reasoning to agentic: Credit assignment in reinforcement learning for large language models. *arXiv preprint arXiv:2604.09459v3*, 2026.

Donglin Zhuang, Xingyao Zhang, Shuaiwen Song, and Sara Hooker. Randomness in neural network training: Characterizing the impact of tooling. In *Proceedings of Machine Learning and Systems (MLSys)*, volume 4, pp. 316–336, 2022.

A THE INSTRUMENT AND THE TRAINING STACK

We trained one model, Qwen2.5-7B-Instruct, with one archived loop that holds its weights in bfloat16 without a master copy and runs at a constant learning rate, on a corridor whose anchor coverage halves with depth; every reading of the pool is conditional on these choices.

A.1 THE RELEASE

Everything a printed number rests on will be released with the paper. The supplementary material of this submission carries the code, with a README: KnobChain, every training-stack revision this paper names, the scoring scripts, the two-execution check of Appendix D.3, the analysis scripts and the two verification programs. The run records, the register with the documents its lines pin and the receipts the scripts printed, and the record supplement, together the full record this appendix points to, will be released with the paper. The run records are those of the 384 paired runs of the main pool and of every other experiment this paper reads, the runs that died included, indexed by cell and seed argument. The register is a single append-only text file whose lines are hash-stamped. The record supplement, a PDF, holds every registration and every registered prediction as written, the expectation ledger, the scoring transcripts and every passage cut from this paper; the transcripts, register entries and receipts behind every appendix section are in the full record.

A training run of either arm is one call of the archived loop revision whose digest the pool's run records carry, with the settings of Table 3, on one GPU; every run record begins with its own argument vector, so any run is relaunched the same way (§3 says why it does not replay bit for bit). Every printed number comes from a receipt, the record a script printed, naming the run records it read; with one environment variable naming the root of the full record the script reprints its receipt, and nothing needs a GPU to check a printed number. The two verification programs check the paper's numbers in the text layer of the compiled PDF (Appendix M). We do not release model weights or optimiser states.

A.2 THE CORRIDOR AND THE TWO CREDIT RULES

Section 2 describes KnobChain; here we give what makes it an instrument. The rule card is a difficulty tier of two-condition priority rules over the K candidate doors of a room card; an incorrect choice that does not end the episode sends the agent back one room, so each error costs two turns net, and the room index reached is logged per training batch and never shown to the agent. Each episode is rendered *deterministically* from a logical state (room index and surface identifier), so an oracle computes state values and per-decision advantage gaps in closed form, and the hash of the rendered card is an exact proxy of the logical state: the anchor collisions on which state-grouping estimators depend, different trajectories reaching the same card, have a measurable ground truth.

The knob λ sets how fast those collisions thin out with depth: at depth i the surface pool holds $M(i) = \mathrm{round}(M_0 g^i)$ surface identifiers, with $g = 1/\lambda$ and $M_0{=}4$, so the collisions thin out at rate λ^i. Cell families use $\lambda \in \{0.95, 0.98, 0.99, 1.00\}$, and $\lambda{=}1.00$ imposes no depth decay at all; every one of the 384 run headers of the main pool carries $\lambda{=}0.95$. Further knobs set a reward delay d (each step reward revealed d turns later, the grouping keys and the reward totals untouched), the reward condition (shaped or purely terminal), the grouping coarseness κ and the group size N (sixteen throughout the main pool), and each run's archived launch command pins its configuration verbatim.

The credit rules and the reward. Both arms score an episode by the z-score of its return within the group. The baseline arm broadcasts this episode-level advantage uniformly to every turn; the method arm regroups turns *across* trajectories by exact observation hash and, with a weight w of one, adds the z-score of the return-to-go inside the anchored turn group. The return-to-go is discounted at 0.95 per remaining turn, the loop's default in every revision, which no run of the pool or of the terminal-reward cells overrides. A correct advance pays 1, reaching the final room a further $0.5H$ and an incorrect choice nothing, so an error-free episode returns $1.5H$; the purely terminal condition, which pays 1 on success and nothing otherwise, is not used in the main pool.

A.3 THE TRAINING LOOP, ITS PRECISION AND ITS BUDGET

We read the loop's settings out of the archived loop and out of each run's recorded argument vector (Table 3); its optimiser is Adafactor (Shazeer & Stern, 2018). No launch command of the sixty-four runs at the shallowest depth sets the learning rate, temperature, gradient clip or generation cap.

Table 3: The pool's loop is one file with one set of settings for every run: sixteen episodes per update, a constant learning rate and bfloat16 weights without a master copy. The settings, as the archived loop and every run's launch command fix them.

setting	value
environment	KnobChain; rule card of difficulty tier R2; $K = 4$ doors; trap probability $q_{\mathrm{term}} = 0.25$; surface pool $M_0 = 4$ at $\lambda = 0.95$
model	Qwen2.5-7B-Instruct
group size	$N = 16$ episodes per update
reward	per-advance shaped reward with a terminal bonus, the delay knob at zero
advantage	the episode's z-score within the group; the method arm adds, with weight $w = 1$, the z-score of the return-to-go inside the anchored turn group, discounted at 0.95 per remaining turn
objective	token-level policy gradient; no ratio clipping, no KL term, no value network, no critic
optimiser	Adafactor with parameter scaling, relative step sizes and warm-up off; constant learning rate 2×10^{-6}; gradients clipped to a global norm of 1 and accumulated over the whole group before a single step
precision	weights and checkpoints in bfloat16; no single-precision master copy (next paragraph)
sampling	rollouts generated by the training model itself; temperature 0.7, `top_p` 1.0, `top_k` 0; at most eight new tokens per turn
tokens	context about one hundred per turn, generation two per turn (a door number); token budget $12{,}000$
budget	horizon-proportional: 5.4–$5.5 \times H$ updates, from 165 at H=30 to 540 at H=100
evaluation	the rollout path and sampler of training, not greedy; 256 episodes per point, each point from a seed of its own; spacing 15, 20, 25, 35, 40 and 50 updates from the shallowest cell to the deepest, the first point at step zero
turn cap	set per cell and recorded in every run's header

Precision. Weights and checkpoints are held in bfloat16, and the optimiser keeps no single-precision master copy: each Adafactor step upcasts a parameter, applies the update and rounds the result back to bfloat16, so an update below half a unit in the last place of the stored weight is lost. On a read-only sample of the base model's linear weights, an update the size of the learning rate changes a stored weight only where $|w| < 2^{-10}$, which holds for 6.9% of them; on a model of the update's distribution, an estimate and not a measurement of the runs, about 3.1–3.9% of all elements change at a step and 77–87% of the linear weights never change in 165 steps. The loop carries none of the noise sources of a served stack (batching in a generation server, train–inference mismatch, reduction order across devices, multi-epoch clipped updates) and carries this one, which they lack.

The second library's regime. All 338 runs we read on TRL, the second library, train under the standard regime described above, with a single-precision master copy. Its served default has nondeterminism shares of 1.347 and 0.490, both reproduced from its 48 runs; that library's failure to replay therefore comes from the standard regime, not from our regime of bfloat16 storage without a master copy.

Software. The public-stack experiments ran verl-agent at the commit Appendix D names, with the released recipes. The second library ran TRL 0.27.0 under its standard regime: parameters stored in single precision as the optimiser's master copy, bfloat16 autocast, fused AdamW at learning rate 10^{-6} with gradient clipping at 1.0 and a linear schedule, and a bfloat16 inference copy in the served generation server refreshed from the single-precision trainer.

Budget and evaluation grid. Budgets are proportional to the horizon in updates; environment turns grow faster, so the rule is not compute-matched (§8). The evaluation grid's spacing scales with depth, about twelve points per run, so the interval censoring of an event time differs by depth too (Appendix F). The learning rate is constant, no schedule constant depends on the step budget, and evaluation runs at the grid's spacing and at the final update, so a longer run is an intended extension of the otherwise identical shorter one, a property registered as a mechanical assertion (Appendix B.1).

Evaluation noise. Evaluation sampling, each point's 256 episodes drawn from a seed of its own on the training sampler, is a source of noise between executions: the binomial variance of a window mean is at most 0.40% of the within-seed mean square in the same-launch experiment's baseline

arm at $H{=}30$ and at most 8.89% in its method arm at $H{=}100$, the arm and depth with the smallest within-seed mean square; both are upper bounds that assume independent draws (Appendix C).

A.4 Single-precision master weights

With a single-precision master copy at a stable learning rate, the execution component of the baseline arm does not halve, so the rounding described above does not produce the execution-to-execution variance of this paper (§8); here we give the two registered blocks and the pilot behind it.

Setup. Both blocks repeat the one-block design of Appendix C.4: three executions of seed arguments 1 to 16 in both arms at $H{=}30$ under a single-precision master copy, compared with the bfloat16 block of that appendix at 2×10^{-6}. We registered two readings over the endpoint window. The execution component, the variance between executions of the same seed argument, is read as the ratio of the bfloat16 block's residual mean square to the master-copy block's, exact $F(30, 30)$, and the prediction asked whether the master copy halves it; the second reading is the seed's share. The first block ran at the loop's own 2×10^{-6}, the second at the rate a registered pilot chose.

Readings at 2×10^{-6}. Neither registered reading resolved. A single-precision master-weight block of the same design does not resolve whether the size of the execution component depends on the master copy: in the baseline arm the residual-variance ratio bfloat16 to single precision is 2.6029, interval $[1.2550, 5.3982]$, at $H{=}30$. Pooled over the two arms, each standardised by its own residual mean square, at the registered endpoint window: one launch block of three executions under single-precision master weights at $H{=}30$ does not resolve the seed's share; the interval is $[-0.1705, 0.2566]$. The baseline arm's residual variance is thus larger under bfloat16 at this rate, but the two blocks sit at different operating points and the ratio is not corrected for it: a bfloat16 block at a mean of 0.67 against a master-copy block at 0.03, whose median run scored 0 at each window step.

The method arm's ratio is a pre-specified secondary reading, because that arm's bfloat16 mean sits near the ceiling gate at 0.90 (Appendix M). The master-copy block did not trail from the start; it then declined, with growing gradient norms and loss bursts after stretches of near-zero loss and no non-finite value or out-of-memory retry, its median run at or below 1 of 256 from step 60 on and, at step 165, most of its runs at or below their untrained success rate, against none in the method arm and one run in the baseline arm under bfloat16. At roughly matched arm means, the bfloat16 block at step 45 (method) or 30 (baseline) against the master-copy block at step 165, the ratio reverses: 0.18 $[0.09, 0.37]$ in the method arm and 0.20 $[0.10, 0.41]$ in the baseline arm; neither this pairing, made after the data were seen, nor the trajectory reading was registered or scored.

The pilot and the stable block. A registered learning-rate pilot with a selection rule fixed in advance ran four runs at each of three rates and called a rate stable when all four runs passed three conditions: (a) success at step 165 above five times the step-0 value, (b) a mean gradient norm over steps 61–165 below twice the mean over steps 1–15, and (c) no evaluation after the run's maximum below half of it; a loss burst is a record above ten times the median of the fifteen before it.

At 2.5×10^{-7} the loop barely learns in 165 updates, at 10^{-6} one baseline run collapses to zero after step 135, and at 5×10^{-7} every run learns and its gradient norm falls in the last window (Table 4). At that rate the block reads ROBUST on the residual-variance prediction and BOUND on the seed's share (the interval's upper end bounds it; Table 5).

What this does not show. Neither block shows why the master-copy loop degrades at 2×10^{-6}. The probe that priced the cell expected faster learning under a master copy, which keeps the updates that bfloat16 rounds away; a larger effective step at the same nominal rate is one untested candidate.

A.5 What the seed pairing buys

The pairing buys the shared stream of task instances that the paired sign tests need, and nothing more: on this pool its gain in precision runs from a saving of 48% to a loss of 34%. Two executions of a bitwise-identical argument vector agree exactly at step zero and diverge from the first training step onward (Appendix B), so pairing on the seed argument does *not* pair the trajectories, and the paired difference carries the full execution noise of both arms.

Table 4: Of three learning rates, only 5×10^{-7} trains the single-precision master-copy loop stably in four of four runs. The pilot's reading, pre-specified and not scored: per rate, the runs passing each stability condition, the arm means at the registered endpoint window and the loss bursts (events, runs), beside the bfloat16 block and the single-precision re-run at 2×10^{-6}.

rate	stable (a, b, c of 4)	window mean, method / baseline	loss bursts, method	loss bursts, baseline
2.5×10^{-7}	no (3, 4, 1)	0.0827 / 0.0794	0 in 0	0 in 0
5×10^{-7}	yes (4, 4, 4)	0.9336 / 0.7188	7 in 2	1 in 1
10^{-6}	no (3, 4, 3)	0.9590 / 0.4694	15 in 2	3 in 1
bfloat16 block, 2×10^{-6}	—	0.8769 / 0.6691	—	—
single-precision re-run, 2×10^{-6}	—	0.1523 / 0.0318	—	—

Table 5: With a master copy at a stable rate, the baseline arm's execution component does not halve, and the seed's share is bounded but not resolved. The single-precision block at 5×10^{-7} against the bfloat16 block at 2×10^{-6}, three executions of seed arguments 1 to 16 per arm, at the registered endpoint window and at the final evaluation; ratio rows are bfloat16 over single precision.

reading	method	baseline	both arms
registered endpoint window, the single-precision block at 5×10^{-7} (scored)			
$\mathrm{ICC}_{\mathrm{within}}$, exact $F(15, 30)$	-0.1751 $[-0.3395, +0.1334]$	$+0.0525$ $[-0.1974, +0.4098]$	-0.0491 $[-0.2241, +0.1735]$; median-unbiased -0.0556; $F_{\mathrm{std}} = 0.8596$, the mean of 0.5529 and 1.1662
execution effect $F(2, 30)$, p	0.2865, 0.7529	0.3306, 0.7211	
residual-variance ratio bfloat16 / single precision at the window, exact $F(30, 30)$			
probability scale	0.5035 [0.2428, 1.0443] (a pre-specified secondary reading; near the ceiling)	0.7887 [0.3803, 1.6357] (scored)	
residual mean squares, bfloat16 / single precision (sampling floor as a share of each)	0.016065 / 0.031905 (0.8% / 0.5%)	0.051715 / 0.065571 (0.4% / 0.4%)	
arm mean, single precision / bfloat16	0.8002 / 0.8769	0.4504 / 0.6691	
final evaluation (a pre-specified secondary reading)			
$\mathrm{ICC}_{\mathrm{within}}$, single precision	-0.1181 $[-0.3066, +0.2118]$	$+0.0076$ $[-0.2278, +0.3623]$	-0.0515 $[-0.2258, +0.1706]$
ratio bfloat16 / single precision	0.5988 [0.2887, 1.2418]	0.5061 [0.2440, 1.0497]	
count			
runs; step 0	96 of 96 complete, 0 dead-run archives, 0 out-of-memory deaths; step-0 records identical to the bfloat16 twins in 96 runs, 0 differ		

With the same n seed arguments in both arms, the variance of the paired arm-mean difference is that of the unpaired design, in which each arm gets its own fresh draw of n seed arguments, times $1 - 2\,\mathrm{Cov}(Y_A, Y_B)/(\mathrm{Var}(Y_A) + \mathrm{Var}(Y_B))$; this needs no model. Read from each launch batch's per-seed arm contrasts, the six correlations run from -0.39 to $+0.78$, and four of the six have intervals containing zero; the seed-pairing correlation between arms of Appendix C.2 is the same quantity read per depth.

Under the two-component model the registered replication prediction assumes, the best case, in which the seed argument acts identically in both arms, the same ratio is the nondeterminism share of the two arms pooled, 0.86, 0.96 and 1.26 on the three comparisons: a saving of a seventh, a twenty-fifth and a loss of a quarter, the last a negative correlation between the arms that a clamp would hide. Averaging the two launch batches' standard deviations between seed arguments, the convention of §3, overstates the share by up to 11% by Jensen's inequality, in our favour.

What this does not show. Sixteen seed arguments do not measure the gain here and three cannot, and the two launch batches of each experiment are two executions of the same seed arguments, so the six readings are six instances and not six independent samples. Pair, because the sign tests need it, but do not budget for a gain you have not measured on your own pool.

A.6 ANCHOR COVERAGE AND AN INSTRUMENT DEFECT

Every run logs anchor coverage and group composition at every training step, computed for both arms under one anchor view so that they describe the policy distribution and not either estimator. Coverage is the fraction of turns whose observation hash is repeated within the group, counting any repeat across trajectories or within one (the two readings differ by at most 0.006); over the first three training records it is 0.825 to 0.856 at every depth in both arms, and over the registered endpoint window it halves from the shallowest cell to the deepest (Table 6), because deep updates carry far more turns (460.6 to 1554.6) while the anchor groups with at least two members per update rise from 127.7 to 226.9 and the turns carrying a step-level advantage from 141.5 to 212.9. The ladder was built so that anchors thin with depth; the coverage column is a pre-specified secondary reading, not a statement about the estimator's quality at any one depth.

Table 6: Anchor coverage halves from $H{=}30$ to $H{=}100$ in both arms, while the method arm's endpoint stays within 0.11 of one. Means by depth, thirty-two runs per arm, over the registered endpoint window $0.8T < \text{step} \leq T$; standard deviations in parentheses, the gap method minus baseline within the seed argument, and the number of seed arguments with a positive gap.

		coverage (window)		endpoint (sd)			
H	T	method	baseline	method	baseline	gap	sign
30	165	0.815	0.802	0.897 (0.140)	0.579 (0.279)	$+0.318$	28/32
40	220	0.734	0.729	0.934 (0.051)	0.642 (0.300)	$+0.292$	29/32
55	300	0.614	0.610	0.949 (0.032)	0.801 (0.135)	$+0.148$	30/32
70	380	0.516	0.515	0.924 (0.096)	0.823 (0.159)	$+0.101$	28/32
85	460	0.440	0.458	0.932 (0.070)	0.830 (0.225)	$+0.102$	26/32
100	540	0.380	0.403	0.924 (0.062)	0.813 (0.237)	$+0.111$	22/32

What we could not control, and what failed. A run replays bit for bit only with nondeterministic kernels switched off (Appendix B). A run whose parse-failure rate exceeds a threshold fixed in advance is excluded and reported. The pool was collected after a defect we found and disclosed: an earlier launcher configuration never passed the maximum-turns argument, so every run then on disk ran under a parser default's turn limit, binding in its deeper cells; we registered the disclosure before any repair and collected the pool on new code and a new stream of task instances.

What this appendix does not show. One group of sixteen episodes per update is a smaller batch than published recipes use. The seed's share at a larger batch is not measured here.

B THE REPLICATION MEASUREMENT

Re-executing a seed argument, the integer a run is launched with, does not reproduce the run on this loop; switching framework-level nondeterminism off does, within one machine and across two hosts. Section 3 reports the headline readings; here we give the designs and every reading behind them. An execution is a run of a given argument vector, a re-execution runs the bitwise-identical vector again, an experiment compares runs with their re-executions, and runs launched together under the same stack revision form a launch batch. The same-launch experiment, which keeps every execution in one launch batch and so removes the launch batch from the comparison, is Appendix C.

Setup. We read seven designs. Three cross-batch experiments are exploratory: a $5\times$-budget launch batch at $H{=}30$, and an early-stop and a $2\times$-budget launch batch at $H{=}100$, each a second execution of sixteen seed arguments per arm of the pool, run for other reasons and read on the shared evaluation grid. Four were registered, each with predictions scored by a released script: a third execution at $H{=}30$, the confirmatory test of the registered programme below; a deterministic-kernel fork at $H{=}30$ with a control, an extension, a full-length replay and a replay across two hosts; a four-execution experiment at $H{=}30$; and a re-execution on two further models.

B.1 THE CROSS-BATCH EXPERIMENTS

Setup. Three of our launch batches are, by construction, *second executions* of runs we had already executed. The $5\times$-budget launch batch at $H{=}30$ and the $2\times$-budget launch batch at $H{=}100$ share

Table 7: No cell's exact interval excludes an intraclass correlation of 0.24; pooled over the six, the four pooling rules put at most 0.17 to 0.30 of the endpoint variance on the seed argument. Each cell of Table 1 with its intraclass correlation as printed there, clamped at zero, and the exact 95% F interval of the one-way random-effects model; below the cells, the pooled reading at the registered endpoint window under four pooling rules.

experiment	arm	ICC	exact 95% interval
$H30$, 5×	method	0.000	$[-0.525, 0.422]$
$H30$, 5×	baseline	0.202	$[-0.298, 0.621]$
$H100$, e-stop	method	0.205	$[-0.296, 0.622]$
$H100$, e-stop	baseline	0.000	$[-0.656, 0.244]$
$H100$, 2×	method	0.000	$[-0.652, 0.249]$
$H100$, 2×	baseline	0.000	$[-0.655, 0.245]$
pooled, the equal-variance reference		0.0591	$[-0.1438, 0.2584]$
pooled, the seed-argument block bootstrap			$[-0.1633, 0.2674]$
pooled, the standardised mean of F		+0.0028	$[-0.2702, +0.1658]$
pooled, the Satterthwaite reference ($\nu_1 = 55.61$, $\nu_2 = 73.56$)			$[-0.1830, +0.3023]$

every argument with their 1× counterparts, including the integer that with the seed argument fixes the stream of task instances, and differ in the code revision, the step budget and the output directory. The constant learning rate and the evaluation trigger make the longer run an intended extension of the shorter one, so on the shared evaluation grid each pair is the same computation, run twice; we asserted record-level equality over that grid on the H=100, 2× experiment. Y is a run's success rate over the endpoint window, the evaluation steps near the end of the budget; σ_{rep}, σ_{tot}, the nondeterminism share and the intraclass correlation (ICC) are as defined with Table 1.

Readings. The two executions are not bitwise replays. All 32 step-0 records match exactly; then every training-derived field, evaluation success included, diverges from the first or second training step onward, and 320 of 352 shared-grid evaluation records and 17,077 of 17,280 training records differ. The argument vectors are identical for all 32 runs and the code-revision step is certified, so the divergence is kernel-level nondeterminism, not a configuration difference. Our registration had written strict extension into the assertion's premise, a scope error we disclose; under its registered consequence the comparisons are read as cross-run rather than nested, and no registered prediction is scored again. At inference the same divergence is the token flip that Yuan et al. (2025) trace to non-associative reductions, which a zero temperature does not remove (Hochlehnert et al., 2025; Laban et al., 2026).

Four of the six cells of Table 1 print an intraclass correlation clamped at zero, and a clamped zero is not evidence that there is no seed effect at this design: sixteen pairs return a negative estimate with probability 0.50 at a true intraclass correlation of zero, 0.35 at 0.1, 0.22 at 0.2 and 0.12 at 0.3, the exact probabilities, not simulated, that the one-way F statistic at 15 and 16 degrees of freedom falls below one. Tables 7 and 8 give the exact intervals and the readings that test them; the pooled value weights the six cells by their own mean squares, and the pooled block bootstrap resamples one vector of sixteen seed positions shared by all six cells.

Takeoff. The step at which a run first reaches the θ=0.70 takeoff threshold reproduces exactly in only 3–6 of the 16 pairs in the two uncensored experiments, with a median absolute discrepancy of one to two evaluation-grid steps. Nor does the seed argument carry over whether a run takes off at all: where takeoff by the budget is uncertain, the two executions agree for 7 of 16 seed arguments in the cross-batch baseline at H=30 (Cohen's $\kappa = -0.091$), and the six execution pairs of the same-launch baseline give κ from -0.500 to $+0.347$, with no exact test rejecting independence.

What survives re-execution. The estimand the paper reports, an arm-level difference, is the more reproducible of the two, while the individual run behind an individual seed, which published protocols implicitly treat as reproducible, is not (Table 9). The sign column of Table 1 is this table's agreement rate, a property of the experiment printed identically in both arms, and not the number of pairs behind the variance components, sixteen in every experiment; the reading at thirteen pairs changes no conclusion. On the endpoint window of §2 rather than the replication script's, the early-stop experiment's two contrasts read 0.300 and 0.339: its 540-step runs read the same in both windows,

Table 8: The F test, the bootstrap, a test of one common correlation and the scatter all agree that sixteen pairs per cell resolve no seed component. Readings of the six cross-batch cells beside Table 7; the bootstrap and exact ranges are those of the nondeterminism share in the first experiment ($H30$, $5\times$).

reading	value
F test of the null that no seed component is present	p between 0.21 and 0.85 in all six cells
nondeterminism share, percentile bootstrap over the seed pairs	method arm 0.96 to 1.32; baseline arm 0.58 to 1.21
the same on the exact interval (the bootstrap is narrower by a factor of 0.37 and 0.68, and is not the right interval for a ratio of variance components on sixteen pairs)	method arm 0.58 to 1.53; baseline arm 0.38 to 1.30
one common intraclass correlation across the twenty arm-level readings, sixteen on KnobChain and four on the public stack, likelihood-ratio test recomputed from the runs (the readings share runs)	not rejected, $p = 0.49$ over all twenty; common value 0.0588 $[-0.0388, 0.1623]$
median same-seed difference at the window	reaches 0.32 in the baseline arms; 0.03 to 0.07 in the method arms, which sit near the ceiling of the scale (perfect success)
largest same-seed difference anywhere on the grid	0.83 to 0.96 in success rate

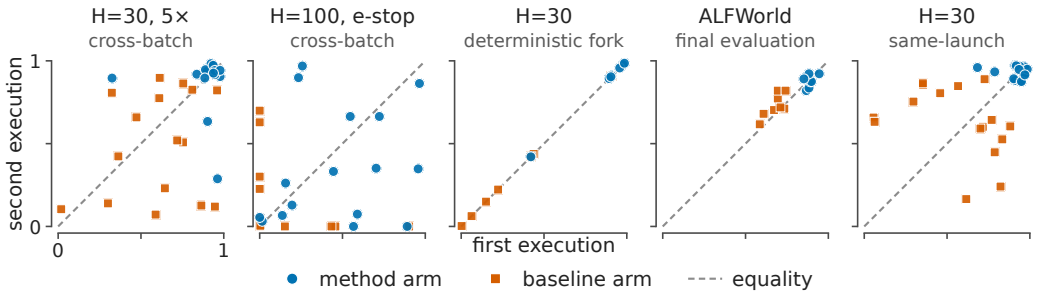


Figure 3: Only under deterministic kernels do two executions of one seed argument lie on the diagonal. Each point is a seed argument in one arm, its first execution's endpoint on the horizontal axis and its second's on the vertical; the dashed line is a bitwise replay. Left to right: one cross-batch experiment at each depth, the deterministic-kernel fork, the public-stack experiment at its final evaluation, and the same-launch experiment's scored pair of executions at the shallow depth.

Table 9: The arm contrast survives re-execution, and its sign agrees across the two executions no more often than independent executions would. Per cross-batch experiment: the mean per-seed contrast between the arms in each execution, on the replication script's own window; the agreement of its sign over the pairs on which it is non-zero; the probability that independent executions, each at its own sign rate, would agree at least this often; and Cohen's κ.

experiment	contrast, the two executions	sign agrees	independent executions agree	Cohen's κ
first ($H30$, $5\times$)	$0.381, 0.280$	$12/16$	0.7127	-0.103
second ($H100$, e-stop)	$0.237, 0.339$	$12/13$	0.7360	$+0.000$
third ($H100$, $2\times$)	$0.097, 0.139$	$11/16$	0.5113	$+0.091$

Table 10: One of the programme's five scored predictions missed, the intraclass correlation of the four-execution experiment, and four hit; of the two predictions of other registrations below the second rule, one hit and one is ambiguous by window. Per prediction: the registered gate, paraphrased below the second rule, the measured value and the outcome.

prediction	registered gate	measured	outcome
replay	$\sigma_{\mathrm{rep}}(\mathrm{det}) \le 0.25\,\sigma_{\mathrm{rep}}(\mathrm{nondet})$ in both arms *and* nd. share ≤ 0.25 in both arms	ratio 0.0000 (vs 0.0680, 0.2530); share $0.0000, 0.0000$	hit
sign	arm contrast keeps its sign *and* paired sign test $\ge 6/8$ for the method arm	sign kept; $8/8$, $p = 0.0039$	hit
branch	branch chosen by read-only reconnaissance before any run of the determinism cell	full operator-level determinism available; a two-step run executed twice was bitwise identical	branch (a)
ICC	$\mathrm{ICC} = \max(0, \frac{\mathrm{MSB}-\mathrm{MSW}}{\mathrm{MSB}+3\,\mathrm{MSW}}) \le 0.30$ in both arms	$0.0000, 0.3327$	miss
within/between	$\sigma_{\mathrm{within}} \ge 0.60\,\sigma_{\mathrm{between}}$ in both arms	ratio 2.3422, 1.1558	hit
two models	nd. share ≥ 0.50 in at least one arm of at least one family; a cell with mean outcome ≤ 0.05 in both arms is *not evaluable*	$0.9548, 1.1579$; $0.8025, 0.9257$	hit
variance ratio	in every arm-cell with $n \ge 16$ whose seed-paired mean gap against its cell baseline is negative, $\mathrm{Var}(\mathrm{baseline})/\mathrm{Var}(\mathrm{arm}) < 1$; *miss* if any such cell exceeds 1 with a bootstrap 95% interval excluding 1	one cell of nineteen triggers: gap -0.0173, ratio 1.1177, interval $[0.4137, 5.9253]$; no cell triggers under the terminal-record window	ambiguous by window
signature	on the zero-delay rung at $H{=}100$, both further arms satisfy the two halves of the reliability signature against the baseline arm; *miss* if either fails both	ratios 246.9231 and 965.5818; low-tail $+0.2422$ and $+0.1457$ against high-tail $+0.0174$ and $+0.0204$; six of the second arm's sixteen seed-argument runs absent	hit

one record at step 150, while its 165-step runs add their off-grid final record at step 165, which moves their gap from 0.237 to 0.300.

B.2 The registered programme

We treat the readings above as exploratory, since they come from launch batches run for other reasons, and we registered a confirmatory test before running it: a third execution of the $H{=}30$ computation, two arms by sixteen seed arguments, cloned argument for argument from the first. Its three predictions hit: the nondeterminism share is 1.2146 in the method arm and 0.8285 in the baseline arm, against a registered floor of 0.50; the ratio of the first-to-third spread to the first-to-second is 0.823 and 0.987, against a registered band of $[0.5, 2.0]$; and the arm-level contrast is 0.3046 against a registered floor of 0.10, with the sign of each seed argument's contrast agreeing in $13/16$ cases. The registration also fixed our wording, runs and not seeds, whatever the outcome, so that the vocabulary would not become a function of the result.

Table 10 gives the programme's five scored predictions on the other designs, on Y defined as the mean evaluation success over steps beyond 132; the branch row chose branch (a) of the deterministic-kernel test before any determinism run was dispatched and is not scored, and the sign row's registered $6/8$ is a sign gate, not a significance gate (one-sided binomial $p = 0.145$). The programme was scored at $H{=}30$, the shallowest of our six depths, and nothing in it licenses a statement about the deeper cells. Two predictions outside the programme are read at $H{=}100$: the full-length replay prediction of Appendix B.3 and the reliability-signature prediction at the foot of Table 10.

Table 11: Under deterministic kernels the same-seed spread is zero while the between-seed spread survives, and the arm contrast moves, carried by the baseline arm. The fork's eight seed arguments at $H{=}30$ beside the ordinary stack on the same eight and on the sixteen seed arguments common to both ordinary executions; placements are percentiles among all $10{,}518{,}300$ eight-argument subsets of the thirty-two nondeterministic seed arguments.

reading	method	baseline
deterministic, eight seed arguments: σ_{repl}(det); nondeterminism share	0.0000; 0.0000	0.0000; 0.0000
between-seed spread σ_{tot}, the mean of the two executions'	0.1844	0.2890
per-seed mean outcome, placement	0.8675, at the 26th percentile	0.3281, below the first
arm contrast, placement	0.5394, above the 99th percentile (two-sided exceedance 0.016), and above every one of the $12{,}870$ subsets of the sixteen seed arguments executed in both the pool and the $5\times$ launch batch	
ordinary stack, the same eight: σ_{rep}; nondeterminism share	0.0680; 1.0036	0.2530; 0.6387
arm contrast, placement	0.2848, at the 36th percentile	
ordinary stack, sixteen: $\sigma_{\text{repl}}/\sigma_{\text{tot}}$/ratio	$0.1403/0.1235/1.2920$	$0.2546/0.2789/0.8329$
per-seed contrasts of two ordinary executions (the sixteen seed arguments executed in both the pool and the $5\times$ launch batch), correlation	$+0.0449$: no more reproducible than across the two modes	

B.3 The deterministic-kernel switch

At $H{=}30$, the same-seed replication spread of the cross-batch experiments is removed by switching framework-level nondeterminism off: under deterministic kernels it collapses to zero.

Setup. The deterministic fork adds exactly 28 lines to the ordinary stack and removes none: one block, executed only when an environment variable is set, that switches the framework to deterministic algorithms, makes cuDNN deterministic and turns its benchmarking off, and checks the cuBLAS workspace variable. The sampler, the episode construction, the optimiser and every arithmetic path are untouched. A control separates the fork's code from its flags: run with the flag unset on seed arguments 1–8, the baseline arm's mean, 0.6017, sits inside the central 90% band $[0.4359, 0.7178]$ of the $\binom{32}{8}$ reference over the pool's $H{=}30$ cell (on the pool's own stack revision), a hit of its registered prediction; under a different cuBLAS workspace setting, a pre-specified secondary reading, it is 0.6873, also inside that band.

Readings. Under deterministic kernels the two executions of each of the eight seed arguments are identical, so the same-seed spread and the nondeterminism share are zero in both arms and the replay prediction holds, against ordinary-stack values recomputed on the same eight seed arguments (Table 11). The arm contrast keeps its sign: the paired sign test returns $8/8$ in favour of the method arm against a registered 6, with no ties and an exact one-sided binomial p of 0.0039. Its size moves, from 0.2848 on the ordinary stack over the same eight to 0.5394, above the 99th percentile of all eight-argument subsets and carried by the baseline arm (exploratory placements); with one effective execution per mode we cannot separate a systematic effect of the deterministic path from a fluctuation of this eight, and the prediction scores the sign, not the size. An exploratory extension to seed arguments 9–16 does not adjudicate the first eight.

At full length. We registered a full-length cell while both of its directories held zero runs: two arms by seed arguments 1–4 by two executions, sixteen runs on the deterministic fork at a budget of 540 steps with a turn cap of 250, each argument vector cloned from the ordinary $H{=}100$ cell with the output directory the one token rewritten. The prediction asks that the two executions of each of the eight pairs return the identical full evaluation vector, all twelve evaluation records on evaluation success, progress and return, at a gate of eight of eight: all 8 of 8 pairs are identical, a hit. Both executions of every pair ran on one machine, so the hit is replay within a machine at this budget; it licenses no single-execution design.

Across two hosts. We then re-ran the fork's own sixteen configurations at $H{=}30$ (seed arguments 1–8, both arms, 165 updates) once on each of two physical nodes, neither of them the machine on which the predictions above were scored, and registered the cell before any of its runs existed. A pair

Table 12: The four-execution experiment resolves a seed component in its baseline arm, where its intraclass correlation misses the registered gate; the within-to-between ratio clears its gate in both arms. One-way random-effects decomposition, eight seed arguments by four executions: the mean squares, the F test and exact interval (exploratory), and the two registered readings of Table 10.

arm	MSB; MSW	F test	ICC, exact 95%	σ_{within}	σ_{between}; threshold	ratio
method	$0.007951; 0.010905$	$F(7, 24) = 0.7291, p = 0.649$	$0.0000, [-0.229, 0.357]$	0.1044	$0.0446; 0.0268$	2.3422
baseline	$0.201513; 0.067297$	$F(7, 24) = 2.9944, p = 0.021$	$0.3327, [0.010, 0.753]$	0.2594	$0.2245; 0.1347$	1.1558

is identical when all twelve evaluation records agree on the three fields (evaluation success as its integer numerator over 256) and the endpoints Y over steps 135, 150 and 165 agree; the prediction reads REPLAYS-ACROSS-HOSTS at sixteen of sixteen identical pairs, DOES-NOT at eight or fewer and UNRESOLVED otherwise. All sixteen pairs are identical on all three comparisons, the scored one, every key but the time keys of every record of the fork's two-step smoke test, and each run against the fork's first execution on the original machine (the last two pre-specified secondary readings): REPLAYS-ACROSS-HOSTS.

The design has one depth, one budget, eight seed arguments, one execution per host and two hosts of one GPU generation with identical software: nothing unpinned (the model weights, the containers' python packages) is excluded, and a replay across two hosts licenses no single-execution design. Determinism removes the difference between two executions of one argument vector and not the spread across argument vectors, so it complements the replication prescription of §6.3 and does not replace it.

B.4 THE FOUR-EXECUTION EXPERIMENT, TWO FURTHER MODELS AND A TERMINAL REWARD

In the registered four-execution experiment the intraclass correlation misses in the baseline arm, at 0.3327 against a registered 0.30, and clears in the ceiling-bound method arm. The four executions (the original run, the confirmatory re-execution and two further re-executions of the same argument vectors) give the one-way decomposition of Table 12 over $k = 8$ seed arguments and $n = 4$ executions; the baseline arm exceeds the gate of ≤ 0.30 by 0.03, a miss. The first execution ran on an earlier stack revision than the other three, so a stack difference enters the within-seed term and 0.3327 is, if anything, biased low as an estimate of the seed's share in that arm. That experiment is the one design in this paper that resolves a seed component, in its baseline arm (exploratory F tests, not scored); the same-launch experiment's four executions, sixteen seed arguments per arm, do not resolve one ($[-0.1371, 0.3110]$ in the baseline arm, Appendix C.2).

A weaker prediction, stated without an intraclass-correlation estimator, clears in both arms: the within-seed standard deviation exceeds 0.60 times the between-seed one, so the reading that the seed argument explains little of the spread does not depend on the ICC estimator. The two are not the same statement: this prediction's 0.60 gate is ICC ≤ 0.717, more than twice the 0.30 the first was held to, so its clearing is not evidence that the first should have cleared.

Re-executing the same argument vectors on two further models reproduces the reading (Table 13): the registered gate asks for a share of at least 0.50 in at least one arm of at least one family, all four arms clear it, and neither family triggers the guard that returns *not evaluable* when both arms sit at or below a mean outcome of 0.05. The reading is not confined to a dense reward either: under a purely terminal reward at H=30, in a registered cell of two arms by sixteen seed arguments with a second execution, the cell's share prediction hit and its two other predictions missed (Appendix N); its band in the table is the share divided and multiplied by the 0.975 quantile of $F(15, 15)$.

B.5 WHAT THE DIFFERENCE BETWEEN TWO EXECUTIONS IS NOT

Seeding is complete on all three surfaces, the framework, the array library and the interpreter's own generator being seeded from the same argument in the same call, so the residual spread is not a seeding gap. Episode construction is a pure function of the instance-stream integer, the step index, the episode index and the seed argument, with no shuffling loader, so data order is not a nondeterministic surface either. Our rollouts use a local generation call at a fixed batch size with no dynamic batching, so a re-execution sees the first execution's batch composition, and the dominant source identified

Table 13: The nondeterminism share stays near or above one half on two further models and under a purely terminal reward. Per arm: the share, the mean outcome with the two executions' means beside it, and for the terminal-reward cell the share's variance-ratio band and the exact interval of $1 - \mathrm{ICC}$ from the same cell's one-way analysis of variance.

cell	arm	nondeterminism share	mean outcome (the two executions) or $1 - \mathrm{ICC}$
a second 7B-class model	method	0.9548	0.8723 (0.8773 and 0.8674)
	baseline	1.1579	0.3901 (0.3800 and 0.4001)
an 8B model from a different family	method	0.8025	0.9610 (0.9528 and 0.9692)
	baseline	0.9257	0.4637 (0.5467 and 0.3807)
terminal reward, 16 seed arguments by two executions	method	0.934 [0.326, 2.674]	$1 - \mathrm{ICC} = 0.921$ [0.463, 1.408]
	baseline	0.839 [0.293, 2.401]	$1 - \mathrm{ICC} = 0.826$ [0.397, 1.324]

Table 14: On the second library only colocated generation replays to the bit, and only for one model and task; every other configuration we ran does not replay without a switch. The second library's five configurations (Qwen2.5-1.5B-Instruct on GSM8K unless the row names another model or task): whether a re-execution replays, the nondeterminism share with its 95% bootstrap percentile band and the design, and what restores a replay.

stack, configuration	replays?	share [95% interval] (design)	what restores a replay
the second library, colocated generation, one process	yes, to the bit	0 (registered 8×2 per arm, 16 of 16 pairs)	—
the same, two trainer processes	no	— (two arguments, two hosts)	its full-determinism flag (3 of 3 pairs)
its released default, a generation server in a second process	no	calibration 1.31 [1.01, 1.57] / 0.69 [0.40, 1.00]; registered 1.347 [0.885, 1.943] / 0.490 [0.087, 1.624] (8×3; 1.082 / 0.691 on three executions)	the engine kept in the serving process: 35 of 35 replay, at 1.075 [1.044, 1.090] of the wall clock
a second model: Qwen2.5-0.5B-Instruct	no / not to the bit (colocated)	1.113 [0.83, 1.51] (12 pairs) / —	its full-determinism flag or eager attention (2 of 2 each)
a second task, colocated: MATH	no, not to the bit	— (2×2)	its full-determinism flag or eager attention (2 of 2 each)

for language-model inference, a reduction kernel whose result depends on how requests happen to be batched at serving time (He & Thinking Machines Lab, 2025), is excluded by construction. Since the step-0 evaluation, 256 episodes sampled on identical weights, is identical in every pair, sampling is deterministic given the weights: the difference enters through the first update's kernels, and the sampler then carries it. One hypothesis we state and do not test: the rollout sampler is an active consumer of the global generator, so a single bit of divergence inside one sampled action desynchronises the stream, after which the two executions explore different trajectories rather than the same trajectory with small numerical differences.

B.6 The second library

On the second library, TRL, generation colocated in the training process replays to the bit for the 1.5B model on GSM8K, and not for the 0.5B model or on MATH; the released default, a generation server in a second process, replays only when the generation engine is kept in the serving process, in 35 of 35 pairs, and what else restores a replay differs by configuration (Table 14; Table 2 merges the five configurations into two rows).

What this does not show. All of this is measured on one training stack, one model, one environment family and one hardware generation, apart from the cross-family re-execution, the stack revisions and the second library above; sixteen pairs per cell resolve no seed component, and the one design here that resolves one rests on eight seed arguments, with its first execution on an earlier stack revision. Determinism removes the difference between executions only where it was tested, at $H{=}30$ and at $H{=}100$ and 540 updates on one host, and licenses no single-execution design anywhere; the cross-batch experiments confound the execution with its launch batch and its code revision, which the same-launch experiment of Appendix C removes. What the scope does not weaken is the direction within the configurations of Table 2: where agents are trained as this literature trains them, sampling at nonzero temperature with nondeterministic kernels and one run per configuration, the seed is a label on a run, not a handle on it.

C THE SAME-LAUNCH EXPERIMENT AND THE ONE-BLOCK RE-RUN

Launching every execution of an experiment in one launch batch on one revision of the training stack does not resolve a positive share of the endpoint variance for the seed argument: no registered reading of the same-launch experiment or of its one-block re-run has an interval above zero, and pairing the two arms on a seed argument buys no stable precision. Section 3 and Figure 1b give the pooled readings; here we give the design and every reading behind them.

Setup. We re-executed seed arguments 1–16 of both arms, the method (GiGPO) against the baseline, paired on the seed argument and its stream of task instances, on KnobChain at the pool's configuration ($\lambda = 0.95$; Appendix A.2) with Qwen2.5-7B-Instruct: four executions at $H{=}30$ (165 updates) and two at $H{=}100$ (540 updates), in one launch batch on one certified stack revision, with a third execution at $H{=}100$ as an escalation cell. All sixteen seed arguments completed in both arms. Two follow-ups re-ran the $H{=}30$ half as one launch block of three executions, under bfloat16 (Appendix C.4) and under single-precision master weights (Appendix A.4).

Before any run existed we registered one scored prediction, R1, the pooled one-factor intraclass correlation of executions one and two over the four cells (two arms at two depths), $\mathrm{df} = (60, 64)$, at the registered endpoint window (steps 135, 150 and 165 at $H{=}30$; 450, 500 and 540 at $H{=}100$), with the final evaluation beside it; the other registered readings stand beside it and are not scored.

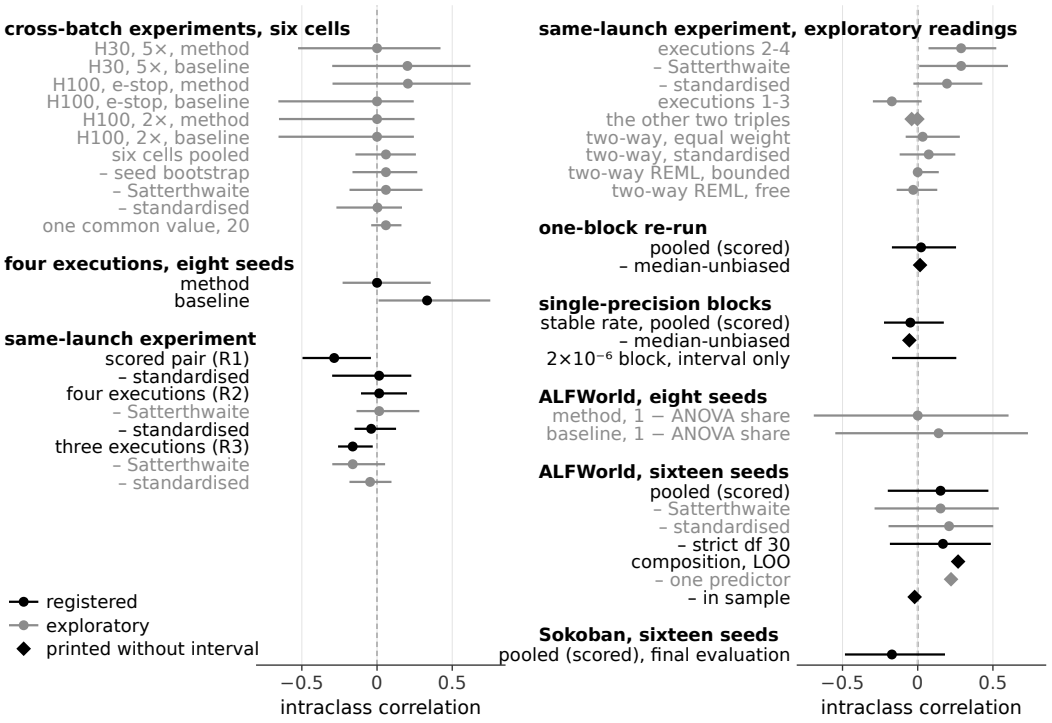


Figure 4: Every registered interval contains zero or lies below it; only exploratory intervals lie above zero: the four-execution experiment's baseline arm and executions two to four of the same-launch experiment. Every printed intraclass correlation of the seed argument's share, grouped by design, with its 95% interval; black: registered, grey: exploratory; the first master-copy block is drawn as its interval alone. The public stack's eight-seed rows draw the exact analysis-of-variance share as $1 -$ share, and negative estimates are clamped at zero.

C.1 THE REGISTERED RULE AND THE SCORED READING

The registration fixed, before any run existed, the rule that names the outcome of the scored reading: a seed component when the interval's lower end exceeds zero, a bound when its upper end is below 0.50, both when both hold and neither otherwise, which at these degrees of freedom is possible only

per cell and not for the pooled reading; when the homogeneity check fires, the pooled reading is withheld and the per-depth readings are printed. A bound licenses one sentence, fixed word for word: "at most U of the endpoint variance at ⟨window⟩ is the seed's; at least $1 - U$ is not".

The four per-cell intervals overlap (Table 15). The registered guard tests whether they overlap, which at sixteen pairs they nearly always do; it does not test the within-seed mean squares, which differ eighty-fourfold (Table 16). Pooled, at $\mathrm{df} = (60, 64)$, $F = 0.5565$ and the intraclass correlation is -0.2849 with the interval $[-0.4955, -0.0406]$ (exact only under homogeneity of the four cells' variance components, which they do not have), so the nondeterminism ratio is 1.2849 $[1.0406, 1.4955]$. That is the scored reading, and the rule calls it a bound; the sentence a bound licenses is undefined for a negative upper end, so this appendix prints the interval and reads no share from it.

C.2 The readings registered beside the scored one

Table 15 gives the scored pair per cell and pooled and the readings registered beside it: R2, the four executions at H=30 (balanced, $k = 4$), and R3, executions one to three at both depths, whose third execution at H=100, the escalation cell, completed for every seed argument before the registered reading time. R2 and R3 are registered and not scored, since the prediction was scored on R1. The standardised pooling divides each cell by its own within-seed mean square before pooling, and the Satterthwaite reading takes the degrees of freedom that the unequal mean squares imply. No registered arm, depth or pooling resolves a positive share; the rows marked exploratory were read after scoring.

Table 15: No registered reading of the same-launch experiment resolves a positive share of the endpoint variance for the seed argument; only exploratory poolings without the first execution exclude zero. The intraclass correlation with its 95% interval, per cell and pooled, at the registered endpoint window and at the final evaluation; sixteen seed arguments per arm.

reading	endpoint window	final evaluation
R1, the scored pair: executions one and two ($k = 2$)		
H=30, method	-0.0821 $[-0.5334, +0.4127]$	-0.1657 $[-0.5914, +0.3399]$
H=30, baseline	-0.3580 $[-0.7100, +0.1456]$	-0.1794 $[-0.6005, +0.3274]$
H=100, method	$+0.3522$ $[-0.1437, +0.7110]$	-0.0115 $[-0.4808, +0.4697]$
H=100, baseline	-0.1724 $[-0.5959, +0.3338]$	$+0.0265$ $[-0.4511, +0.4988]$
pooled, $\mathrm{df} = (60, 64)$ (scored)	-0.2849 $[-0.4955, -0.0406]$	-0.0973 $[-0.3345, +0.1536]$
pooled, standardised	$+0.0141$ $[-0.2984, +0.2279]$	-0.0748 $[-0.3772, +0.1420]$
R2: the four executions at H=30 ($k = 4$)		
method	-0.1105 $[-0.2181, +0.1186]$	$+0.1142$ $[-0.0768, +0.4180]$
baseline	$+0.0240$ $[-0.1371, +0.3110]$	$+0.0449$ $[-0.1236, +0.3372]$
pooled, $\mathrm{df} = (30, 96)$	$+0.0143$ $[-0.1066, +0.1991]$	$+0.0484$ $[-0.0813, +0.2409]$
pooled, standardised	-0.0389 $[-0.1495, +0.1265]$	$+0.0809$ $[-0.0630, +0.2742]$
pooled, Satterthwaite	$[-0.1365, +0.2811]$	$[-0.1145, +0.3212]$
R3: executions one to three at both depths ($k = 3$)		
pooled, $(60, 128)$	-0.1618 $[-0.2591, -0.0285]$	-0.0873 $[-0.2005, +0.0613]$
per cell, method then baseline	-0.1977, -0.1703 at H=30; $+0.2092$, -0.1497 at H=100	
pooled, standardised	-0.0461 $[-0.1833, +0.0960]$	$+0.0240$ $[-0.1269, +0.1738]$
pooled, Satterthwaite	$[-0.2976, +0.0536]$	$[-0.2366, +0.1244]$
exploratory, read after scoring		
executions two to four at H=30, pooled	$+0.2886$ $[+0.0718, +0.5224]$	$+0.2611$ $[+0.0461, +0.4982]$
standardised	$+0.1946$ $[-0.0289, +0.4296]$	$+0.2917$ $[+0.0586, +0.5180]$
Satterthwaite	$[+0.0097, +0.5998]$	$[-0.0195, +0.5781]$
executions one to three at H=30, pooled	-0.1720 $[-0.2981, +0.0263]$	-0.1216 $[-0.2632, +0.0918]$
the other two triples at H=30	-0.0403 and -0.0021	$+0.0077$ and $+0.0446$
within-block share, four cells, equal weight;	$+0.0329$ $[-0.08, +0.28]$; $+0.0732$	
standardised	$[-0.12, +0.25]$	
one common correlation (REML), bounded; free	0 $[0, 0.14]$; -0.03 $[-0.14, +0.13]$	

Pairing. The pairing reading is the correlation between the arms' outcomes across seed arguments. At H=100 we read it on the sixteen seed arguments complete in both arms and both executions, with execution 1 as the registered reading, against the pool at the same depth, where Pearson's correlation is $+0.3950$ with permutation $p = 0.0260$ on $B = 20{,}000$ relabellings. The registered entry asked for a positive correlation with $p < 0.05$, and the answer is no: one execution gives $+0.1321$ $[-0.3891, +0.5892]$, an interval that includes the pool's value, and the other -0.2548 $[-0.6664, +0.2757]$, an interval that excludes it. Whatever the pairing buys is therefore not a stable

property of the seed argument, and no pre-specified secondary reading, Pearson's or Spearman's on each execution, on their average or on the two stacked, finds the positive correlation either.

Launch timing. At $H{=}30$ every run of the first execution was launched before any run of the other three, the first of them about ten hours before the other executions began. We read the launch times from the first line of each run's log.

C.3 Readings added after scoring

We added six readings after the cell's prediction was scored and its data unblinded, under a reading rule written after their values were known; together they carry about sixty intervals and forty correlations with no correction for multiplicity. None of these readings enters a registered prediction, a branch, an outcome, an expectation or a tally. Leaving one execution out, the triple that excludes the first reads the highest and executions one to three the lowest (Table 15); with the choice of execution counted, permuting the four execution labels within each seed argument gives $p = 0.11$ at the registered endpoint window and 0.27 at the final evaluation, and the split rests on seed argument 1.

The $H{=}30$ baseline cell carries most of the pooled sums of squares, and the evaluation's binomial noise is at most 8.89% of the within-seed mean square in any cell (Table 16), so what is left is the training run's own nondeterminism. Modelling each execution as an additive block moves the seed's share by less than 0.01 in every $H{=}30$ cell, so the difference between the four-execution readings and those without the first execution is a seed-by-execution interaction and not a block shift.

Table 16: The $H{=}30$ baseline cell carries most of the pooled sums of squares, and the evaluation's own sampling noise is at most 8.89% of the within-seed mean square in any cell. Per cell: the share of the pooled within-seed sum of squares and the within-seed mean square; the binomial variance of a mean over records of 256 episodes, an upper bound that assumes independent draws; the execution effect of the two-way model.

cell	$H{=}30$ method	$H{=}30$ baseline	$H{=}100$ method	$H{=}100$ baseline
share of the pooled within-seed sum of squares, window	0.0353	0.7128	0.0085	0.2434
within-seed mean square, window	0.003869	0.078100	0.000930	0.026672
the same share, final evaluation	0.0331	0.6066	0.0388	0.3215
binomial variance as a share, at most: window	2.32%	0.40%	8.89%	0.64%
the same, final evaluation	6.16%	1.43%	5.96%	1.49%
execution effect F (p), on $(3, 45)$ and $(1, 15)$ df	0.40 (0.75)	0.72 (0.54)	4.20 (0.06)	0.12 (0.74)

C.4 One launch block

We re-executed the experiment's $H{=}30$ half as one launch block: three executions of every seed argument and arm, interleaved in a random launch order, registered before any of its runs existed. The seed's share is read as $\mathrm{ICC}_{\mathrm{within}}$, its share once each execution is modelled as an additive random effect. Pooled over the two arms, each standardised by its own residual mean square, at the registered endpoint window: one launch block of three executions at $H{=}30$ does not resolve the seed's share; the interval is $[-0.1715, 0.2549]$. The registered rule calls the share resolved when the interval's lower end exceeds zero and bounded when its upper end is below 0.25; this interval is neither, and at the parent experiment's threshold of 0.50 it would have read as bounded.

Within the block, in readings made after it was scored, the baseline arm's three pairwise readings are more uneven than execution labels permuted within seeds produce ($p = 0.0442$), and eight same-host pairs against forty cross-host pairs read $+0.5859$ against -0.0278 ($p = 0.1122$), a difference that sixteen seeds cannot resolve. The same one-block design under single-precision master weights is Appendix A.4.

What this does not show. The experiment ran on one certified stack revision and the pool on another, and the registered runs on the pool's revision that would have separated the two were never launched, so every comparison of this experiment with the pool's runs confounds the launch batch with the stack revision. No share is read from the scored interval; the readings registered beside it resolve none (Tables 15 and 17), and their upper ends bound the share near a quarter. Pairing buys no stable precision.

Table 17: Over three executions the block resolves no share; executions one and two alone would have resolved one in the baseline arm. Per arm and pooled over the arms (each standardised by its own residual mean square), at the registered endpoint window and at the final evaluation, with the parent experiment's readings beside.

reading	method	baseline	both arms
registered endpoint window (scored)			
$\mathrm{ICC}_{\mathrm{within}}$, exact $F(15, 30)$	-0.1401 $[-0.3195, +0.1823]$	$+0.1442$ $[-0.1310, +0.4984]$	$+0.0223$ $[-0.1715, +0.2549]$; median-unbiased $+0.0155$; $F_{\mathrm{std}} = 1.0684$, the mean of 0.6313 and 1.5056
pairs of executions at the window, one-way, $k = 2$ (exploratory)			
executions 1 and 2	-0.0085 $[-0.4786, +0.4720]$	$+0.6461$ $[+0.2506, +0.8591]$	$+0.5350$ $[+0.2359, +0.7434]$
executions 1 and 3	-0.0963 $[-0.5436, +0.4008]$	-0.1571 $[-0.5857, +0.3476]$	-0.1466 $[-0.4655, +0.2101]$
executions 2 and 3	-0.2285 $[-0.6323, +0.2808]$	-0.0699 $[-0.5246, +0.4229]$	-0.1034 $[-0.4305, +0.2515]$
final evaluation (pre-specified secondary reading)			
$\mathrm{ICC}_{\mathrm{within}}$, exact $F(15, 30)$	-0.2203 $[-0.3645, +0.0659]$	$+0.2489$ $[-0.0473, +0.5875]$	$+0.0702$ $[-0.1348, +0.3065]$; median-unbiased $+0.0631$; $F_{\mathrm{std}} = 1.2263$, the mean of 0.4583 and 1.9943
the parent experiment's $\mathrm{ICC}_{\mathrm{within}}$ beside this cell's, window then final evaluation			
window, 1–4, $k = 4$	-0.1175 $[-0.2230, +0.1084]$	$+0.0194$ $[-0.1415, +0.3065]$	
window, 2–4, $k = 3$	$+0.0313$ $[-0.2119, +0.3878]$	$+0.3106$ $[+0.0064, +0.6349]$	
window, 1–3, $k = 3$	-0.2071 $[-0.3572, +0.0862]$	-0.1814 $[-0.3430, +0.1244]$	
final, 1–4, $k = 4$	$+0.0971$ $[-0.0905, +0.4002]$	$+0.0417$ $[-0.1273, +0.3346]$	
final, 2–4, $k = 3$	$+0.3019$ $[-0.0015, +0.6284]$	$+0.2613$ $[-0.0369, +0.5973]$	
final, 1–3, $k = 3$	-0.0693 $[-0.2770, +0.2736]$	-0.1351 $[-0.3166, +0.1891]$	

D THE PUBLIC-STACK EXPERIMENTS

We find that on the public stack that published work uses, the seed argument's share of the final evaluation's variance is bounded and not resolved, on ALFWorld and on Sokoban, and that on ALFWorld the endpoint and takeoff-hazard readings of the method effect separate at the registered success threshold and not below it. Section 5.3 reports the headline readings.

Setup. We ran every experiment, the set of runs made for one registered question, on verl-agent at commit `20bd331bdbc9` with the released recipes' loss settings, which neither recipe overrides: the loss is averaged over all response tokens of a micro-batch (token-mean aggregation), the per-batch length normalisation that Liu et al. (2025, Table 2 and Listing 1) find in verl (Sheng et al., 2025), and group advantages are divided by the group's standard deviation in the GRPO arm and, on ALFWorld but not on Sokoban, in the GiGPO arm. On ALFWorld the reward is ten times the episode's success, paid when the task is won.

D.1 THE EIGHT-SEED EXPERIMENT, READ BY TASK TYPE

At eight seed arguments both readings of the method effect grow with the number of sub-goals, and the experiment is not precise enough to tell them apart.

Setup. We read the comparison per task type, with the number of sub-goals a task type requires as the proxy for its length, and registered that reading before any scored run completed; it rests on a proxy and licenses no sentence about why the two readings move differently. A reading's width is the number of evaluation points it averages: the final evaluation alone, or the registered six-point window that ends at it.

Readings: the horizon. The endpoint gap grows from 0.1857 at two sub-goals to 0.3393 at four. Its shortest-to-longest ratio is 0.55 $[0.44, 0.69]$, and the takeoff hazard ratio's is 0.57 $[0.21, 1.46]$: the point estimates almost coincide and only the precision differs. Under the registered per-type gates, which pass three of the six types, the two ratios are 0.63 $[0.47, 0.86]$ and 0.80 $[0.22, 2.55]$ on 2,000 replicates. On the relative failure reduction, the gap over the baseline's failure rate, the four-sub-goal type's reduction is the larger on both ALFWorld experiments at both widths, so the growth survives the change of scale.

Readings: replication. The experiment's scored replication reading, each arm's nondeterminism ratio, is not robust to the width. At the registered six-point window the registered estimator, $\sigma^2_{\text{rep}}/\sigma^2_{\text{tot}}$ with its seed-resampling percentile band, reads 1.179 $[0.726, 1.803]$ in the method arm and 0.744 $[0.236, 1.753]$ in the baseline arm, and the ANOVA share $1 - \text{ICC}$ with its exact one-way random-effects interval on $F(7, 8)$ reads 1.095 $[0.396, 1.691]$ and 0.861 $[0.267, 1.548]$; the registration quotes the band, which on the pool is 0.37 to 0.68 times as wide as the exact interval we print for every share. At the final evaluation alone, the width most papers use, the ratios are 0.5788 and 0.2761, so the baseline arm falls below the registered floor of 0.50. The scored prediction stands as registered.

What this does not show. At this precision the experiment cannot separate the endpoint and hazard readings, and its reading on the relative failure reduction is exploratory.

D.2 The sixteen-seed ALFWorld re-run

At sixteen seed arguments the seed argument's share of the final evaluation's variance is bounded, not measured, and the endpoint and takeoff-hazard readings separate at the registered success threshold, on the strength of the longest task type.

Setup. We re-ran the experiment at the benchmark's own depth with Qwen2.5-1.5B-Instruct, GiGPO (method) against GRPO (baseline) paired on the seed argument, on sixteen new seed arguments (128 to 368 spaced 16) with two executions each: sixty-four runs of 150 updates, evaluated every five steps on 128 games, per task type and pooled, and read at the final evaluation (step 150) and at the six-point window (steps 125 to 150).

Before any run existed we fixed two registered predictions on the final evaluation. The seed prediction is a hit when the exact F interval on $(31, 32)$ degrees of freedom of the arms-pooled one-factor intraclass correlation excludes zero. The estimand prediction reads the ungated $D = \ln R_{\text{end}} - \ln R_{\text{haz}}$, the log-ratio of the endpoint gap's and the takeoff hazard ratio's shortest-to-longest ratios, on a seed-argument bootstrap of $B = 2000$, as separating when its interval excludes zero and agreeing when it lies inside ± 0.88. Both were scored once, with all sixty-four runs complete; the secondary readings below were registered beside them and not scored.

Readings: the seed. The pooled interval contains zero, so the seed prediction is a miss, and the share the seed argument controls is at most 0.470; the seed argument draws the evaluation set, and on this stack it moves only the environment instances and the random streams. The design had power 0.89 against a true intraclass correlation of 0.48, and 0.53 against 0.30. These registered figures are one-sided; for the two-sided test that the interval scores they are 0.8244 and 0.4001. The nondeterminism share, a secondary reading, clears its floor of 0.50 in both arms at both widths.

No pooling resolves the seed's share of the final-evaluation variance. Per arm and pooled, each arm centred on its own mean, with exact F intervals (per arm on $(15, 16)$ degrees of freedom); the standardised twin divides each arm by its own residual mean square before pooling, and the Satterthwaite twin takes the degrees of freedom that the unequal mean squares imply.

reading	arm	MSB	MSW	F	ICC	95% interval	p	σ_{rep}	σ_{tot}	nd. share
final	gigpo	0.001234	0.000730	1.6897	0.2564	$[-0.245, 0.655]$	0.1543	0.0279	0.0302	0.8553
final	grpo	0.007180	0.005266	1.3635	0.1538	$[-0.343, 0.589]$	0.2724	0.0634	0.0739	0.7368
final	pooled, df $(31, 32)$	0.004071	0.002998	1.3580	0.1518	$[-0.199, 0.470]$	0.1969			
final	pooled, strict $\text{df}_1 = 30$				0.1678	$[-0.185, 0.486]$	0.1740			
window	gigpo				0.1658	$[-0.332, 0.597]$				0.9668
window	grpo				0.1866	$[-0.313, 0.611]$				0.8223
window	pooled, df $(31, 32)$			1.3985	0.1662	$[-0.185, 0.481]$	0.1751			
window	pooled, strict $\text{df}_1 = 30$				0.1821	$[-0.171, 0.497]$				
final	standardised twin				$+0.2084$	$[-0.194, +0.502]$				
final	Satterthwaite twin ($\nu_1 = 20.01$, $\nu_2 = 20.35$)					$[-0.287, +0.539]$				

In the method arm the within-seed mean square sits at the binomial floor of a 128-game evaluation, $p(1-p)/128$ at the arm's step-150 mean p; in the baseline arm it is above it.

arm	step-150 mean p	MSW	$p(1-p)/128$
gigpo	0.9009	0.000730	0.000697
grpo	0.6909	0.005266	0.001668

The evaluation-sampling split. Re-evaluating saved checkpoints does not resolve how much of the within-seed variance is evaluation sampling, in either arm. In a cell registered before any of its runs existed, we retrained four of the re-run's seed arguments (128, 144, 160 and 176) once per arm and evaluated each final checkpoint three times on the 128 games of its own step-150 validation, the generation seed offset by 10,000 per repeat so that sampling at temperature 0.4 is the only difference. The share S, the mean variance across the three evaluations over the within-seed mean square, reads EVAL-DOMINATED or TRAINING-DOMINATED when its exact interval lies above or below 0.5, and UNRESOLVED otherwise. Both arms read UNRESOLVED, 0.26 $[0.08, 1.08]$ in the method arm and 0.20 $[0.07, 0.83]$ in the baseline arm; four checkpoints per arm resolve a share only below 0.12 or above 1.56, and the readings made after scoring, marked in the table, do not change the outcome.

arm	checkpoints	V_{eval} (df)	MSW (df)	S	exact 95% interval	branch
gigpo (method)	4	0.000193 (8)	0.000730 (16)	0.2641	$[0.0845, 1.0766]$	UNRESOLVED
grpo (baseline)	4	0.001071 (8)	0.005266 (16)	0.2035	$[0.0651, 0.8293]$	UNRESOLVED
not scored: the binomial reference $\bar{p}(1-\bar{p})/128$ at the arm's mean $\bar{p}$, and V_{eval} over it						
gigpo		0.000692	$\bar{p} = 0.9017$	0.2787		
grpo		0.001404	$\bar{p} = 0.7650$	0.7628		
not scored: the run's own step-150 validation as a fourth evaluation, V_{eval4} (df) and $V_{\text{eval4}}/\text{MSW}$; the intervals and the pooled row read after scoring						
gigpo		0.000144 (12)		0.1974	$[0.07, 0.62]$	
grpo		0.001145 (9)		0.2174	$[0.07, 0.81]$	
pooled				0.19	$[0.09, 0.44]$	
read after scoring: S pooled over the arms on $(16, 32)$ degrees of freedom, and each arm standardised						
pooled				0.21	$[0.09, 0.54]$	
standardised				0.23	$[0.10, 0.60]$	

Composition of the evaluation set. The registered adjustment of each seed's mean for the task-type composition of its 128 games, a leave-one-seed-out regression on the five free composition counts, does not lower the pooled correlation: 0.1518 raw, 0.2685 after the registered adjustment and 0.2215 under a one-predictor diagnostic; only the in-sample fit, -0.0204, reads lower. Nothing is subtracted from any scored quantity, and the seed prediction is read on the raw field.

Readings: the estimand. The interval on $D = \ln R_{\text{end}} - \ln R_{\text{haz}}$ excludes zero, so by the registered rule the endpoint and hazard readings separate; its upper end exceeds 0.88, so the flag that would read the ALFWorld separation as smaller than that of the pool's horizon ladder does not hold. Takeoff is the first evaluation step with success at or above $\theta = 0.70$ on the grid 0 to 150 by 5, right-censored at 150.

The longest task type carries the largest endpoint gap and by far the largest takeoff hazard ratio. Per task type at the final evaluation: the sub-goal proxy, the paired gap (method minus baseline), the observed-over-expected hazard ratio with its event counts (O: runs that took off, of 32 per arm) and the registered per-type ceiling gate's decision (gated out at the ceiling of 0.90, with the baseline and method means).

type	proxy	paired gap	HR O/E	O_m	E_m	O_b/E_b	gate
pick_and_place	2	0.1757	2.960	32.0	16.2	32.0/47.8	gated out, 0.769 / 0.945
look_at_obj_in_light	2	0.1316	4.587	32.0	12.3	26.0/45.7	pass
pick_clean_then_place_in_recep	3	0.1973	4.535	32.0	11.7	31.0/51.3	gated out, 0.724 / 0.921
pick_heat_then_place_in_recep	3	0.1988	3.887	32.0	13.1	32.0/50.9	gated out, 0.757 / 0.956
pick_cool_then_place_in_recep	3	0.1900	4.603	32.0	12.2	26.0/45.8	pass
pick_two_obj_and_place	4	0.3399	17.686	31.0	10.7	4.0/24.3	pass

Robustness. Every variant of D but the registered one is exploratory, read after the data were in on the same seed resamples. The long type's hazard ratio rises from 4.6 to 53.6 across $\theta = 0.50$ to 0.80 because its baseline arm mostly sits below the threshold: its mean is 0.519 at step 150, and 25, 13, 8, 4 and 1 of its 32 runs ever reach 0.50, 0.60, 0.65, 0.70 and 0.80. R_{haz} excludes one at every threshold, so both readings grow with the sub-goal count everywhere and the hazard grows more than the endpoint at $\theta \geq 0.70$; more baseline takeoffs, which a lower threshold produces, remove the separation. The long type's ratio 17.686 rests on 31 of 32 method and 4 of 32 baseline takeoffs, two of the four at step 150; its seed bootstrap interval is $[10.19, 61.68]$.

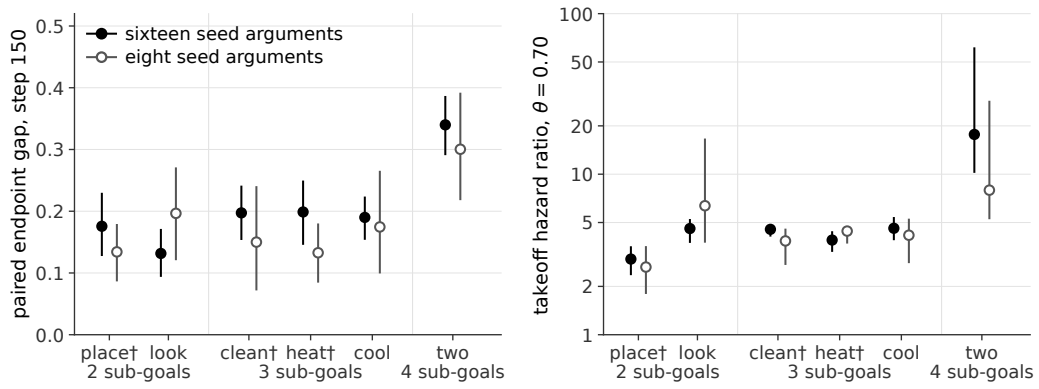


Figure 5: On ALFWorld both readings are larger on the four-sub-goal type than on the two-sub-goal types, and in the sixteen-seed re-run the hazard ratio rises more than the endpoint gap. Per task type, the paired endpoint gap at step 150 (left) and the takeoff hazard ratio at $\theta = 0.70$ (right, log axis), with 95% intervals from 2,000 seed-argument resamples; filled: sixteen seed arguments (32 runs per arm), hollow: eight (16), †: gated out at the ceiling.

Table 18: Both shortest-to-longest ratios lie below one, the hazard ratio's further, and their log-difference D excludes zero at the registered threshold and window and under leave-one-out, but not with the ceiling gate at the final evaluation, at thresholds of 0.65 and below or under a two-crossing takeoff. The sixteen-seed re-run's two ratios and D under each variant, with 95% intervals from the registered seed-argument bootstrap ($B = 2000$).

quantity	value
short-group gap against the long type's	0.1536 against 0.3399
R_{end}	0.4520, [0.3401, 0.5906] (median 0.4513, SE 0.0637)
short-group hazard ratio against the long type's	3.774 against 17.686
R_{haz}	0.2134, [0.0629, 0.3610] (1986 of 2000 draws defined)
Spearman of proxy with gap; with ln HR	0.9258; 0.6172
D as registered: ungated, $\theta = 0.70$, final (scored)	0.7507, [0.2259, 1.9358] (median 0.7489, SE 0.4042)
window	0.8104, [0.3279, 2.0252]
gate applied (no pick_and_place), final	0.4008, [−0.1773, 1.5999]
gate applied, window	0.7699, [0.2545, 1.9797]
$\theta = 0.50$	−0.1648, [−0.4573, 0.0614]
$\theta = 0.60$	0.0394, [−0.3716, 0.4760]
$\theta = 0.65$	0.3724, [−0.1046, 1.1043]
$\theta = 0.75$; 0.80	below the registered defined-fraction floor (1771 and 1272 of 2000 replicates defined)
$\theta = 0.90$	undefined: no baseline takeoff on the long type
two consecutive crossings	0.515, [−0.071, 1.746]
leave-one-out of the long type's four baseline takeoffs	0.9723 to 0.9724, every interval excluding zero; as a censoring, 1.0008 to 1.0010; removing the seed argument, 0.8903 to 0.9586

What this does not show. The seed's share is bounded, not measured: a share of 0.30 would have been missed more often than not. The separation is a property of this benchmark at its own threshold and depth: it rests on the longest task type, whose hazard ratio is carried by four baseline takeoffs in thirty-two runs, it does not survive the ceiling gate at the final evaluation or a threshold of 0.65 and below, and it is not a horizon ladder, so we read it as neither smaller nor larger than the corridor's. This stack logs no per-game outcome, so a per-game difficulty adjustment is not identified. The registration fixed the bootstrap seed at 20260917; the scoring script draws with its constant 20260906 and prints it, the amendment that fixed the time of scoring records the constant as unchanged, and we used the script's value.

The evaluation-sampling split rests on four saved checkpoints per arm, not on the sixteen seed arguments, and with one training run per seed argument it measures the training component only as a complement. The three repeats of a checkpoint differ by two to five games of 128 in the method arm and by five to thirteen in the baseline arm, and their variance lies below the binomial reference because the 128 games are fixed and a policy near the ceiling is near-deterministic on most of them, so the binomial floor printed with the seed readings overstates the evaluation's share.

D.3 Four stacks under one read-only tool

Two of the five configurations whose logs we hold replay to the bit; in the other three a re-execution moves the endpoint between 0.70 and 15.93 times as far as a change of seed argument.

Setup. The read-only two-execution check released with the code, `stackcheck`, reads the logs of two runs of one command and prints, field by field and with no tolerance, whether they are the same computation; where a run under a second seed argument exists, it also prints how far re-executing the command and changing the seed argument move the endpoint, and their ratio. Table 19 is its output. Under the two-component model the ratio's expectation is at most one, so a reading above one at these counts is sampling or a component shared within an execution index, not a ratio of two effects. The second library's rows are calibration runs, with Qwen2.5-1.5B-Instruct, of two cells registered before any of their runs existed and since scored (Appendix B.6); they enter no registered prediction. The first two rows agree with the scored readings of §3 and of the public-stack experiment.

Table 19: Two configurations replay to the bit; in the others a re-run moves the endpoint 0.70 to 15.93 times as far as a change of seed argument. The check's output per configuration: the mean endpoint displacement of re-running the same command and of changing the seed argument, their ratio, the pairs read (re-run / seed change) and the first record at which the re-run pairs part.

stack	configuration	re-run $\lvert\Delta Y\rvert$	seed change $\lvert\Delta Y\rvert$	ratio	pairs	first differing record
this paper's pool (own logger)	the registered stack	0.060547	0.054199	1.12	8 / 4	training record 2
the public stack of the experiment (verl-agent)	as released	0.028500	0.023500	1.21	4 / 2	training record 1
a second public library (TRL 0.27.0)	colocated generation	0.000000	0.036086	0.00	4 / 1	bitwise replay
the same library	its released default: a generation server (the first two pairs)	0.111404	0.006994	15.93	2 / 2	evaluation record 0
the same library	its released default, 42 calibration pairs (dapo)	0.061239	0.047607	1.29	42 / 41	evaluation record 0
the same library	its released default, 42 calibration pairs (grpo)	0.041843	0.059611	0.70	42 / 41	training record 1
the same library	the server default, in-process engine	0.000000	–	–	34 / –	bitwise replay

Readings. We also ran four switch cells on the public stack, registered before any switch run existed. Under the first switch set and under the deterministic-kernel switch the nondeterminism ratio is 1.5008 $[0.8234, 1.9659]$ and 0.8799 $[0.2464, 1.5523]$ by the registered estimator with its seed-resampling band, and 1.403 $[0.648, 1.828]$ and 0.814 $[0.246, 1.513]$ as the ANOVA share with its exact interval, the two estimators of Appendix D.1; both are at or above the registered floor of 0.50: two hits. The re-execution variance under the released default against the stricter switch, $V(\text{default})/V(\text{stricter switch set}) = 1.957$, does not reach the one-sided $F_{0.95}(7, 7) = 3.787$ ($p = 0.1977$): a miss. The same ratio on a second, batch-invariant stack was scored as a miss, 0.249 against the registered critical 3.787.

What this does not show. The tool names no mechanism: a replay it reports is a statement about the runs it was given, not a promise about another machine.

D.4 The second public environment

On the field's own second public environment, at the final evaluation, at least 0.819 of run-to-run variance is not the seed argument: the exact 95% interval on the seed component is $[-0.484, 0.181]$, whose upper end is below one half.

Setup. We ran verl-agent's released Sokoban configuration, a 6×6 room with one box, at most 15 steps and visual observations, with Qwen2.5-VL-3B-Instruct, GiGPO (method; published comparand 81.0 ± 3.6) against GRPO (baseline; 67.1 ± 4.7), both as published, on sixteen seed arguments (400 to 640 spaced 16; eight declared same-host and eight different-host, alternating) with two executions each: sixty-four runs of 150 updates, evaluated every five steps on 128 puzzles that the seed argument generates, redrawn at every evaluation.

Before any run existed we fixed the registered prediction, the seed prediction of Appendix D.2 on this cell: the arms-pooled one-way intraclass correlation of the success rate at step 150 (an integer over 128; each arm centred on its own mean) with its exact F interval on $G - 1$ numerator degrees of freedom reads BOTH when the interval's lower end is above zero and its upper end below one half, SEED or BOUND when only the first or only the second holds, and NONE otherwise, after three guards in this order: an arm mean below 0.474 (the published GRPO band's lower end 0.624 less a margin of 0.150), both arm means at or above 0.90, and a zero within-seed mean square. It was scored once, with all sixty-four runs complete; the secondary readings below were registered beside it and not scored.

Readings. The registered outcome is BOUND: over all sixteen seed arguments and sixty-four runs the arms-pooled intraclass correlation is -0.1716, exact 95% interval $[-0.4838, 0.1813]$ ($F(31, 32) = 0.7071$, $p = 0.8314$), whose upper end is below 0.50 and whose lower end is not above zero. The between-seed mean square, 0.013591, is smaller than the within-seed one, 0.019221, so the point estimate is negative; we print the interval unclamped and read the share in the sentence above from its upper end. The pre-specified, unscored reading at the six-point window is -0.1622 $[-0.4764, 0.1907]$, pooled over the arms. The arm gap is 0.0165 against a published 0.1390, and each arm's standard deviation over its thirty-two runs is larger than the published three-seed spread; no run ended at or above 0.90, and neither the floor gate at 0.474 nor the ceiling gate fired. Although the four runs of a seed argument draw the same 128 puzzles at every evaluation, the step-0 record is identical in 28 of 32 pairs of executions and in no pair over the whole grid; the executions of a pair first differ at step 0 (four pairs), 5 (25 pairs) or 10 (three pairs).

Table 20: Per arm neither correlation resolves a seed component, the nondeterminism share is at or above its floor of 0.50 (re-executing a seed argument moves the endpoint further than changing it), and each arm mean is within 0.10 of its published value. Pre-specified readings of the Sokoban cell, not scored.

at the final evaluation	GiGPO (method)	GRPO (baseline)
intraclass correlation, exact 95% interval	-0.2245 $[-0.6298, 0.2847]$	-0.0697 $[-0.5244, 0.4230]$
nondeterminism share	1.4819	1.0932
arm mean (standard deviation over 32 runs)	0.7353 (0.1364)	0.7187 (0.1218)
range over the runs	0.117 to 0.867	0.367 to 0.883
published (Feng et al., 2025, Table 3)	0.810 ± 0.036	0.671 ± 0.047
rise from the step-0 mean of 0.1571	0.5782	0.5617

Two seed arguments, 416 and 496, already differ at step 0 across their four runs (16, 14, 15 and 17 puzzles solved of 128; 20, 21, 20 and 19), so the seed argument does not reproduce even the untrained policy's sampling. Same-host pairs of executions have a within-seed variance of 0.023664 (16 pairs) against 0.015760 for different-host pairs (15), $F(16, 15) = 1.5015$ with exact 95% interval $[0.5294, 4.1853]$, which contains one; the baseline arm's pair at seed argument 416, declared different-host, ran both executions on one machine and is left out of this contrast only.

What this does not show. The cell reads one horizon: Sokoban prints one success rate with no split by length, all 128 puzzles are 6×6 with one box, and the harder rungs do not learn within the released budget, so the cell says nothing about the horizon joint. The interval's lower end lies below zero, as it does when the executions of a seed argument scatter more than seed arguments do; we read it as no resolvable seed component, not as a negative one. The seed argument generates the evaluation puzzles, so its component contains the evaluation set, and the published comparands are means over a number of runs the paper does not state (it states three seeds for its ALFWorld and WebShop results only).

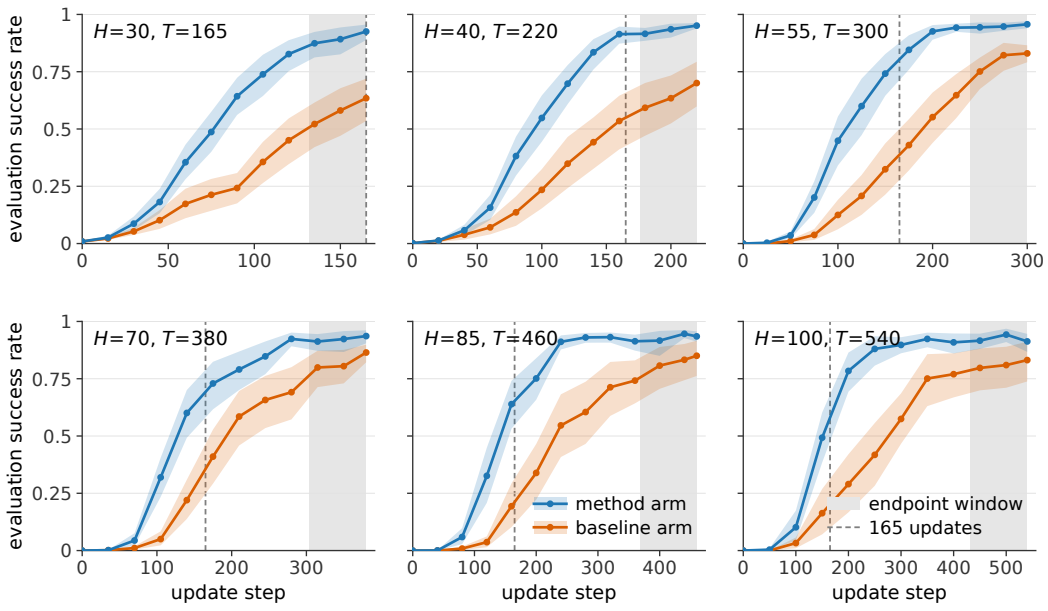


Figure 6: At every depth the method arm rises earlier, and the endpoint gap closes with depth through the baseline arm's rise over the longer budgets. Mean evaluation success of the 32 runs per arm ($\lambda = 0.95$) against the update step, with 95% bands from a joint seed-argument bootstrap of 2,000 replicates. Shaded: the endpoint window, $0.8T < \text{step} \leq T$, whose arm means are those of Table 6; dashed: the fixed count of 165 updates (Table 28).

E THE CEILING DECOMPOSITION AND THE SPECIFICATION UNIVERSE

We find that how much of the endpoint decay is lost headroom depends on the efficacy scale, that the gap grows from H=30 to H=100 once both arms are off the ceiling, and that under the registered budget no success-based specification grows with the horizon; Section 4 reports the headline of each.

Setup. Except for the off-ceiling cell, the one registered experiment here, every reading is computed on the 384-run pool (32 seed arguments per arm at each of six depths) after the pool had been read, and none was registered.

E.1 HOW MUCH OF THE ENDPOINT DECAY IS HEADROOM

On the relative failure reduction the baseline's shrinking headroom accounts for about three quarters of the log-decay of the endpoint gap from H=30 to H=100; on the other efficacy scales it accounts for less, and on the log success ratio for less than none.

Setup. Write $\Delta\text{Succ}(H)$ for the endpoint mean gap at horizon H and b for the baseline arm's endpoint success. The gap is the baseline's failure rate times the relative failure reduction $\text{RFR} = \Delta\text{Succ}/(1-b)$, the share of those failures that the method removes, so from the shallowest to the deepest cell the log change of the gap splits exactly into an efficacy term and a headroom term:

$$\Delta \ln \Delta\text{Succ} \;=\; \underbrace{\Delta \ln \text{RFR}}_{\text{efficacy}} + \underbrace{\Delta \ln(1-b)}_{\text{headroom}} \;=\; -0.2414 - 0.8099 \;=\; -1.0513.$$

We compute it as a ratio of means with a bootstrap interval, never as a mean of per-pair ratios: the per-seed-pair ratio $(m_i - b_i)/(1 - b_i)$ is heavy-tailed, with mean 0.24 and standard deviation 2.24 at H=30, and -0.07 and 2.05 at H=85. The decomposition does not use the ceiling gate of 0.90; the method arm is above that gate at five of six depths (Table 6), at all six under a gate of 0.85 and at none under 0.95.

Readings. The headroom term is 77.0% of the total, bootstrap 95% interval $[58.5\%, 113.9\%]$, $P(\text{headroom} > 50\%) = 0.994$; the identity is exact on every scale, the share is not (Table 21). There the residual accounting books to headroom what the change of the efficacy term leaves, and the counterfactual and Shapley accountings order the two terms instead. On the progress channel, the number of rooms a run reaches, whose ceiling is H itself, the decomposition books 100.7% to headroom while the endpoint reach gap grows from 5.445 to 8.826 rooms (flat in Table 23): it measures the ceiling, not the method, and the three rows of that channel that Table 23 reads as rising share that ceiling and are not decomposed.

Table 21: How much of the decay is headroom depends on the efficacy scale: 77.0% on the relative failure reduction, −17% on the log success ratio. The share of the log-decay of the endpoint gap from H=30 to H=100 booked to headroom, by efficacy scale and accounting.

efficacy scale	headroom share, residual	counterfactual and Shapley
relative failure reduction	77.0% [58.5%, 113.9%]	—
log failure ratio	57%	77%
logit	44%	63–70%
complementary log-log	23%	33–57%
log success ratio	−17%	—
progress channel (rooms reached)	100.7%	—

E.2 The off-ceiling cell at λ=1.00

Off the ceiling, with anchor coverage held constant, the endpoint gap grows from H=30 to H=100, and the hazard reading cannot be evaluated for want of takeoff events in the shallow cell.

Setup. We re-ran the depths H=30 and H=100 with the anchor pool held at λ=1.00, where anchor coverage does not thin with depth: one launch batch on one stack revision, sixteen seed arguments per arm and depth, one execution each, sixty-four runs.

Before any run existed we registered two predictions, on the two-point endpoint decay factor and on its separation from the two-point factor of the takeoff hazard ratio, with the pool's estimators and a seed-argument bootstrap of 2000 replicates, read at the budget multiple c^* that a registered ceiling rule selects on the window $0.8\,cH < \text{step} \le cH$: c=3 if both arm means are at or below 0.90 at both depths there, else c=2, else the cell is ceiling-bound and nothing is scored. Since c^* is not the registered 5.5, the cell's factors are the profile of another budget rule at another knob value, not the pool's profile re-read.

Readings. The rule selected c^*=2: at c=3 the H=100 depth is on the ceiling, at c=2 both depths are off it (Table 22). The endpoint prediction was scored as *grows*: the factor is 0.4389 $[0.2580, 0.7042]$, an interval below one, and on the five scales of Table 29 its interval lies below one on the risk difference, the logit and the log failure ratio. The method arm's mean at $W_3(100)$ is 0.9584, beside the pool's 0.898 at the same step.

The separation prediction was scored as not evaluable: by $2H$ two of the sixteen method runs and none of the baseline runs take off at θ=0.70 at H=30, below the event gate of four per arm and depth, so the hazard ratio is defined at H=100 only (6.1565); at c=3, a secondary reading, likewise (3.0354), and at θ=0.90 at both multiples (3.6708 at c=3, 13.5277 at c=2), so neither multiple has a two-point factor.

E.3 The twenty-four specifications, with a pair-level slope

Ten of the twenty-four specifications read a negative horizon trend and fourteen a positive one; every significant increase between the arms is on the rooms-reached channel, and read pair by pair the endpoint success gap decreases.

Setup. Crossing the three convention families of §4.2 gives six estimands times five (stopping rule, budget rule) pairs, a nominal thirty specifications; the iso-cost family of six cannot be computed,

Table 22: Off the ceiling the endpoint gap grows at the scored multiple, $c^*=2$, and the hazard reading has too few takeoff events at $H=30$ to be evaluated. Upper block: arm means at the two reading windows, the off-ceiling decision, and anchor coverage over $W_3(H)$ (all-update mean in parentheses) beside the pool's at $\lambda=0.95$; then the two-point factors, with the takeoff events at $\theta=0.70$ by cH. Lower block: the endpoint factor on the five scales of Table 29.

depth	arm	mean W_3	mean W_2	off at 3	off at 2	cov. W_3 (all)	pool cov. W_3
$H=30$	method	0.6953	0.3561	yes	yes	0.9683 (0.9382)	0.8153
$H=30$	baseline	0.2523	0.1172	yes	yes	0.9325 (0.9248)	0.8025
$H=100$	method	0.9584	0.8943	no	yes	0.9841 (0.9482)	0.3805
$H=100$	baseline	0.7847	0.3500	no	yes	0.9741 (0.9243)	0.4028

two-point factors ($H=30$ over $H=100$), point [interval], branch; events method/baseline at $H=30$, $H=100$		
$c=3$	endpoint F_{end}	2.5496 [1.0795, 5.7614] decays hazard F_{haz} n/a; events 12/0, 16/15
$c=3$	$D = \ln F_{\text{end}} - \ln F_{\text{haz}}$	n/a: not evaluable (the event gate)
c=2	endpoint F_{end}	0.4389 [0.2580, 0.7042] grows hazard F_{haz} n/a; events 2/0, 15/5
c=2	D	n/a: not evaluable (the event gate)

endpoint factor, by scale	$c^*=2$ (scored multiple)	$c=3$
risk difference	0.4389 [0.2580, 0.7042]	2.5496 [1.0795, 5.7614]
logit	0.5180 [0.2740, 0.8557]	1.0363 [0.5013, 2.1094]
log failure ratio	0.1737 [0.0827, 0.3304]	0.5459 [0.2509, 1.1318]
log success ratio	1.1846 [0.5997, 2.2225]	5.0678 [2.0128, 12.4610]
complementary log-log	0.7638 [0.4130, 1.2883]	1.9339 [0.8970, 4.1284]

because its read step falls between the evaluation records of the cell at $H=85$. The estimands are the success gap and the median success gap; the relative failure reduction, a ratio of arm means; the rooms-reached gap, the paired gap in the mean deepest room reached per training episode over the window, a count bounded by H, and the same gap over H; and the baseline failure rate, the share of baseline runs whose windowed success is below 0.5, a single-arm quantity.

The proportional budget rule reads each depth at its registered budget, the fixed count at 165 updates. The endpoint and fixed stopping rules read the registered endpoint window at the step the budget rule sets, the peak reads each cell at the step, among fixed fractions of its budget, where the mean success gap is largest, and the area under the curve is taken under each arm's curve and normalised by the budget. The six-mean trend is the Spearman correlation ρ_S across the six depth means, with exact p over all 720 orderings; DOWN and UP are significant at $\alpha = 0.05$, and FLAT is not resolvable from six grid points, not a claim of no trend.

A rank test on six depth means has six numbers to work with, while the thirty-two pairs behind each mean carry more, so we also regress each contrast's per-pair gap on $\ln H$ over the 6×32 seed-argument pairs (the median success gap by least absolute deviations), with a 95% interval from 5,000 replicates of a seed-argument bootstrap drawn jointly across both arms and all six depths; a slope's class is FLAT when that interval contains zero. The relative failure reduction's slope, marked †, is that of the six cell-level ratios on $\ln H$ and is not counted with the pair-level slopes.

Readings. Among the twenty specifications that contrast the arms, a restriction we made after the enumeration, seven six-mean trends are negative and thirteen positive; two decrease significantly, four increase, all on the rooms-reached channel, and fourteen are flat. The endpoint-success family that published work reports is flat in all four of its specifications, which is why §4 reports that trend without a significance claim.

Pair by pair, the sixteen contrasts with a per-pair value give three decreases, four increases, again all on the rooms-reached channel, and nine flat slopes, each keeping the sign of its six-mean trend; no success-based slope increases, and the endpoint success gap decreases, slope -0.2020 $[-0.2976, -0.1019]$ per unit $\ln H$. One class turns on the bootstrap: the rooms-reached gap over H at the peak, marked ‡, is flat under the joint bootstrap and rises under a depth-stratified one, $[+0.0199, +0.1985]$. A peak value is the largest of nineteen candidate readings and so is biased upward (Patterson et al., 2024, §3.2), here by at most about 0.03 on the success gap, which changes no row's class; the slopes re-select the step in every replicate.

Table 23: Ten specifications read a negative six-mean trend and fourteen a positive one, and every significant increase between the arms is on the rooms-reached channel. The twenty-four specifications that compute: the Spearman correlation ρ_S of the six depth means with its exact p and class, and the pair-level slope, in the estimand's units per unit $\ln H$, with its 95% interval and class.

estimand	stopping rule	budget rule	ρ_S	sign	p	slope [95%]	class
success gap	endpoint	proportional	-0.771	FLAT	0.1028	-0.2020 $[-0.2976, -0.1019]$	DOWN
success gap	peak	proportional	$+0.600$	FLAT	0.2417	$+0.0624$ $[-0.0784, +0.1931]$	FLAT
success gap	fixed	fixed count	$+0.257$	FLAT	0.6583	$+0.0326$ $[-0.1082, +0.1596]$	FLAT
success gap	area under curve	proportional	-0.657	FLAT	0.1750	-0.0259 $[-0.0980, +0.0458]$	FLAT
rooms-reached gap	endpoint	proportional	$+0.543$	FLAT	0.2972	$+1.8529$ $[-1.7946, +6.0478]$	FLAT
rooms-reached gap	peak	proportional	$+1.000$	UP	0.0028	$+24.6608$ $[+16.2022, +31.4922]$	UP
rooms-reached gap	fixed	fixed count	$+1.000$	UP	0.0028	$+23.4361$ $[+16.6861, +29.6683]$	UP
rooms-reached gap	area under curve	proportional	$+1.000$	UP	0.0028	$+10.0679$ $[+6.6172, +13.6427]$	UP
relative failure reduction	endpoint	proportional	-0.771	FLAT	0.1028	-0.1908 $[-0.3669, -0.0670]$	DOWN†
relative failure reduction	peak	proportional	$+0.086$	FLAT	0.9194	$+0.0194$ $[-0.1663, +0.1698]$	FLAT†
relative failure reduction	fixed	fixed count	-0.943	DOWN	0.0167	-0.2965 $[-0.4272, -0.1535]$	DOWN†
relative failure reduction	area under curve	proportional	$+0.657$	FLAT	0.1750	$+0.0399$ $[-0.0428, +0.1213]$	FLAT†
rooms-reached gap over H	endpoint	proportional	-0.543	FLAT	0.2972	-0.0937 $[-0.1546, -0.0277]$	DOWN
rooms-reached gap over H	peak	proportional	$+0.943$	UP	0.0167	$+0.1136$ $[-0.0035, +0.2117]$	FLAT‡
rooms-reached gap over H	fixed	fixed count	$+0.771$	FLAT	0.1028	$+0.1260$ $[+0.0309, +0.2108]$	UP
rooms-reached gap over H	area under curve	proportional	$+0.600$	FLAT	0.2417	$+0.0100$ $[-0.0454, +0.0650]$	FLAT
median success gap	endpoint	proportional	-0.943	DOWN	0.0167	-0.1590 $[-0.2695, -0.0977]$	DOWN
median success gap	peak	proportional	$+0.829$	FLAT	0.0583	$+0.1614$ $[-0.1463, +0.3451]$	FLAT
median success gap	fixed	fixed count	$+0.029$	FLAT	1.0000	$+0.1568$ $[-0.1244, +0.2942]$	FLAT
median success gap	area under curve	proportional	-0.371	FLAT	0.4972	-0.0360 $[-0.1480, +0.0633]$	FLAT
baseline failure rate	endpoint	proportional	-0.577	FLAT	0.2500	—	—
baseline failure rate	peak	proportional	-0.058	FLAT	0.9333	—	—
baseline failure rate	fixed	fixed count	$+0.941$	UP	0.0167	—	—
baseline failure rate	area under curve	proportional	-0.883	DOWN	0.0444	—	—

E.4 THE COMPOSITION SHARE ACROSS SUCCESS THRESHOLDS

The composition share is bounded, not identified: as the success threshold θ runs from 0.30 to 0.95 it spans $[0.00, 0.89]$ over the six cells.

Readings. Remark 2 (Appendix G.3) says that the composition share COMP$/\Delta$ is indexed by θ and not identified by the endpoint data, and the sweep makes this concrete (Table 24). The share rises with θ in every cell, though not monotonically: raising θ removes runs from the baseline's takeoff set faster than from the method's, so a larger part of the same gap is booked to composition. At the threshold fixed in advance the six point values average 0.430, a property of that threshold, so we report the bound and not the average; the point version at $\theta = 0.70$ is a registered prediction that will be scored when its deferred runs complete (Appendix N).

Table 24: The composition share is not identified: as the threshold runs from 0.30 to 0.95 it spans $[0.00, 0.89]$ over the six cells. COMP$/\Delta$ by cell, its range over θ and its value at $\theta = 0.70$; the last row gives the interval over the six cells and the mean of the six point values.

H	range over $\theta \in [0.30, 0.95]$	at $\theta = 0.70$
30	$[0.16, 0.89]$	0.520
40	$[0.21, 0.82]$	0.564
55	$[0.00, 0.57]$	0.179
70	$[0.00, 0.68]$	0.183
85	$[0.26, 0.79]$	0.524
100	$[0.24, 0.72]$	0.611
all six	$[0.00, 0.89]$	0.430

What this does not show. The specifications are one enumeration of the conventions, not a sample of them. The headroom share moves with the efficacy scale, and the composition share is bounded by its threshold, not identified. The off-ceiling cell has two depths, no fixed-count reading and one execution per seed argument, so its intervals resample runs whose seed and execution components enter together and bound neither on its own (§3).

F BUDGET RULES: THE MULTIPLE, THE FIXED COUNT AND THE FIVE SCALES

We find that the takeoff hazard ratio is stable at the registered threshold under every estimator we computed, that below the registered budget multiple the endpoint and hazard readings stop being distinguishable, and that at a fixed count of updates the endpoint gap stops decaying on the risk difference while it decays on one scale and grows on another (Sections 4.1 and 5 give the headlines).

Setup. The proportional budget rule gives a run cH updates, where c is the budget multiple; the fixed-count rule gives every horizon the same number. We registered the takeoff hazard ratio at $\theta = 0.70$ with the observed-over-expected estimator on the pool's six cells, and fixed the five scales before the analysis script read the pool; every other reading here, the fixed-count column of the five scales included, is exploratory.

F.1 THE TAKEOFF HAZARD RATIO: DEFINITION, ESTIMATOR AND THRESHOLD

At the registered threshold the method arm takes off faster at every depth, and the estimator we registered gives the smallest ratio of the four we computed.

Setup. The takeoff time of a run is the first evaluation step at which its success rate reaches a threshold θ, right-censored at the run's own step budget; the censoring is administrative and identical within a cell, so it is non-informative by construction. We compare takeoff times with a log-rank observed-over-expected (O/E) hazard ratio, $(O_m/E_m)/(O_b/E_b)$, where O and E count each arm's observed and expected takeoffs, with a 95% interval from the Poisson-type standard error $\sqrt{1/E_m + 1/E_b}$ and a tie-corrected log-rank test whose variance carries the $(n-d)/(n-1)$ correction the evaluation grid makes necessary; it is not a Cox fit.

The test, $(O_m - E_m)/\sqrt{V}$, and the interval coincide only for Peto's one-step estimator, for which they are algebraically the same number (we verified the identity to 10^{-12} in all six cells); that estimator is biased away from the null when the ratio is far from one, and Table 25 prints it for the identity alone. On this pool the O/E log hazard ratio is 0.51 to 0.72 times Peto's while the two standard errors differ only by a factor of 0.77 to 0.89, so the gap between our Wald and log-rank statistics lies in the point estimate, and in the conservative direction: the estimator we report attributes the least of the observed excess to the hazard ratio.

Table 25: The reported estimator gives the smallest ratio of the four in every cell, and the largest is up to 2.6× the smallest. The takeoff hazard ratio at $\theta = 0.70$, first crossing, under four estimators, with the log-rank z and the fraction of the training budget returned, the restricted-mean reading of §1.

cell	O/E (reported)	Cox–Breslow	Cox–Efron	Peto	log-rank z	budget returned
H=30	3.248	3.669	4.145	5.103	4.68	25.6%
H=40	2.944	3.502	4.238	5.177	4.86	22.7%
H=55	3.134	4.673	5.778	7.654	5.75	26.0%
H=70	2.293	2.502	2.825	4.048	4.17	18.5%
H=85	2.550	3.244	4.220	5.548	4.85	21.5%
H=100	2.745	4.003	5.349	7.120	5.34	22.0%

Readings. Six cells resolve no horizon dependence in the ratio at $\theta = 0.70$ or 0.50. At $\theta = 0.90$, a level the shallow baseline rarely reaches within its budget, the factor from H=30 to H=100 (the shallow ratio over the deep) is 2.34 $[1.25, 5.56]$ on the O/E estimator. At $\theta = 0.70$ the interval of that factor contains one under every estimator, its separation from the endpoint reading excludes zero under five estimators of six, and O/E gives the smallest separation (Table 26), so the registered estimator is conservative for whether the effect exists.

A robustness table in the full record re-reads the ratio with longer budgets, an early-versus-late split, a seed-stratified log-rank, midpoint event placement, a grouped-time model and two stricter takeoff definitions, and checks proportional hazards two ways; no re-read takes the ratio below one or changes its sign.

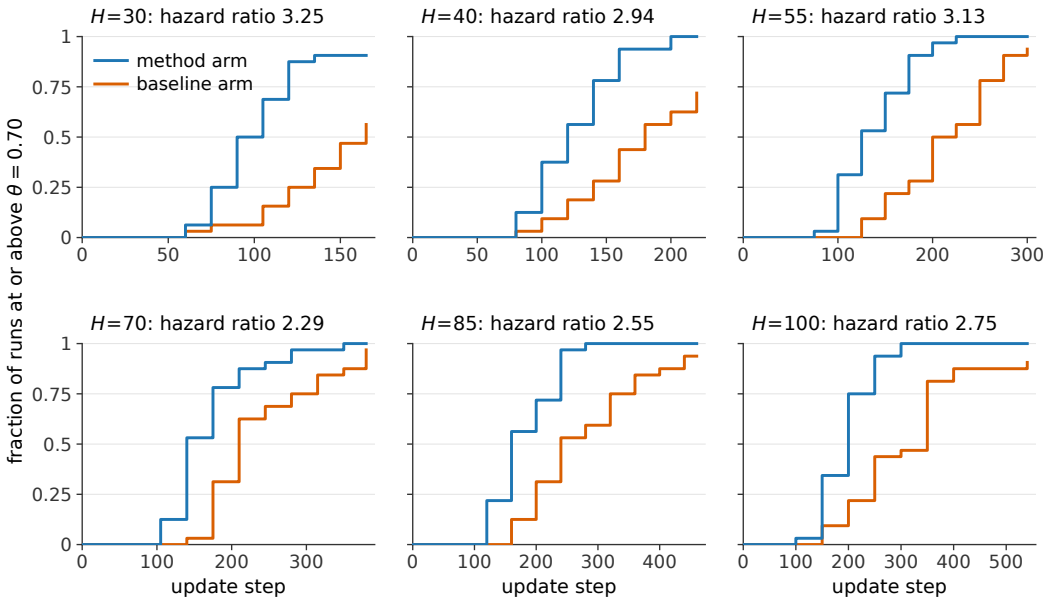


Figure 7: The method arm takes off earlier at every depth, and its hazard ratio stays between 2.29 and 3.25 from $H{=}30$ to $H{=}100$. Kaplan–Meier estimate of the fraction of the 32 runs per arm whose evaluation success has reached $\theta = 0.70$ (first crossing) by each evaluation step, right-censored at the cell's budget T (the right end of each axis); each panel prints the O/E hazard ratio of Table 25, re-derived from the event tables drawn here.

Table 26: Under every estimator the interval of the two-point hazard factor contains one, and its separation from the endpoint reading excludes zero under five of six. The factor from $H{=}30$ to $H{=}100$ at $\theta = 0.70$ and the separation, the endpoint-minus-hazard log-decay, with 95% intervals on the registered reading's own 5000 seed resamples; the grouped complementary log-log factor is defined on 4999 of them.

estimator	factor, $H{=}30$ over $H{=}100$	separation, endpoint − hazard
O/E (registered)	1.183 [0.740, 1.911]	0.883 [0.255, 1.814]
Cox, Breslow ties	0.916 [0.440, 2.196]	1.138 [0.212, 2.184]
Cox, Efron ties	0.775 [0.337, 2.153]	1.306 [0.257, 2.426]
Peto one-step	0.717 [0.306, 1.973]	1.384 [0.324, 2.551]
Mantel–Haenszel	1.063 [0.470, 2.978]	0.991 [−0.065, 2.107]
grouped complementary log-log	0.722 [0.292, 2.135]	1.377 [0.273, 2.606]

F.2 Off the registered budget multiple

As the budget multiple falls, the endpoint–hazard difference narrows, and its point estimate changes sign at $c = 3.5$ and below, while the difference from the restricted mean grows: below the registered budget the two survival readings fail in opposite directions.

Setup. Truncating each run at $\tau(c, H) = \min(B(H), cH)$ updates, where $B(H)$ is the registered budget, keeps a prefix of the same records, so a lower budget is a re-read, not a re-run. We audited that premise on the nearest available copies of the training code, not on the bytes that produced the pool, whose file the released tree carries as a copy recovered after the pool had run (the full record describes the recovery); in those copies the nine source lines the premise depends on, from the step bound to the evaluation trigger that fires off-grid at a run's final step, are identical.

Readings. Every interval of the endpoint–hazard difference below $c = 5.0$, the last budget multiple at which it excludes zero, contains zero, while the difference from the fraction of the budget returned (the restricted mean) excludes zero at every c (Table 27); the two readings are not distinguishable there on this pool, so their difference, registered at the proportional budget, is a result at that budget.

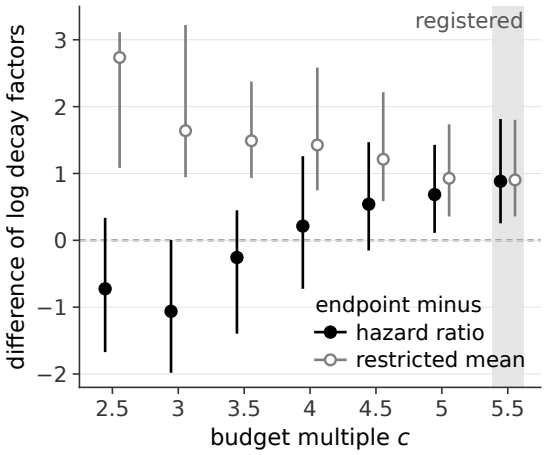


Figure 8: Off the registered budget the two survival readings fail in opposite directions: the endpoint–hazard difference narrows and changes sign at $c \leq 3.5$, while the endpoint–restricted-mean difference grows. Paired differences of the $H{=}30$-to-$H{=}100$ log decay factors, endpoint minus takeoff hazard ratio (filled) and endpoint minus the fraction of the budget returned (hollow), with the 95% intervals of Table 27; shaded: the registered budget, $c = 5.5$.

The $c = 3.0$ row is the one place where the interval turns on the bootstrap draw rather than on the data: it excludes zero in four of ten independent streams of the same 5,000 replicates, and a single 25,000-replicate bootstrap returns $[-1.984,\ 0.000]$ at $p = 0.050$, so we read it as containing zero; the registered row excludes zero on all ten streams.

The endpoint levels show the mechanism: from $c = 5.5$ to $c = 3.0$ the $H{=}100$ method arm moves from 0.924 to 0.889 and the baseline arm from 0.813 to 0.496, so the $H{=}100$ gap grows from 0.111 to 0.393, a factor of 3.53, while the shallow cell's gap peaks at $c = 3.5$ (0.391) and is 0.274 at $c = 2.5$: the ceiling mechanism of §4, read from the other side.

Table 27: Below the registered budget the endpoint–hazard difference narrows and turns negative in point estimate, while the difference from the restricted mean grows. Paired differences of log decay factors from $H{=}30$ to $H{=}100$ (top row: the registered budget; τ: the truncation point at $H{=}30$) on the 5,000 replicates of §1, with the bootstrap p of the hazard column ($p < 0.001$ throughout for the restricted mean); last two columns: the shallowest cell's arms at $\theta = 0.70$.

c	τ	endpoint − hazard	p	endpoint − restricted mean	censored m/b	events m/b
5.5	165	+0.883 [+0.255, +1.814]	0.004	+0.903 [+0.358, +1.802]	9.4%/43.8%	29/18
5.0	150	+0.685 [+0.111, +1.428]	0.018	+0.928 [+0.356, +1.736]	9.4%/53.1%	29/15
4.5	135	+0.541 [−0.153, +1.468]	0.138	+1.212 [+0.587, +2.215]	9.4%/65.6%	29/11
4.0	120	+0.213 [−0.725, +1.258]	0.625	+1.426 [+0.746, +2.585]	12.5%/75.0%	28/8
3.5	105	−0.257 [−1.397, +0.449]	0.498	+1.489 [+0.933, +2.374]	31.2%/84.4%	22/5
3.0	90	−1.062 [−1.982, +0.004]	0.051	+1.640 [+0.943, +3.221]	50.0%/93.8%	16/2
2.5	75	−0.725 [−1.673, +0.336]	0.214	+2.735 [+1.084, +3.113]	75.0%/93.8%	8/2

Two of the five readings of the horizon effect are ratios of Kaplan–Meier medians, and such a median is not always reached. On a separate bootstrap draw, 1469 of 5000 replicates leave undefined the shallow baseline's sustained median, its median time to hold the threshold, and 951 its first-attainment median, its median time to reach it. Almost all fall in the shallowest cell (46 and 15 of the same 5000 fall at $H{=}40$), and none at $H{=}55$ or deeper. The dropped replicates are those whose ratio would be largest; of the two substitutions we computed, the more conservative, an unreached median replaced by twice the budget, still gives $[0.16, 2.32]$ for the endpoint-minus-sustained-median difference.

The hazard ratio does not depend on where inside its evaluation interval an event falls, but the median readings do: one grid step of resolution puts the two ratios anywhere in $[0.71, 1.47]$ and $[0.00, 1.38]$.

F.3 A FIXED COUNT OF UPDATES: THE GAP AT 165, 300 AND 540 UPDATES

Held at the shallowest cell's 165 updates, the endpoint gap stops decaying on the risk difference because both arms fall with depth: a fixed count moves the read-out along the learning curve, and neither budget rule reads efficacy.

Setup. At a fixed count T each depth takes the registered read-out, the mean over the evaluation records with $0.8T < \text{step} \leq T$, and enters only if its runs reach T: at 300 updates the $H{=}40$ runs, which stop at 220, drop out, and at 540 only $H{=}100$ remains of the pool. Beyond its own budget of 165, the shallow end comes from the five-fold-budget cell at $H{=}30$ (marked * in Table 28), sixteen seed arguments of a later launch batch on a different stack revision.

Table 28: Held at 165 updates the gap stops decaying, because both arms fall with depth. Endpoint arm means and gaps at a fixed count of T updates, each depth read over its records with $0.8T < \text{step} \leq T$ (at 165 updates the four deepest cells on a single record, at step 140 to 160); last row, the Spearman correlation of the gap with H and its exact two-sided p.

	$T = 165$			$T = 300$			$T = 540$		
H	method	baseline	gap	method	baseline	gap	method	baseline	gap
30	0.8971	0.5789	0.318	0.9662*	0.8178*	0.148*	0.9611*	0.8799*	0.081*
40	0.8744	0.4886	0.386	—	—	—	—	—	—
55	0.7417	0.3242	0.417	0.9493	0.8010	0.148	—	—	—
70	0.6006	0.2201	0.380	0.8857	0.6743	0.211	—	—	—
85	0.6390	0.1935	0.446	0.9302	0.6046	0.326	—	—	—
100	0.4927	0.1631	0.330	0.8889	0.4961	0.393	0.9239	0.8127	0.111
	$+0.2571$ ($p = 0.6583$), six depths			$+0.9000$ ($p = 0.0833$), five depths			not defined, two depths		

Readings. At 165 updates the gap has no monotone trend with depth (Spearman $+0.26$, against the 0.886 criterion): the baseline's rise with depth under the proportional budgets becomes a fall at this count, so the flat sequence sets a gap read mid-learning at the deep end against one read at the end of learning at the shallow end. At 300 updates, where every depth but $H{=}40$ can be read, the gap rises with the horizon in point estimate.

The count is not what decides it. At 540 updates, the deepest cell's budget and 3.3 times the shallow cell's, the two ends are not distinguishable: the five-fold-budget cell, both of whose arms are near the ceiling, gives 0.081 $[0.049, 0.125]$ on its sixteen re-executed seed arguments, against 0.111 $[0.041, 0.194]$ on the thirty-two pairs at $H{=}100$ (2,000 seed-argument resamples). A cell trained for 165 updates outright at $H{=}100$ gives 0.300 on sixteen seed arguments, against 0.339 for the same arguments' 540-step runs read at ≤ 165 (0.330 over all thirty-two).

F.4 THE FIVE SCALES UNDER EACH BUDGET RULE

Whether the endpoint difference decays depends on the scale it is measured on: at the fixed count only the log failure ratio decays and only the log success ratio grows, while under the proportional budget every scale decays in point estimate.

Setup. One estimand, the endpoint difference that published work reports, is read on five scales applied to the arm means: the risk difference (in success rate), the logit (in log-odds), the log failure ratio (LFR, the log of the baseline's failure rate over the method's), the log success ratio (the log of the method's success rate over the baseline's) and the complementary log-log difference.

Every pool interval here resamples seed arguments jointly across both arms and all six depths. That is the right unit even though §3 finds that the seed argument carries little of the run-to-run variance: what is resampled is the paired run the seed argument indexes, and §3 denies that the label names a population of its own, not that the runs are draws. Resampling runs instead would drop the pairing the sign tests need, and would not narrow every interval, since the six seed-pairing correlations on this pool run from -0.39 to $+0.78$.

Readings. Under the registered budget the factor's interval excludes one on every scale but the log failure ratio, and only barely on the logit; a second bootstrap of 5,000 replicates returns the same

Table 29: At the fixed count only the log failure ratio decays and only the log success ratio grows; under the proportional budget every scale decays in point estimate. Two-point factors $F_S = S(30)/S(100)$ of the endpoint difference, with 95% intervals from 2,000 seed-argument replicates drawn jointly; $S(30)$ is common to the two rules, whose budgets coincide at $H{=}30$. The risk-difference row is the factor §1 prints as $2.86\times$ $[1.47, 7.90]$ from 5,000 replicates, re-drawn on this table's stream, hence $[1.425, 7.507]$.

		fixed count, 165 updates		proportional budget	
scale	$S(30)$	$S(100)$	F_S [95%]	$S(100)$	F_S [95%]
risk difference	0.3182	0.3296	0.965 [0.573, 1.617]	0.1112	2.861 [1.425, 7.507]
logit	1.8474	1.6061	1.150 [0.596, 2.049]	1.0287	1.796 [1.023, 3.975]
log failure ratio	1.4094	0.5006	2.816 [1.668, 4.959]	0.9005	1.565 [0.912, 3.392]
log success ratio	0.4380	1.1056	0.396 [0.178, 0.762]	0.1283	3.415 [1.539, 9.784]
complementary log-log	0.9667	1.3381	0.722 [0.359, 1.306]	0.4302	2.247 [1.223, 5.286]

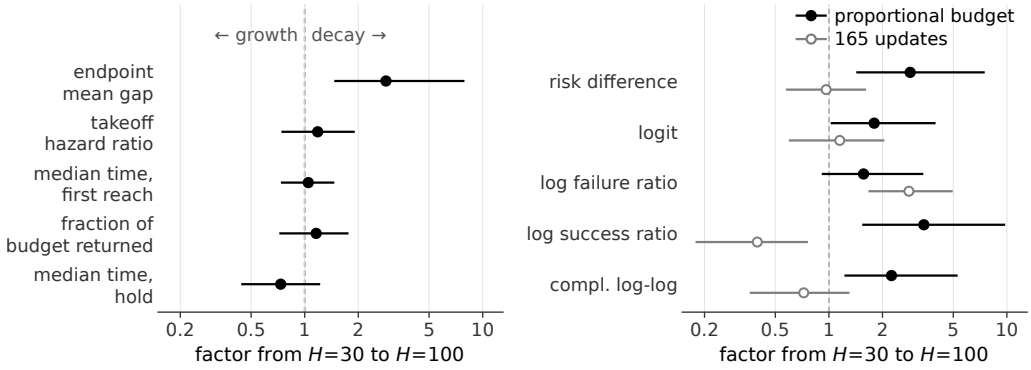


Figure 9: Whether the benefit decays with the horizon depends on the reading and on the scale: only the endpoint mean gap excludes one among the five readings, and at a fixed count of updates the scales disagree in direction. Two-point factors $F = S(30)/S(100)$ on a log axis, above one a decay. Left (Table 32): the five readings of §1. Right (Table 29): the endpoint difference on five scales at the registered budget (filled) and at 165 updates (hollow).

classification on every scale under both rules. The takeoff-hazard factor at the registered budget is 1.183 (1.366 $[0.844, 2.215]$ when takeoff requires two consecutive crossings). Three of the five scales separate from the hazard reading and two do not: the logit (paired log-difference from the hazard reading 0.417 $[-0.053, 1.128]$) and the log failure ratio (0.280 $[-0.171, 0.979]$). The other four readings of §1, no two of which separate, could have disagreed: the design had the resolution to tell them apart, and did not (Appendix H.2).

The fixed-count decay on the log failure ratio is the one Table 23 reads across six depths: the relative failure reduction on the arm means is $1 - \exp(-\mathrm{LFR})$, and its six-depth sequence at 165 updates, from 0.756 at $H{=}30$ to 0.394 at $H{=}100$, is that table's fixed-count row. The interaction on the risk difference is one that a monotone transform removes when one arm is at the ceiling, the removable interaction of Loftus (1978) and Wagenmakers et al. (2012).

What this does not show. The hazard reading's flatness across depth is a statement at the registered threshold: six cells cannot resolve a horizon dependence in it, which is not the same as its having none. The main text makes its cross-reading claims only at the registered budget; the readings off that budget locate where the claim stops and are not findings of their own. Neither budget rule reads efficacy, and whether a reading shows decay depends on its scale.

G THE IDENTIFICATION REMARKS

We show that under a proportional budget the endpoint gap sequence cannot tell a speed advantage from a composition difference (Remark 1), that the composition share is indexed by a convention

and not identified by the data (Remark 2), and that one constant speed advantage reads as a rise and then a fall at a fixed update count (Remark 3); fitted to this pool, neither pure model of Remark 1 reproduces the six depths.

G.1 Remark 1 and its construction

Remark 1 (speed and composition are observationally equivalent under a proportional budget). Fix a horizon grid $H_1 < \cdots < H_k$ and the budget rule $B(H) = cH$ in updates (under a fixed budget the remark does not apply), and write $\Delta(H)$ for the effect. *Every gap sequence in* $(0, 1)$ *compatible with two monotone outcome curves bounded in* $[0, 1]$*, and whose increases sum to at most* $\Delta(H_k)$ *(every non-increasing sequence qualifies), is reproduced exactly both by* (a) *a pure time-rescaling model, in which the arms share one learning curve and differ by a constant rate factor, and by* (b) *a pure composition model, in which they differ only in the probability of having taken off by the budget; neither uses horizon-dependent efficacy.*

In (a) the two arms share one learning curve F and differ only by a constant rate factor α (not one particular value, but every α at or above some threshold $\alpha_0 > 1$) that does not depend on H, so that $\mathbb{E}[Y_b(H)] = F(cH)$ and $\mathbb{E}[Y_m(H)] = F(\alpha cH)$; in (b) the two arms share one post-takeoff curve G and one never-takeoff level μ and differ only in the probability $\pi_a(H)$ that a run has taken off by the budget, so that $\mathbb{E}[Y_a(H)] = \pi_a(H)\, G(cH) + (1 - \pi_a(H))\, \mu$.

Here $Y_a(H)$ is the endpoint outcome of arm $a \in \{m, b\}$ (method, baseline) at horizon H, measured in a window at the end of the budget, and $\Delta(H) = \mathbb{E}[Y_m(H)] - \mathbb{E}[Y_b(H)]$.

Construction. For (b), set G and μ to any pair with $G(cH_j) - \mu = g_j > 0$ and put $\pi_m(H_j) - \pi_b(H_j) = \Delta(H_j)/g_j$, which is admissible whenever $g_j \geq \Delta(H_j)$; the arms then differ in nothing but how many runs have learned, and the entire sequence is matched by construction. For (a), the $2k$ values $F(cH_j)$ and $F(\alpha cH_j)$ are constrained by the k equations $F(\alpha cH_j) - F(cH_j) = \Delta(H_j)$ together with monotonicity. Take any $\alpha \geq H_k/H_1$, so that every cH_j lies at or below every αcH_j: a monotone F through the $2k$ points then exists exactly when the increases of the gap sequence sum to at most $\Delta(H_k)$ (take $F(cH_j) = b_j$ and $F(\alpha cH_j) = b_j + \Delta(H_j)$ with the b_j below), and the arms then differ in nothing but speed. Neither construction uses any horizon-dependent efficacy: in (a) the rate factor α is a single constant, and in (b) the post-takeoff curve is shared. □

The hypothesis holds on this pool. Write b_j and m_j for the two curves' values at the jth depth. The measured gap sequence is $\Delta = (0.318, 0.292, 0.148, 0.101, 0.102, 0.111)$. Set $b_1 = 0$ and $b_{j+1} = b_j + \max(0, \Delta_j - \Delta_{j+1})$: only three of the five consecutive differences are positive, and they sum to $0.026 + 0.144 + 0.047 = 0.217$, so b is non-decreasing and ends at 0.217. Then $m_j = b_j + \Delta_j$ is non-decreasing as well, ends at 0.328, and both sequences lie inside $[0, 1]$. A pair of monotone outcome curves compatible with this pool's gap sequence therefore exists; its increases sum to $0.001 + 0.009 = 0.010 \leq \Delta(H_6) = 0.111$ and $b_6 = 0.217 \leq m_1 = 0.318$, which is what the hypothesis of Remark 1 asks for.

G.2 Remark 1's two models, fitted to this pool

Fitted to the $H{=}30$ trajectories and extrapolated, the pure time-rescaling model over-predicts the decay and the pure composition model does worse; matching the six endpoint levels as well as the gaps needs a rate factor about twice the one the trajectories support.

Setup. Remark 1(a) can be fitted rather than asserted. The remark is free to choose F to match the gap sequence; our fit asks more, requiring F to be the curve both arms' own mean evaluation trajectories trace at $H{=}30$: one shared monotone F and one rate factor α, fitted isotonically on twenty-four pooled grid means of the $H{=}30$ pair of cells, so that its R^2 measures monotone interpolability rather than parsimony.

Readings. The fit reaches $R^2 = 0.9997$ at $\hat{\alpha} = 1.68$. The criterion is numerically flat over $[1.68, 1.74]$, and a bootstrap of 200 draws over the thirty-two seed pairs puts α at a median of 1.75 with a 90% band of $[1.50, 2.00]$, which is also its 95% band. Held fixed and extrapolated under the same proportional budget, that fit gives 0.317 at $H{=}30$ and predicts 0.114, 0.013, 0.000, 0.000 and 0.000 at the other five depths, against the observed 0.318, 0.292, 0.148, 0.101, 0.102 and 0.111. The

pure time-rescaling model does not under-predict the reported decay; it over-predicts it, and predicts a gap that has vanished by $H{=}70$. The composition analogue, Remark 1(b), fitted and extrapolated under the same discipline, does worse still: its RMSE is 0.162 against 0.118, and it takes the gap to zero by $H{=}40$.

Fitted instead to all six endpoint levels at once, the pooled residual reaches its floor only for α at or above 3.34, the first point of a 0.01 grid above the algebraic threshold $H_k/H_1 = 3.33$. Matching the levels as well as the gaps forces every admissible α above H_k/H_1, because the method arm's endpoint level at the shallowest depth already exceeds the baseline's at the deepest, and a single monotone F can accommodate that only if the method arm's rescaled budget at H_1 passes the baseline's budget at H_k. The rate factor the within-cell trajectories support, 1.68, and the rate factor the cross-depth levels require, 3.34 or more, are not the same number; the second is a property of the pool, not of the method.

None of these fits contradicts Remark 1. Its hypothesis is that the gap sequence be compatible with *some* pair of monotone outcome curves, with increases that sum to at most $\Delta(H_k)$; the construction above gives such a pair, and the arithmetic above shows that this pool satisfies it. The pooled level fit asks for more: one shared F through the measured levels themselves. Those levels are not monotone in budget (the baseline arm reads 0.830 at $H{=}85$ and 0.813 at $H{=}100$), so that fit has a residual floor. The floor it reaches on the levels, 5.85×10^{-4}, is exactly the isotonic residual of the two observed level sequences taken separately; on the gap sequence alone the smallest residual is 4.44×10^{-4}.

G.3 Remark 2, and the decomposition it indexes

The composition decomposition. Suppose one adopts model (b) and asks how much of Δ is composition. The standard decomposition writes

$$\Delta \;=\; \underbrace{(\pi_m - \pi_b)\big(\mu_b^{\text{post}} - \mu_b^{\text{never}}\big)}_{\text{COMP}} \;+\; \underbrace{\pi_m\big(\mu_m^{\text{post}} - \mu_b^{\text{post}}\big) + (1-\pi_m)\big(\mu_m^{\text{never}} - \mu_b^{\text{never}}\big)}_{\text{COND}},$$

where π_a is the fraction of arm a's runs that have crossed a success threshold θ, and μ_a^{post} and μ_a^{never} are that arm's mean outcomes among the runs that have and have not.

Remark 2 (the composition share is indexed by a convention, not identified by the data). *The target parameter is the composition share with respect to the latent takeoff event, of which every θ-thresholded takeoff is an admissible proxy. The threshold θ is a reporting convention. The endpoint data place no restriction on it, and the ratio* COMP$/\Delta$ *of the decomposition above varies with it. Consequently* COMP$/\Delta$ *is not point-identified; the data identify an interval, obtained by sweeping θ over its admissible range.*

In our pool the identified interval is $[0.00,\ 0.89]$, essentially the whole of it, and we report a Manski-style bound. Appendix E.4 gives the sweep behind it, its value at the threshold fixed in advance, and the status of the registered prediction on its point version.

G.4 Three instances on this pool

Three readings of this pool are instances of the remarks. The θ sweep of Appendix E.4 is Remark 2 on our own data. The five readings of §1 are Remark 1's summary problem in one line: five quantities, one sentence, and a sequence of gaps that does not say which quantity the sentence is about; the full record works both through. The third is a constructive null.

A constructive null. Remark 1 concerns which mechanism produced an effect; a constructive null is stronger. We built a synthetic comparison in which horizon-dependent efficacy is removed from the method arm by construction, the baseline arm keeping its measured distributions. Resampling from this null 4000 times under the protocol of published work yields an in-grid crossing of the half-effect level in 98.0% (3920 of 4000) of resamples; the crossing is the depth at which the gap falls to half its value at $H{=}30$. Its median is 52.3, 95% $[48.6, 72.7]$, which covers the measured 54.6. Holding the method arm fixed makes the crossing a property of the baseline curve alone, so the 98.0% is a statement about the readout, not about the method. The reported shape is thus compatible not only with either mechanism, but with neither.

G.5 What separates the two models, and what it costs

The two models are told apart on the event-time scale, not by the endpoint, and that scale costs a trajectory for every run.

Readings. The remarks are statements about a particular observable, the endpoint under a proportional budget, and not about the underlying process, which is perfectly well defined. Models (a) and (b) differ sharply in a quantity that is measurable: the distribution of the time at which a run first attains a given level. Under (a) the two arms have proportionally shifted takeoff distributions and equal eventual attainment; under (b) they have equal post-takeoff behaviour and unequal attainment.

On that scale the takeoff hazard ratio exceeds one at all six horizons, every 95% interval excluding one and log-rank z between 4.17 and 5.75 (Table 25). The takeoff probability itself differs too: at $\theta = 0.70$ it is between 0.906 and 1.000 for the method arm, against 0.562 to 0.969 for the baseline. Within a finite budget both pure forms predict a hazard ratio above one and a lower takeoff share for the baseline, since equal eventual attainment lies beyond the budget, so these readings do not by themselves reject either form; §5 reports what the event-time data support. From the endpoint side the fits of Appendix G.2 reach a conclusion of their own: fitted on the $H{=}30$ trajectories and extrapolated, neither pure model reproduces the measured sequence.

The cost of this move may be why published work has not made it. A takeoff distribution requires the evaluation trajectory of every run, at a grid fine enough to resolve the event, and a replication count large enough for a survival comparison to have power. Reports in this subfield give endpoints, at a modal replication count of three.

Relation to existing frameworks. Neither remark is new mathematics. Remark 2 is an instance of partial identification in the sense of Manski (2003): a parameter that is a function of the data *and* of a convention the data do not pin down is identified only up to the range of that convention, and the honest report is the range. Remark 1 is the observational-equivalence half of the same idea, specialised to the budget rule that agent RL uses. The addendum on estimands of clinical-trials guidance (International Council for Harmonisation of Technical Requirements for Pharmaceuticals for Human Use, 2019) requires a trial to state which quantity it is estimating, separately from how it is estimated, because the same summary statistic can be produced by treatment mechanisms with different meanings. What we add is the demonstration: a pool large enough to exhibit the non-identification on the exact quantity published work reports, with both constructions instantiated rather than described.

G.6 Remark 3, in four lines

Remark 3 says that one constant speed advantage reads as a rise and then a fall at a fixed update count and can read as decay under a proportional budget.

Take Remark 1(a)'s model: one learning curve F, a constant rate factor $\alpha > 1$, a time constant $\tau(H)$ increasing in H, so that at a budget fixed at T updates the arms sit at $F(u)$ and $F(\alpha u)$ with $u = T/\tau(H)$. Write $v = \ln u$ and $G(v) = F(e^v)$; the gap is $g(v) = G(v + \ln\alpha) - G(v)$ and $g'(v) = G'(v+\ln\alpha) - G'(v)$. Let G' be unimodal with peak at m. Then $g' \geq 0$ while $v + \ln\alpha \leq m$ and $g' \leq 0$ while $v \geq m$; on the remaining interval $m - \ln\alpha < v < m$, $G'(v)$ is rising and $G'(v + \ln\alpha)$ is falling, so g' is decreasing there. A function that is non-negative, then decreasing, then non-positive changes sign at most once, so g is unimodal in v and therefore in the horizon, which enters only through $\tau(H)$: the fixed-count gap rises with the horizon while the budget is more than enough and falls once it is not. Under a proportional budget $u = cH/\tau(H)$ instead: if τ grows more slowly than H, u rises with the horizon and the same g is traversed in the other direction, which past its peak reads as decay; if $\tau \propto H$, u and the gap are constant.

What this does not show. Remark 1 is an existence statement: a model of each kind reproducing a given gap sequence exists, and we make no claim that this pool was generated by either one. Remark 3 is an observation about the model of Remark 1 and not a theorem about our pool: it assumes the pure time-rescaling model and a unimodal G' (the logistic, Gompertz and Weibull takeoff shapes satisfy it), and it says nothing about where on the curve any particular set of horizons and budgets

sits, which is an empirical question the fixed-count profile of §4.1 answers and this argument does not. Nor does it say, and nothing here measures, what a hazard ratio read at a fixed count would do.

The rate factor the model holds constant is constant *by assumption* inside the model; a measured hazard ratio estimates it only where the two arms' hazards stay proportional, which a pure time-rescaling does not give, so the two are different objects and neither licenses a sentence about the other. What this pool does on the proportional hazards question is checked two ways in Appendix F.

H SAMPLE SIZE, POWER AND WHAT EACH STATISTIC COSTS

On this pool an exact paired test cannot reject at all below six runs per arm, and at eight runs the best statistic detects the shallow cell with power 0.80 while no statistic detects the deep one. The four horizon readings that show no decay are measured more precisely than the endpoint reading that shows one, so their agreement is a measurement and not a lack of resolution.

Setup. None of these readings is registered. We counted detection by gap size over the comparison rows of the matrix of configurations fixed before any was scored, a row being one comparison read at one replication count, and again on a later snapshot of the pool after dropping any seed pair whose endpoint is undefined in either arm; we estimated the power of six statistics on the pool's $H{=}30$ and $H{=}100$ cells by subsampling the same seed arguments in both arms without replacement; and we read the precision of the five horizon readings of §1 from that section's seed-argument bootstrap.

H.1 WHAT A REPLICATION COUNT DETECTS, BY STATISTIC

Section 6.1 gives the floor of an exact sign or signed-rank test on n pairs, a two-sided p of 2^{1-n} whatever the effect: 0.25 at three pairs and 0.0625 at five, so no three-run study can reach $\alpha = 0.05$ and $n \geq 6$ is an arithmetic requirement rather than a recommendation.

On this pool every one of the thirty-six comparison rows we score at $n \leq 5$ is a miss under the exact sign test. The endpoint reading is the weak one: at $H{=}100$ the method arm's endpoint exceeds the baseline's in 22 of 32 seed arguments (exact two-sided $p = 0.050$), while the takeoff hazard ratio of the same cell has a log-rank z of 5.34. At $H{=}30$ the endpoint count is 28 of 32 ($p = 0.0000193$).

Detection depends on the size of the gap more than on anything a practitioner controls (Table 30, where a dash marks a band that a count does not report): five replications detect none of the rows below a gap of 0.05 and still miss 21% of those above 0.30, where eight miss none, and the repeat on a later snapshot of the pool gives the same three readings. On the pool's own two cells Table 31 gives probabilities instead of single realisations: at $n = 8$ the $H{=}30$ cell (gap 0.318) is detected with probability 0.29 to 0.86 depending on the statistic, and the $H{=}100$ cell (gap 0.111) with 0.03 to 0.23.

The choice of test matters as much as n. On another count, a Wilcoxon signed-rank test in place of a t-test at eight replications lowers the miss rate on the success reading at $H{=}40$ from 0.264 to 0.011, a factor of about twenty-four, and by a factor of seven on a second comparison; on the two cells of Table 31 it is no more powerful than t at $n = 8$, and on one deep cell the median is the worse statistic. We therefore report a mean-based and a rank-based test together.

In Table 31 a prescribed count ($n@.80$, $n@.50$) is the smallest at which power reaches that level and stays there for every larger count up to eight, not a first crossing; S is the probability that a significant result has the wrong sign and M the median factor by which it exaggerates the truth; where an exact test cannot reach $\alpha = 0.05$ at that n, its power is 0.00 by construction and NA marks S and M.

Three-run error bars. A standard deviation computed from three runs is 11.4% low on average, and its interval for σ runs from 0.52 to 6.29 times the printed value (§6.1). The ends of that interval differ by a factor of 12.1, against 3.9 at six replications and 1.7 at thirty-two, and under the measured shape of the pool it covers 0.90 rather than 0.95.

Intervals for a ratio of two variances are as wide at the degrees of freedom of our designs: the ends differ by a factor $F_{0.975}/F_{0.025}$ of 12.7 at $(8, 16)$ and at $(7, 24)$, 7.9 at $(15, 16)$, 6.1 at $(15, 30)$, 5.7 at $(16, 32)$, 4.3 at $(30, 30)$, 3.5 at $(30, 64)$ and 2.7 at $(60, 64)$. The relative standard error of one

Table 30: Five replications detect none of the smallest comparisons and still miss a fifth of the largest; eight detect every row above a gap of 0.15. Rows detected by a mean test at five and at eight replications, by the size of the measured gap, over the 106 scoreable comparison rows of the matrix and, in a repeat on a later snapshot of the pool, over 127 rows; one realisation per row.

	first count		later snapshot	
measured gap	$n = 5$	$n = 8$	$n = 5$	$n = 8$
below 0.05	0 of 7	–	0 of 7	2 of 5
0.05 to 0.15	5 of 20	7 of 7	–	8 of 9
0.15 to 0.30	5 of 10	9 of 9	–	7 of 7
above 0.30	19 of 24 (21% missed)	16 of 16	23 of 28 (18% missed)	16 of 16

Table 31: At eight replications the best statistic reaches power 0.80 at H=30, and no statistic does at H=100. Power (pw) of six statistics against the ground truth of the pool, subsampling the same seed arguments in both arms without replacement (finite-population factor 0.774 at $n = 8$), with the smallest counts that reach power 0.80 and 0.50 and, at three and eight replications, the sign (S) and magnitude (M) errors of a significant result.

		prescription		$n = 3$			$n = 8$		
cell	statistic	n@.80	n@.50	pw	S	M	pw	S	M
H=30	Welch t	8	3	.54	.00	1.3	.82	.00	1.1
H=30	paired t	>8	2	.57	.00	1.3	.78	.00	1.1
H=30	exact sign	>8	>8	.00	NA	NA	.29	.00	1.1
H=30	exact signed-rank	>8	7	.00	NA	NA	.72	.00	1.0
H=30	exact rank-sum	8	5	.00	NA	NA	.86	.00	1.0
H=30	bootstrap median	8	2	.56	.00	1.1	.83	.00	1.0
H=100	Welch t	>8	>8	.14	.02	0.6	.13	.00	0.9
H=100	paired t	>8	>8	.18	.04	0.7	.23	.00	1.2
H=100	exact sign	>8	>8	.00	NA	NA	.03	.00	2.4
H=100	exact signed-rank	>8	>8	.00	NA	NA	.23	.00	2.0
H=100	exact rank-sum	>8	>8	.00	NA	NA	.20	.00	1.7
H=100	bootstrap median	>8	>8	.14	.02	2.0	.15	.00	1.7

variance, $\sqrt{2/d}$, is 0.50, 0.35, 0.25 and 0.18 at $d = 8$, 16, 32 and 64, so halving it takes four times the runs.

H.2 How precise the five horizon readings are

In log-decay the four readings that show no decay (the takeoff hazard, the two median times and the fraction of the budget returned) have standard errors between 0.18 and 0.26, against 0.42 for the endpoint reading (Table 32); each would have resolved a decay of the endpoint's size several times over. This differs from the paired difference of §1, which says that the endpoint and hazard readings differ; here the four that agree could have disagreed. Priced in replications at the endpoint's own decay (a log-decay of 1.0513), power 0.80 takes 39 paired seed arguments on the endpoint gap and 13 on the takeoff hazard (39.4 and 13.4 before rounding), an exchange rate of 2.94×; the $\sqrt{n}$ law holds to a factor of 1.2 over the sample sizes we tried.

Other intervals. Wilson and Jeffreys intervals on every count the main text prints change no printed exclusion of zero or one. Neither does a Beta–Bernoulli interval that keeps the seed argument as the unit (Bowyer et al., 2025, §3.1), which we computed on the six endpoint gaps and the ten factors of Table 29 with our own choices of Dirichlet weights over seed arguments and a Jeffreys posterior for each run. Pooled over evaluation episodes, the same interval is eight to eighteen times narrower and would exclude one for three of those factors and zero for both terminal-reward gaps, but the 768 window episodes of a run are one draw of that run, not 768.

Two count statements stay point readings: the count behind the first prescription, five of twenty-four rows missed at five replications against none of sixteen at eight (Fisher's exact $p = 0.071$), and the terminal-reward method arm's 0.9054, whose seed-argument interval, $[0.747, 0.984]$, contains the 0.90 ceiling.

Table 32: The four readings that show no decay are resolved more precisely than the endpoint reading that shows one. The five readings of §1 as two-point factors from $H{=}30$ to $H{=}100$, with the 95% interval of the 5,000-replicate seed-argument bootstrap of §1 (the median readings on their defined replicates), the standard error of each log-decay over 200,000 replicates of the same bootstrap, and that error as the probability of detecting the endpoint's own $2.86\times$ decay under a normal approximation.

reading	factor	95% interval	standard error	detection probability
endpoint mean gap, as published	$2.86\times$	$[1.47, 7.90]$	0.42	0.70
takeoff hazard ratio	$1.18\times$	$[0.74, 1.91]$	0.24	0.99
fraction of the budget returned	$1.16\times$	$[0.72, 1.76]$	0.23	1.00
median time to first attainment	$1.05\times$	$[0.74, 1.47]$	0.18	1.00
median time to hold	$0.73\times$	$[0.44, 1.22]$	0.26	0.98

What this does not show. The counts by gap have one realisation per comparison row, so their fractions are not detection probabilities; the probabilities of Table 31 hold for the pool's two cells only; and the detection probabilities of Table 32 assume a symmetric band that a right-skewed statistic does not have, so they are standard errors in another unit and not a second finding.

I THE THREE PRESCRIPTIONS IN FULL

Of the four things a study of this kind can change, only more replications cost training runs; the other three are free if they are decided before the runs start (Table 33). Here we give the three prescriptions of §6.3 in full, with their price on this pool.

Setup. Table 33 prices each change in training runs, from quantities this paper has measured. Its first row states the precision of two readings of one pool under the null that the profile does not depend on the reading, not the cost of seeing an effect that published work reports, and its counts come from measured standard errors rather than an assumed $1/\sqrt{n}$ law (which holds here to within a factor of 1.2).

The correlation in the second row and the share s in the third, the part of run-to-run variance that the seed argument does not control, are one quantity read two ways, the number of §3, and not two findings. The third row holds under the two-component model that the registered replication prediction assumes (execution noise independent of the seed effect, one variance across seed arguments), under which one run per seed argument is optimal for every share and merely tied at a share of one; the rule does not depend on believing our number.

Two numbers to read first. The across-seed standard deviation of each arm and of the per-seed difference between them, read in one launch batch, fix the share s of a comparison, which is also what the pairing bought, and the smallest arm difference the study could have detected, $(t_{1-\alpha/2} + t_{\beta})\,\mathrm{sd}(Y_A - Y_B)/\sqrt{n}$. At three runs per arm with a per-seed contrast standard deviation of 0.08, an example input and not a measurement of this pool, that is 0.25, twenty-five points of success rate. One released command prints the table and both numbers; its worked example is this pool read at the count we ran and at the count published work reports.

1. Count n in runs, not in seed arguments, and report at eight replications, run in one launch batch. Eight replications cost about sixty per cent more compute than five ($2.7\times$ the compute of three, $8\times$ that of one) and are executable today; §6.3 and Appendix H give what they buy on this pool. Do not buy evaluation episodes instead of runs: here the evaluation layer contributes a binomial standard error of at most 0.031 per point, well below the run-to-run spread of §3. If eight is out of reach, report the three runs, the twelvefold range their interval carries (§6.1) and their power.

Whatever the count, re-execute one configuration per arm twice: two executions cannot tell a run-to-run scatter from an offset shared by first launches, three can, and they also show whether re-executing a seed argument moves the outcome as far as changing it does. One released file with no dependencies reads the two sets of logs, or four columns of a CSV, and prints the share and the intraclass correlation

Table 33: Only more replications cost training runs; naming the estimand, pairing the arms and splitting the budget are free if decided before the runs start. The four things a comparison of this kind can change, what each is worth on this pool, what it costs, and where the paper measures it.

what you can change	what it is worth	what it costs	where
name the estimand you read (the same runs, a different reading)	its horizon profile measured about three times as precisely in run-equivalents: 39 paired seed arguments for the endpoint gap against 13 for the takeoff hazard, at power 0.80 on this pool	no training run; the evaluation trajectories, at a grid fine enough to resolve the event	§5
pair the two arms on the same seed arguments	$1 - 2\,\mathrm{Cov}(Y_A, Y_B)/(\mathrm{Var}(Y_A) + \mathrm{Var}(Y_B))$ times the runs: correlations from -0.39 to $+0.78$ here, four of six intervals containing zero	nothing, if decided before running	§2
split a budget of R runs between more runs and longer runs	a run repeated m times costs $m(1-s)+s$ per unit of precision; the diagnostic pair costs $1 + 2(1-s)/R$, at most 1.05 at $R = 8$ here	nothing, if decided before running	§6.3
run more replications	$\sqrt{n}$	linear	§6.1

with their intervals, what the split costs, and what a smaller study of the same pool would have published.

2. Name the threshold that defines the outcome, and report across its admissible range. A success threshold is a reporting convention, not a measurement. For a new environment, fix θ before training at the midpoint between the untrained policy's success rate and the level the strongest published arm reports, and print the sweep beside it, as Appendix E.4 does for this pool.

3. Write the domain of validity next to every registered assertion, and check it when a cell is added. It is the line we were missing all three times (Appendix M). Its price is a field and a check, not compute: a `domain:` field on the registered line naming the cells, arms and parameter ranges the assertion covers, and one comparison of that entry against a new cell's argument vector before the cell is launched. It needs two companion lines: where a registration names the differences two runs are permitted to have, that set must contain the factor under test, or the text is unsatisfiable by construction for one of the arms; and every guard a scorer applies to its inputs must run before it prints any reading.

What to report. Whatever a study concludes, it should report six things, of which only the third costs runs: (1) the estimand, the quantity the comparison is about, and the budget rule it is read under, in the sentence that states the result; (2) both readings of the same runs, the endpoint at a fixed update count and a time-to-threshold reading such as the takeoff hazard; (3) whether three executions of one seed argument per arm replay to the bit, which costs four runs, decides what the interval is over, and tells a low pair from a launch-batch effect; (4) n counted in runs, not in seed arguments, and the number of launch batches they were run in; (5) the width of the evaluation window the reported number is averaged over, and the number of episodes behind each point; (6) the reward regime, since a result under a dense shaped reward is not a result under a terminal one.

What this does not show. The prices are this pool's, on one environment family, and the per-seed standard deviation of 0.08 is an example input. The share reads high when the seed argument does different things in the two arms, which is the direction we argue for, so we do not offer it as evidence; telling that case apart from a seed that does nothing is what the two extra runs of the first prescription are for.

J THE ESTIMATORS, THE THREE PERTURBATIONS AND FOUR PRE-SPECIFIED READINGS

The pool's two estimators and a third run behind one training loop. The third estimator and a second model family reproduce the pool's dependence of the horizon reading on the estimand and the budget rule, with the arm under test at the ceiling, and the further checkpoint stops at the ceiling gate. Of four readings specified before they were computed, none is scored; the one the main text uses is that in every family the endpoint decay separates from the hazard reading on the risk difference and the log success ratio, and on neither the logit nor the log failure ratio.

Setup. The estimators were registered before the pool ran, and each perturbation and group-size cell before it ran. The runs of the first two perturbations used the stack revision of the same-launch comparison (Appendix C), each argument vector being the pool's own for the same estimator with the one change rewritten. The four further readings were defined in writing after the scored tables and before any was computed, then printed from one completed run of an analysis script. They are pre-specified exploratory readings: no prediction is scored on them and no outcome moves. No multiplicity correction is applied to any of the 107 intervals of that run, and none is claimed; a reader who wants one has the count.

J.1 THE ESTIMATORS

Every credit-assignment method on KnobChain is a turn-advantage estimator behind one fixed training loop, so that cells differ only in the estimator and the knobs. The baseline, `grpo-turn`, broadcasts the episode-level group-relative advantage uniformly to every turn, so its *turn-level* credit content is zero by construction; every paired contrast on KnobChain is taken against it, while the public-stack and second-library comparisons contrast their own released arms. The method arm, `gigpo`, the published reference method's estimator, adds to the episode layer a micro layer that regroups turns *across* trajectories by exact observation hash, a state anchor. No grouping key sees the true room index.

The third arm, `gagpo`, has no learned critic: each turn's advantage is a step-level temporal-difference estimate built from value proxies taken from group statistics. The loop hosts five further estimators that enter no reported comparison: a coarsened-key and a similarity-keyed variant of the method's, two tree-based estimators and a reference-model temporal-difference estimator.

J.2 HOLDING, AND THE THREE PERTURBATIONS

Section 5 reports that at five times the registered budget every run of both arms reaches $\theta = 0.90$ at least once (the fraction is 1.000 in both arms), while all sixteen method runs *stay* above it and eight of sixteen baseline runs do. A registered prediction of asymptote parity is scored a hit on this result, which falls at exactly its boundary, so that hit is to be discounted; its companion, a point prediction, missed. A registered rescue prediction tests the reaching side at $H{=}30$ and misses: warm-starting sixteen baseline runs that had not taken off rescues thirteen of sixteen in either arm (0.8125).

Three perturbations change one thing at a time (Table 34): the estimator, by adding `gagpo` as a third arm; the model family, to Llama-3.1-8B-Instruct; and the checkpoint within the pool's family, to Qwen2.5-Coder-7B-Instruct, which as registered is a same-family specialisation of Qwen2.5, one further checkpoint and not a second model family. The third arm has 32 seed arguments, one execution per estimator and 96 runs over three depths, and all five of its registered predictions hit: its endpoint gap decays; the endpoint convention reads more decay than the event-time convention, and the proportional budget more than the fixed count; most of the decay is headroom compression rather than a change of efficacy with depth; and the reading that does *not* decay does not decay here either.

On the second model both convention predictions hit. One of its 128 runs was lost to a memory failure; the rule registered in advance never re-runs a lost run, so we read the cell under both extreme imputations, the floor first in Table 34 and the ceiling after the semicolon, and score a prediction only if both agree. In both perturbations the arm under test stands at or above 0.90 at both depths (the third arm at 0.9235 and 0.9686), so these hits check the mechanism at the ceiling; within that scope the dependence of the horizon reading on the estimand and the budget rule is a property neither of the pair of estimators this paper compares nor of the model family.

On the checkpoint the registered ceiling gate fired, so both predictions are not evaluable and nothing is scored. What it leaves is a description: at $H{=}30$ its arms take off in 32 of 32 runs and in 7 of 32, at $H{=}100$ every run takes off in both arms, and one convention says the arms have converged (0.033 apart) while the other says they are still 0.336 apart.

Table 34: The added estimator and the second model family read the two conventions as the pool does; the checkpoint stops at the ceiling gate. Log-decays from $H{=}30$ to $H{=}100$ with 95% seed-argument bootstrap intervals, and endpoint arm means as method / baseline.

reading	pool	third arm	second model	checkpoint
endpoint log-decay	factor $2.86\times$ $[1.47, 7.90]$	0.7925 $[0.2598, 1.4683]$, a factor of 2.209		
endpoint minus event-time log-decay (registered)	0.8831	0.6343 $[0.1791, 1.2454]$	0.6505 $[0.1157, 1.4129]$; 0.7548 $[0.2034, 1.5616]$	not evaluable
proportional minus fixed-count log-decay (registered)	1.0865	1.2311 $[0.7307, 1.8604]$	0.7117 $[0.0043, 1.4357]$; 0.8148 $[0.0117, 1.6396]$	not evaluable
headroom share, against a null of one half	0.770 $[0.585, 1.139]$	1.0219 $[0.8663, 1.4905]$		
event-time log-decay	factor $1.18\times$ $[0.74, 1.91]$	0.1582 $[-0.2756, 0.6059]$		
arm means at $H{=}30$			0.9060 / 0.5723	0.8620 / 0.4287
arm means at $H{=}100$			0.9256 / 0.7478; baseline 0.7791	0.9493 / 0.9161, at or above the ceiling of 0.900
arm means at $H{=}100$ at published work's fixed count				0.3617 / 0.0259

Raising the group size at $H{=}30$ from the pool's 16 to 32 and 64, every other argument cloned byte for byte, keeps the seed-paired endpoint gap positive at a fixed update count in 15 of 16 and 16 of 16 seed arguments, and at a fixed compute in 14 of 16 in both cells; both registered predictions hit.

Five further models. We also ran Qwen2.5 at 1.5B, 3B and 14B, Qwen2.5-Coder-7B and Llama-3.1-8B in the launch batches that added models (not the perturbation cells above). The 1.5B and 3B models sit at a task floor where a pre-registered degeneracy guard returns *not evaluable*, so the scale claim is confined to the other three, where the registered scale predictions hit. The seed-paired endpoint gap is $+0.070$ on the 14B cell (arm means 0.953 and 0.883, against a gap of 0.318 at 7B; sign test $15/16$, exact $p = 0.0005$), $+0.417$ on the Coder-7B cell and $+0.270$ on the Llama-8B cell, and in each the baseline arm's variance exceeds the method's.

At $\theta = 0.90$ the attainment difference is $+0.4375$ at 14B against $+0.6562$ at 7B, inside the 7B seed-bootstrap 95% band $[0.4062, 0.8750]$ by 0.031: by the registered containment test it is not distinguishable from the 7B value, nor, at sixteen pairs, from a halving. The 8B cell of the other family passes both of its predictions.

J.3 The decay on five scales, in three families

We expressed the change of the endpoint difference from the shallow to the deep cell on five scales and asked, per scale, whether it separates from the takeoff-hazard reading of the same cells, that is, whether the interval of their paired log-difference D_S excludes zero (Table 35). Reading form (i) is the ratio form of the table; reading form (ii) replaces the log of the ratio of a scale's shallow and deep values by their difference and is defined on the log scales only. Profile A reads the third arm against the baseline, profile P the pool's method arm against its baseline.

The risk-difference entries recompute results printed earlier, the pool's in Table 26 and the other two families' scored ones in Table 34, on this analysis's own bootstrap draws, and all intervals rest on 5000 of 5000 defined replicates except the second model's ceiling imputation (4999 of 5000). A third profile, B (the method arm against the third arm), is a pre-specified secondary reading: at every scale its two-point factor is defined in 0 of 5000 replicates, so no D_S is printed.

The registration forbids reading the pool, the third arm and the second model as agreeing or disagreeing as a group, so each family is stated on its own. On the third arm (profile A), two of five scales separate in reading form (i), the risk difference and the log success ratio, and none of four in reading form (ii). That is the fewer-than-half branch, whose sentence the registration fixed in advance: on

this cell family the endpoint separation is a statement about headroom at the shallow depth and not about a horizon-dependent estimand.

On the second model the complementary log-log separates in reading form (i) under the ceiling imputation only, so the count is reported as BOUNDED, the range 2 to 3 of 5, and two, under the floor, takes profile A's sentence; in reading form (ii) none of the four separates under either imputation. On the pool (profile P) three of five separate in reading form (i), the complementary log-log among them, and two of four in reading form (ii). In every family the interval of $\log F_{HR}$, the logarithm of the two-point hazard factor, contains zero. The registration names no branch for a count of two, but its sentence for one and for zero is the same, so no permitted sentence depends on the missing label.

Table 35: In every family the risk difference and the log success ratio separate from the hazard reading, and the logit and the log failure ratio do not. Reading form (i): D_S, the paired log-difference of each scale's endpoint reading and the hazard reading, bold where its 95% bootstrap interval excludes zero; the second model's lost run imputed at the floor and at the ceiling. RD: risk difference; LOGIT: logit; LFR: log failure ratio; LSR: log success ratio; CLL: complementary log-log.

scale	A: third arm	P: the pool	second model, floor	second model, ceiling
RD	0.6343 $[\mathbf{0.1791}, \mathbf{1.2452}]$	0.8831 $[\mathbf{0.2790}, \mathbf{1.8114}]$	0.6505 $[\mathbf{0.1157}, \mathbf{1.4096}]$	0.7548 $[\mathbf{0.2037}, \mathbf{1.5593}]$
LOGIT	-0.0569 $[-0.4490, 0.4166]$	0.4173 $[-0.0144, 1.1235]$	0.3406 $[-0.1891, 0.9757]$	0.3803 $[-0.1773, 1.0594]$
LFR	-0.2052 $[-0.6316, 0.2909]$	0.2798 $[-0.1491, 0.9603]$	0.2368 $[-0.3141, 0.8518]$	0.2625 $[-0.3168, 0.9356]$
LSR	0.8200 $[\mathbf{0.3184}, \mathbf{1.4686}]$	1.0600 $[\mathbf{0.3952}, \mathbf{2.0357}]$	0.7881 $[\mathbf{0.1587}, \mathbf{1.6405}]$	0.9122 $[\mathbf{0.2932}, \mathbf{1.7809}]$
CLL	0.2471 $[-0.1284, 0.7357]$	0.6414 $[\mathbf{0.1502}, \mathbf{1.4220}]$	0.4993 $[-0.0088, 1.1673]$	0.5663 $[\mathbf{0.0338}, \mathbf{1.2899}]$

J.4 THREE FURTHER PRE-SPECIFIED READINGS

The knob at $\lambda = 1.00$. Anchor coverage thins with depth in the pool, whose cells never run $\lambda = 1.00$, where it does not thin. To ask whether coverage accounts for the pool's separation, we read four other ladders that run the knob at $\lambda = 1.00$ (a ladder is a set of cells of a common design at different depths, its rungs). The registration writes one sentence for this outcome and we write no other: the $\lambda = 1.00$ ladders as executed cannot resolve the question, at five shared seed arguments on the two ladders that have two depths, with takeoff events of five against two, one against one and none against none; one ladder's evaluation count differs from the pool's and the fourth has one depth. The coverage sentence of the apparatus appendix stands unchanged.

What would be needed, as the registration asks us to say, is a ladder with both rungs on one stack, under the pool's proportional budget and evaluation count, with enough seed arguments that both arms meet the event gate (a minimum count of takeoff events per arm) at the shallow depth, where the same-stack ladder's baseline arm takes off in two of its five. None of the four ladders we ran is such a ladder; the registered off-ceiling cell of Appendix E.2 answers the question this reading asked.

The published per-type numbers against our three-run band. Every per-type gap the GiGPO paper publishes (Feng et al., 2025, Table 1) lies inside the band of gaps that three-run studies drawn from our runs would publish when a study is a set of three runs, over the endpoint window and at the final evaluation alike; when a study is a set of three seed arguments, three of six do at the window.

The deterministic-kernel baseline shift, and the arm contrast. We compared the baseline arm with deterministic kernels switched on against the same arm on the ordinary stack, over sixteen pairs at H=30. The baseline shift is -0.1025 $[-0.3002, 0.0851]$, and the effect on the arm contrast is $+0.1172$ $[-0.0887, 0.3360]$. Neither is resolved, and an interval that contains zero is not evidence that the deterministic path is neutral.

What this does not show. The first two perturbations check the mechanism at the ceiling, not off it. The four pre-specified readings move no outcome: the $\lambda = 1.00$ ladders cannot say whether anchor coverage accounts for the pool's separation, because no ladder we ran meets the event gate at both depths; the comparison with the published gaps compares the widths of two stacks and two evaluation sets and audits nothing; and the deterministic-kernel intervals contain zero.

K THE REPLICATION-REPORTING SURVEY

Most recent papers on reinforcement-learning post-training of large language models do not say how many training runs stand behind their main results, and a second, blind coding finds the same rate. None of the eleven credit-assignment papers among them ties its training budget to the horizon.

Setup. We fixed the protocol before collecting the sample and did not adjust it after the rates were read. A paper entered if its first arXiv submission fell between 2025-09-01 and 2026-08-28, if it was in English, if its subject was reinforcement-learning post-training of a large language model, and if it carried at least one experiment its authors had trained themselves; surveys, purely theoretical papers and evaluation-only papers were excluded. Papers entered in search-ranking order, never selected on how well they reported, and each was read in full text, appendices and hyperparameter tables included; 78 entered.

A paper is coded as reporting a count only if it says how many independent training runs each configuration of its main result received: repetition at the evaluation layer (pass@k, avg@k, resampling one checkpoint, an inference seed), the group size of a group-relative estimator and a bare seed value do not count, and a number that does not distinguish training from evaluation is coded as partial. Each row of the coding carries the arXiv identifier and a verbatim evidence sentence. Both codings were made by language-model assistants (the ones the statement on generative AI describes), working under the authors' direction from each paper's full text, the second blind to the first's per-paper values.

Readings. Section 1 reports that 71.8% of the 78 papers state no per-configuration replication count for their main results. Among those that do, the modal count is 3 (median 3, max 5), and only 2 of 78 state what the seed controls. If papers that report a replication count for only part of their experiments, or that name seeds without a number, are also counted as not meeting the bar, the under-reporting rate is 80.8%. In their main results table, 62.8% carry no dispersion of any kind, and one of the 78 reports a confidence interval across runs. The non-reporting rate moves with the coding, between 62.8% and 83.3%, and the modal count is 3 under every coding.

Double coding. The two codings agree on the three-level code (a count for the main results; a count for part of them, or seeds without a number; none) in 73 of 78 papers (Cohen's $\kappa = 0.855$, 0.86 rounded), with $\kappa = 0.87$ on none against any and 0.91 on a stated count, whose value agrees in all 13 papers where both find one, and less well on whether the main table carries any dispersion (66 of 78, $\kappa = 0.66$) and on whether a paper says what its seed controls (75 of 78, $\kappa = 0.38$).

The credit-assignment papers. Eleven of the 78 papers propose or compare turn- or step-level against trajectory-level credit assignment for multi-turn agent RL, the kind of comparison this paper makes, and Table 36 names them; the second, blind coding recorded this property and the training budget each paper fixes, the first recorded neither. Ten fix a count of steps, iterations or epochs, and one states no budget, so none ties its budget to the horizon. The two published comparisons that stratify by horizon fix it as well: SPA-RL (Wang et al., 2025) in epochs, with its run count unstated, and ReBel (Tang et al., 2026) in iterations, unequal between the arms, over three seeds.

What this does not show. The sample was drawn in search-ranking order rather than at random, so the rates describe the retrievable, highly ranked papers of this kind; they are not an estimate for the published literature as a whole. The rates err in one direction only: a paper that states its replication count only in a camera-ready version or a code repository is scored here as not stating it, so the non-reporting rate is if anything an overestimate.

The codebook records whether a count is stated for the main results, not how the runs are combined or which results they cover, so it does not separate the best of N runs, the mean of N and a replicated main table beside single-run ablations; in their evidence sentences, all eleven papers that the second coding marks as stating a count above one report a mean or a mean with a spread, and none reports a best of N.

Table 36: None of the eleven credit-assignment papers ties its training budget to the horizon. The eleven papers of the 78 that the second coding marks as proposing or comparing turn- or step-level against trajectory-level credit assignment for multi-turn agent RL, with that coding's replication count for the main results, the kind of dispersion it found beside the main comparison, and the training budget the paper fixes.

arXiv	paper	runs per configuration	dispersion	budget
2509.21240	Tree-GRPO	none stated	none	fixed: training steps
2512.17008	Turn-PPO	none stated	none	fixed: training steps
2605.02178	T2PO	3	across runs	fixed: epochs, per task
2605.06200	A2TGPO	none stated	none	fixed: steps
2605.13217	GAGPO	3	across runs	fixed: steps
2606.12384	APPO	none stated	none	fixed: epochs, per task
2606.25852	SCPO	3	across runs	fixed: iterations
2607.04242	ProGPO	3, for part of the results	unclear	not stated
2607.06223	IGRPO	none stated	none	fixed: iterations
2607.07435	RLVP	5	across runs	fixed: iterations, per domain
2608.19803	MileGPO	none stated	across evaluations	fixed: steps

L THE LITERATURE SIMULATION

At three runs per arm, the modal count of the survey, a literature that runs the standard study on our pool publishes a distribution over conclusions rather than noise around the pool's value. More runs make the published horizon direction reliable under the proportional budget, but under a fixed count of updates it stays a coin toss at every n up to sixteen.

Setup. We ran the standard study design of published work ten thousand times on our pool, with the full pool's values as the ground truth; each simulated study stands for a parallel literature. A study draws n runs per arm on the same seed arguments in both arms, as a typical paper does, compares the arms' mean success at the end of training (the endpoint) at all six horizons under the horizon-proportional budget, and publishes what those numbers say. We then read the same studies under a fixed count of updates and under the takeoff hazard, the rate at which runs cross a fixed success level.

L.1 TEN THOUSAND PARALLEL LITERATURES

The miss rates of Appendix H describe single comparisons; here is what they do to a *literature*. At $n = 3$, the modal replication count in the survey of Appendix K when one is reported at all:

- 16.4% of the parallel literatures publish a horizon trend opposite to the full-pool endpoint reading: where the full pool reads the gain as decaying with the horizon, they publish it *growing*. Separately, 4.4% get the sign of the method itself wrong at the shallowest horizon.
- Among studies whose endpoint gaps at the shallowest and the deepest horizon are both positive, the published decay factor (the shallow gap over the deep gap) spans $0.55\times$ to $34.2\times$ (5th to 95th percentile; median $2.90\times$, full pool $2.86\times$); a three-run study on this pool prints one draw from this distribution.
- 53% of the studies call the shallow-horizon gain significant with a Welch t-test as typically scripted, where an exact sign test cannot go below $p = 2^{1-n} = 0.25$; the two tests disagree about half the time at this n. The normal-approximation form rejects 18.5% of the time at $n = 3$ under a permutation of the arm labels, which carry the effect, so that rate is not a size; under the two nulls built by splitting one arm against itself it rejects 12.6% and 7.2% against a nominal five, and the exact form 4.5%, 4.2% and 1.9% (in both forms the lowest is the split of the method arm, near the ceiling). The 53% is thus a mixture of a real effect and a test that is anti-conservative wherever there is variance to inflate; it is not a count of false positives.
- 90.5% of the studies place the apparent crossing depth, where a study's fitted gap would have halved, inside the grid (Figure 10), at a median depth of 50.2 with a 95% band of $[35.6,\ 87.6]$. A constructed pool with *zero* horizon-dependent effect (98.0% in-grid crossings, median 52.3) and the full n=32 measurement (54.6) put the crossing at the same depth, since all three share the measured baseline curve.

Reading the best checkpoint over the trajectory instead of the endpoint window does not rescue the single-run convention: at $n = 1$, 9.0% of simulated studies still publish the wrong sign at $H{=}30$, and 18.0% still publish the gain as growing with the horizon.

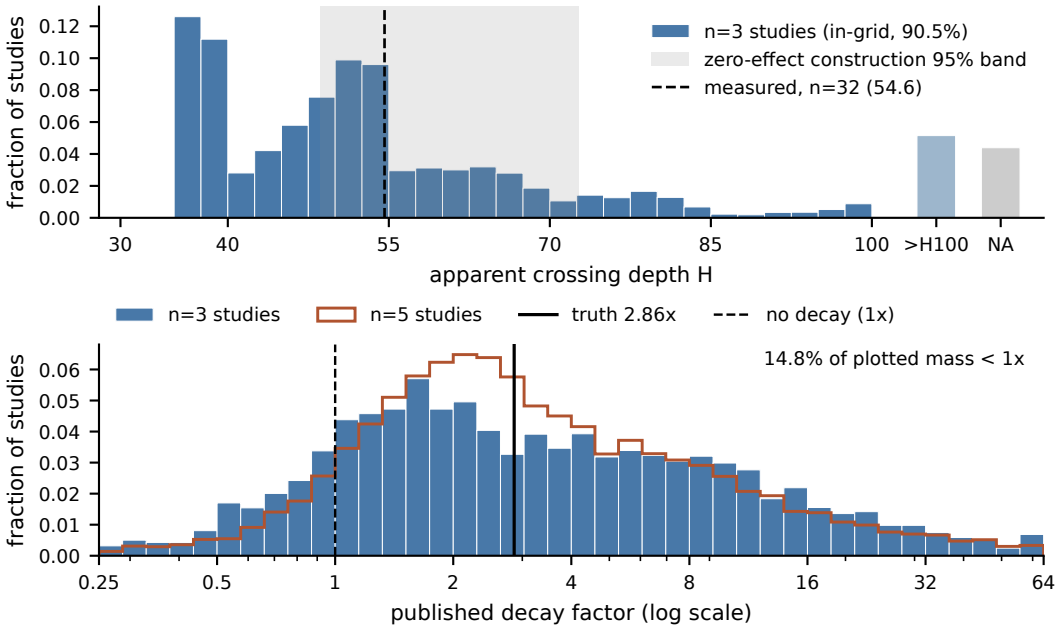


Figure 10: At the modal replication count the outcome is a distribution over conclusions, not noise around one. *Bottom:* the published decay factor against a truth of $2.86\times$, among the studies whose two endpoint gaps are positive (14.8% of them below $1\times$, against a reversal rate of 16.4% over all studies). *Top:* the apparent crossing depth of each study; 90.5% fall inside the grid, and the *zero-effect band* is where studies with no horizon-dependent effect put it.

L.2 The same study under three conventions

Table 37 reads every simulated study under the two budget rules of §4.1 and under the takeoff hazard, on the same draw of runs, at every n from one to thirty-two.

In the table, *sign* is a wrong published sign at $H{=}30$ and *growth* a published gap at $H{=}100$ at or above the one at $H{=}30$; *factor* is the median and $[q_{05}, q_{95}]$ of the $H{=}30$-to-$H{=}100$ ratio among the studies that pass the reading's gate (both gaps positive; both ratios defined), *pass* the share that passes, *HR decay* a hazard ratio that falls with depth and *NA* one that is undefined; the last two columns count decay under both budget rules and under the proportional rule only. Reading (b) takes $H{=}100$ at its step-150 record, the last before 165, and reading (c) is the O/E log-rank form.

Under the fixed count the published horizon direction is a coin toss at every n up to sixteen, because the pool's own fixed-count factor is $0.97\times$, while under the proportional budget the same studies publish decay more often as n grows (Figure 11). Read as the takeoff hazard, 0.62 of the three-run studies that can form a ratio publish a decay, against a pool value of $1.18\times$. The exact paired sign test reaches $p < 0.05$ in no study below $n{=}6$ and from $n{=}6$ to $n{=}8$ only in the unanimous ones, while the normal-approximation Welch t of typical scripts declares significance in more than half of the studies from $n{=}3$ on. At $n{=}3$ the endpoint columns read 0.171 and 0.041, the published rates of Appendix L.1 within Monte-Carlo error.

What this does not show. The simulation measures how often a small study recovers the full-pool *endpoint* reading of §4, and nothing more: the $2.86\times$ it calls the truth is that estimand's value over all thirty-two runs per arm, not a statement about the method. Its truth is itself an estimate: at the full pool the direction, the $H{=}100$ endpoint gap minus the $H{=}30$ one, is -0.207, 95% $[-0.339, -0.075]$.

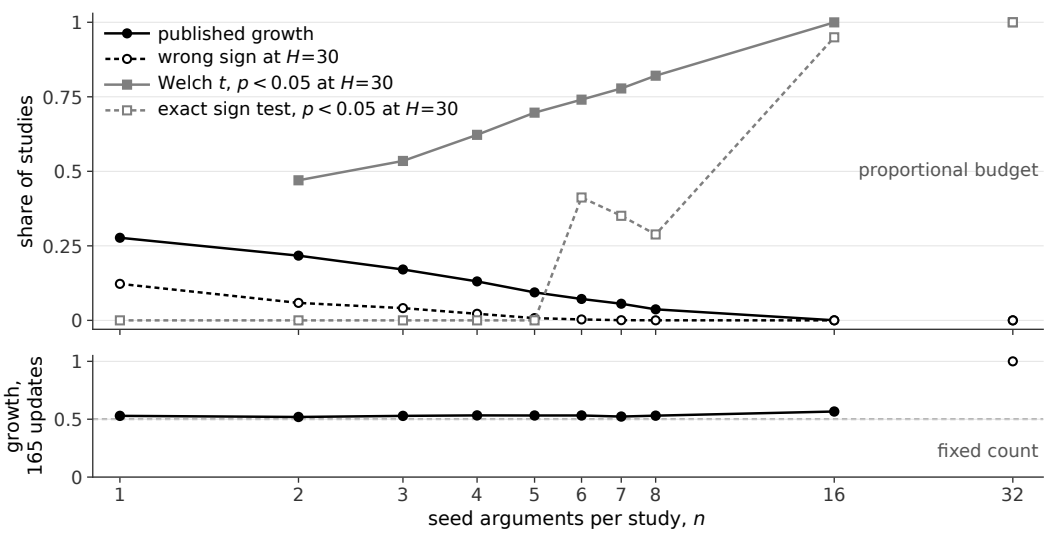


Figure 11: More seed arguments make the published horizon direction reliable under the proportional budget, but under a fixed count of 165 updates it stays a coin toss at every n up to sixteen. Upper panel, proportional budget: the share of studies with the wrong sign at H=30, with growth, and with $p < 0.05$ at H=30 under the exact sign test and Welch's t; lower panel, the growth share at 165 updates; open markers, the pool itself (Table 37).

Table 37: Under the fixed count about half of the published studies show growth at every n below the full pool; under the proportional budget that share falls towards none as n rises. Ten thousand studies per row, each on n seed arguments shared by both arms, read (a) as the endpoint gap at the registered budget, (b) at 165 updates for both depths and (c) as the takeoff hazard ratio at $\theta = 0.70$; the last row is the pool itself.

	(a) proportional budget, endpoint				(b) fixed 165 updates			(c) takeoff hazard				$p < 0.05$ at H=30		one study, two rules	
n	sign	growth	factor	pass	growth	factor	pass	HR decay	NA	factor		Welch	sign test	both decay	prop. only
1	0.122	0.277	3.63 [0.27, 21.9]	0.626	0.529	0.70 [0.14, 25.9]	0.688	–	1.000	–		–	0.000	0.377	0.346
2	0.059	0.217	3.27 [0.42, 36.3]	0.751	0.519	0.92 [0.19, 8.7]	0.870	0.510	0.204	1.05 [0.19, 3.1]		0.470	0.000	0.416	0.367
3	0.041	0.171	2.90 [0.52, 32.9]	0.824	0.529	0.95 [0.26, 6.1]	0.934	0.619	0.071	1.24 [0.32, 3.0]		0.535	0.000	0.428	0.401
4	0.022	0.131	2.79 [0.65, 30.0]	0.885	0.533	0.96 [0.27, 3.9]	0.968	0.641	0.025	1.27 [0.39, 3.1]		0.622	0.000	0.437	0.432
5	0.008	0.094	2.82 [0.75, 27.2]	0.926	0.532	0.96 [0.31, 3.2]	0.988	0.659	0.010	1.28 [0.45, 3.0]		0.697	0.000	0.447	0.459
6	0.003	0.072	2.89 [0.86, 23.2]	0.954	0.532	0.96 [0.35, 2.6]	0.996	0.668	0.002	1.25 [0.51, 2.8]		0.740	0.412	0.455	0.473
7	0.001	0.056	2.90 [0.95, 20.3]	0.969	0.523	0.97 [0.39, 2.4]	0.999	0.676	0.000	1.27 [0.55, 2.7]		0.778	0.351	0.468	0.477
8	0.000	0.037	2.93 [1.08, 17.3]	0.980	0.531	0.97 [0.44, 2.2]	1.000	0.686	0.000	1.24 [0.58, 2.4]		0.821	0.288	0.464	0.499
16	0.000	0.000	2.83 [1.60, 6.8]	1.000	0.567	0.96 [0.63, 1.5]	1.000	0.747	0.000	1.18 [0.79, 1.8]		1.000	0.950	0.433	0.566
32	0.000	0.000	2.86 [2.86, 2.9]	1.000	1.000	0.97 [0.97, 1.0]	1.000	1.000	0.000	1.18 [1.18, 1.2]		1.000	1.000	0.000	1.000

Every subsampling simulation of this paper draws seed arguments without replacement (finite-population factors 0.774 at $n = 8$ and 0.516 at $n = 16$), so the n=16 row, which draws half the pool, is optimistic: its exact sign test's rate is 0.950, against 0.870 with replacement. Recomputing the $n = 3$ and $n = 1$ rates at $B = 5000$ moves each by less than 0.015.

M THE REGISTER: RULE, TALLY, THE ELEVEN MISSES AND FOUR FAILURES

The register. Every miss is printed as it was scored, and no failure of the machinery led to a rescored outcome. The register is a single append-only text file whose lines are hash-stamped. A registered prediction, a sentence with its scoring rule, is appended before its data exist, with a machine assertion that the cells it covers hold zero runs; a scoring script later scores it against its registered text verbatim and appends the outcome, no line is ever edited, and every disclosure was stamped before its repair. A prediction scores a hit, a miss, or neither, an outcome its registration named as neither: not evaluable at a floor or a ceiling, inconclusive, a bound and the like. One scored prediction is one scoring line that quotes the stamp of the prediction it scores, a line that scores a prediction in parts counting once per part; three instrument assertions are held out of every count.

The tally. Table 38 gives the tally at three hundred and fifty register lines, the state at which §6.3 reads it; no bare hit rate is printed, since one rate would fold the twenty-four outcomes of neither kind into hits or misses. Two scored predictions are read nowhere in the main text. The control cell of the deterministic stack at $H{=}30$ hit: with the determinism flag unset, its baseline arm mean over eight seed arguments, 0.6017, lies inside the registered central 90% interval of the reference distribution over the pool, $[0.4359, 0.7178]$ (Appendix B). The shallowest depth ($H{=}30$) of a registered six-depth ladder at a harder difficulty tier was not evaluable at its floor gate, its baseline arm mean of 0.3132 lying below the registered 0.350, and the ladder's remaining depths were not run.

Table 38: Of seventy-four scored predictions, 39 hit, 11 missed and 24 returned an outcome of another kind. The scored predictions by the outcome each registration named for itself (two spellings of bound merged), read from the register alone as it stood at three hundred and fifty lines, the state at which the tally of §6.3 is read; three instrument assertions are held out.

outcome, as the registration named it	predictions
hit	39
miss	11
not evaluable	2
not evaluable at a ceiling	2
not evaluable at a floor	1
not evaluable for want of events	2
not evaluable by the instrument	1
ambiguous by window	1
early randomness	1
inconclusive	1
scored per cell	1
bound	3
grows	1
none	3
replays across hosts	1
robust	1
selected	1
separates	1
unresolved, both arms	1
total scored	74
held out (instrument assertions)	3

The matrix and its outcome vocabulary. The matrix of 45 configurations (cells) around the main pool's ladder of depths, fixed before any of its rows was scored, holds about seventy comparison rows, each a live arm against the degenerate baseline in the same cell. Three gates were fixed first: a *floor*, baseline final success of at least 0.10; a *ceiling*, both arm means at most 0.90, a readability rule; and a *precision* gate, a half-width of at most 4.00 percentage points for the paired gain's 95% confidence interval. Each row reads *consistent*, *indistinguishable*, *contradicted*, or *unreadable* with the gate that failed, a statement about readability, never about the effect.

The eleven scored against us. Two of the misses of Table 39 changed a sentence of this paper. Under a purely terminal reward at the registered budget, the sign prediction missed in a cell that had not learned, so "dense shaped reward" stays in the statement of the effect in the abstract and §1; and the failure of the variance proxy removed the quantitative dose–response from the paper. The two public-stack switch predictions changed no sentence: the first prescription stands, and no fourth one is added. Two further predictions are registered and deferred by design (Appendix N); we expected both to fail, and said so in dated analysis files before the runs that score them existed.

Four failures of the machinery. Each was caught by the machine or disclosed, and no outcome was rescored. A variance proxy came out with the sign opposite to its registration: a run's whole-trajectory coefficient of variation in gradient norm was registered to fall as its probability of taking off rises, Spearman at most -0.60, and read Spearman $+0.894$; the quantitative dose–response is removed from the paper, and the mechanism is reported as a candidate, undetermined and not refuted. Three times a check written for one domain, and correct there, was carried into a cell it had not been written to cover; our own machinery caught each drift, each check was widened, and the missing line, a domain entry on the registered line, is the third prescription of §6.3.

A stop of every running job was stamped after its runs had finished: the stop killed eighty-six runs in flight, and the authorisation to score the complete experiment, half of the two-depth field, was

Table 39: Every prediction that missed was registered before the runs that score it existed, and none was rescored. The eleven, in the order of their scoring: what each asserted, what was read, and where the full account stands ("archive": the full record).

prediction	asserted, and read	full account
P-PHI-6-POINT	the point prediction beside the asymptote-parity prediction: at five times the budget at least 0.688 of the baseline arm's runs (11 of 16) end at or above $\theta = 0.90$; read 8 of 16, which scores the parity prediction itself as a hit at exactly its boundary	Appendix J.2
P-VAR-1	the variance proxy: a Spearman correlation of at most -0.60 between a run's whole-trajectory coefficient of variation in gradient norm and its probability of taking off; read $+0.894$	below
P-PHI-4(c)	the common-mode control of the knob dose–response prediction: along the temperature ladder the variance ratio and the low-tail gap show no monotone direction; the control experiment fails, and the registered consequence discounts the two experiments that passed	archive
P-REPL-C-1	the eight-seed, four-execution experiment: an intraclass correlation of at most 0.30 in both arms; read 0.0000 and 0.3327	Appendix B
P-RESCUE-1	warm-started from untaken-off baselines at $H{=}30$, the method arm rescues a strictly larger fraction; read 13 of 16 in either arm (0.8125)	Appendix J.2
P-SPARSE-1	under a purely terminal reward at $H{=}30$, at the registered budget, the seed-paired endpoint gap is positive in at least 13 of 16; read 6 of 16, a miss in a cell that had not learned (arm means 0.155 and 0.171)	Appendix N
P-SPARSE-2	the same cell's reliability signature; read a baseline-over-method variance ratio of 0.528 and a low-tail gap of -0.0107 against 0.1315 in the high tail	Appendix N
P-REALSTACK-BI-1	the batch-invariant second stack lowers the between-execution variance by the registered factor; read an F ratio of 0.249 against the registered critical 3.787	Appendix N
P-REALSTACK-SW-2	the stricter switch set on the public stack lowers the re-execution variance by the registered factor; read $V(\text{default})/V(\text{stricter switch set}) = 1.957$ against the registered one-sided critical ratio 3.787 ($p = 0.1977$)	Appendix N
P-TRL-SRV-2	on the second library's served default the mean re-execution displacement reaches a registered 0.05; read 0.0453 $[0.0245, 0.0638]$ (dapo) and 0.0452 $[0.0265, 0.0631]$ (grpo)	archive
P-RS2-SEED-UNIT	on the public stack's sixteen-seed re-run the arms-pooled exact 95% interval of the intraclass correlation at the final evaluation excludes zero; read 0.1518 $[-0.199, 0.470]$, which contains zero	Appendix D.2

stamped two and a half days after that experiment finished writing; the machine certifies that no pre-committed instrument read the experiment in between, not that no author opened a run file by hand. Finally, no prediction registered before the public-stack experiments stated its power; for the headline replication prediction on the public stack a miss probability between 0.15 and 0.33 was computed from published inputs only and published before its data existed, and that prediction has since hit.

Expectations we wrote down before reading. Beside the register, a separate ledger carries the expectations we wrote down about our own apparatus before each reading, each with a probability and, after the reading, its outcome; its scores are in the full record. Fourteen further expectations were registered blind on the ledger alone and read the next day: five yes, five no and four not evaluable.

How the build checks this text. Two released verification programs check every numeric claim of the main text against the receipt it comes from, and a released script checks at build time every count this paper prints about its register against the register as read, at 350 lines for the tally of §6.3, Table 38 and Appendix N's status sentences.

N Registered cells scored after the pool or still open

The outcome of a prediction deferred by design is not assumed. Besides the first master-copy block, fifteen predictions were scored after the pool had been read, each with its pre-outcome text and the outcome appended; Table 40 lists the cells other sections point to; the full record lists every open cell, among them a mixed-horizon corridor cell and a budget-rule cell on the public stack that hold no runs, and a Llama-8B cell at $H{=}70$ stopped after an out-of-memory failure, none of which is read.

Registered cells read in other appendices. Under a purely terminal reward at $H{=}30$, at the registered budget, the sign and reliability predictions miss and the share prediction hits (Appendix O). Of the switch cells on the public stack two hit and the re-execution variance ratio misses, as does

Table 40: No open cell is read in this paper. The registered cells that other sections point to, each registered before its scored runs existed, with its prediction and its outcome or state.

cell	design	registered prediction	outcome
the composition share, point version	deferred by design; a dated, unregistered note of ours expects it to miss, and dated analysis files the other two to fail	the point version; two further predictions	scored verbatim when the deferred experiment completes
the horizon joint, two predictions	Llama-3.1-8B-Instruct, a second model family, registered before its own cells ran and after the pool had been read	the mechanism on a second family, not an independent test of the joint	both hits, under both imputations of the one lost run (Appendix J.2)
the master-copy block at a stable rate	5×10^{-7}, chosen by a registered pilot (four of four runs stable); ninety-six runs	the execution component; the seed's share	ROBUST; BOUND (Appendix A.4)

the same ratio on a second, batch-invariant stack (Appendix D). On the second library's colocated configuration our registered prediction of a share of at least 0.50 is false as stated (Appendix B.6).

The ceiling. Of the properties of the instrument that bound the horizon reading (§8), the ceiling matters most. The method arm sits within 0.11 of a success rate of one at every depth, and its endpoint mean is above 0.90 in five of the six main-pool cells (Table 6), the matrix's readability gate (Appendix M), which governs the matrix's rows and not the depth ladder; but in five of six cells the method arm has little room left in which a horizon dependence could show, so the finding that the hazard is not resolvably different across depth bears on the effect and the ceiling jointly.

A further property bounds the replication reading from the other side. This stack serves its own rollouts at a fixed batch, without the batch-composition source of nondeterminism that a served stack carries, so we expect a served stack's nondeterminism share to be no lower than ours. The second library's served default does not bear this out on its grpo arm: it reads 0.69 in calibration, 0.501 in a later calibration run of 30 pairs and 0.490 in the registered cell, against this pool's lowest cell at 0.79 (Table 2). The mixed-horizon design of published work, one policy read per task type at a fixed count, follows neither convention of §4.1 and carries transfer across lengths, which we do not measure.

Further limitations. The pair of runs that would have told the launch batch and the code revision apart, the same-launch experiment re-run on the pool's stack revision, was never launched, because its certificate run never ran (Appendix C). Both headline takeoff predictions, the hazard ratio at $\theta = 0.70$ against thresholds of 2.00 and 1.50, hit on the complete experiment of their two-depth field of fresh runs (192 of 192 runs, seed arguments disjoint from the pool: 2.25, 95% $[1.66, 3.07]$); their outcome stays undetermined until the deeper experiment, which the stop recorded in Appendix M interrupted, completes, and the disclosure that authorised scoring half the field, stamped while zero re-executed runs existed, excludes depth-coupled predictions. The reward-delay boundary is blocked, not read: its zero-delay reference cell is inconclusive and two rungs of its ladder were never dispatched.

The share of episodes that ended at the token budget or the turn cap is not recorded, since no record of the pool's runs carries a per-episode termination reason, and we substitute no proxy; the recorded aggregates bound it from above, and at $H{=}85$ and $H{=}100$ no evaluation point can have had more than 2.5% of its 256 episodes reach the turn cap. Three further launch batches registered while their cells held zero runs (a deterministic-kernel causal test, a four-execution variance-components decomposition, a cross-family re-execution) carry five predictions; only the intraclass-correlation form of the variance-components statement misses (Appendix B).

O THE TERMINAL-REWARD CELL AT THREE TIMES THE BUDGET

At three times the registered budget the cell's sign prediction falls one seed argument short of a hit and is scored inconclusive, and its reliability prediction is a hit; at the registered budget both had missed (Table 39) and its share prediction had hit (Appendix B.4). Under a purely terminal reward an episode is paid only at its end, and only if it succeeds; a turn's return-to-go is that payment discounted at 0.95 per remaining turn, so the anchored return-to-go, which the method arm compares across trajectories, carries the episode's outcome and length and not the turn's own action, and in a group where no episode succeeds every return is zero and the group-normalised advantage vanishes (Appendix A).

Setup. We re-ran the terminal-reward cell at three times its budget, registered before any of its runs existed, rewriting two tokens of each of its 32 argument vectors, the number of updates ($165 \to 495$) and the output directory, and holding everything else: $H{=}30$, a purely terminal reward, Qwen2.5-7B-Instruct, the same stack revision, both arms, and seed arguments 1–16 with one execution each.

We evaluated 256 episodes every 15 updates, 34 evaluation points in all, and read Y_{495}, the mean success over the seven evaluation points after update $396 = 0.8 \times 495$, that is updates 405 to 495, so it is a multiple of $1/1792$.

The predictions. Before any run existed we registered two. The sign prediction counts the k seed arguments whose seed-paired gap $d_s = Y_{495}(\text{method}, s) - Y_{495}(\text{baseline}, s)$ is strictly positive, a tie counting as not positive: with all 16 seed arguments it is a hit at $k \geq 13$ (exact two-sided binomial $p \leq 0.022$ at $k = 13$), a miss at $k \leq 11$ and inconclusive at $k = 12$, the parent cell's rule. The reliability prediction applies the parent's definition to Y_{495}: a hit when the baseline arm's sample variance exceeds the method arm's, $V_B/V_M > 1$, and the low-tail gap exceeds the high-tail gap, a tail being the mean of an arm's four smallest or four largest values and a gap method minus baseline. Three guards come first, in this order: a floor (either arm mean at or below 0.05 makes both not evaluable), a learnability guard (both arm means below 0.30: not learned at three budgets, neither a hit nor a miss), and a ceiling (either arm mean at or above 0.98 makes both not evaluable).

Readings. All 32 runs finished at update 495, each on its registered argument vector, byte for byte, and on the registered stack revision. The arm means of Y_{495} are 0.9054 (method) and 0.8477 (baseline), clear of all three guards. The sign prediction is one short of a hit and scored inconclusive: of the sixteen seed-paired gaps, listed below in seed-argument order, twelve are strictly positive and four negative, with no ties, so $k = 12$, exact two-sided binomial $p = 0.0768$. The mean gap is 0.0577, and most of it is carried by four seed arguments on which the two arms ended the window far apart, two in each arm's favour; the other twelve lie between -0.0106 and $+0.0943$, ten of them positive.

s	1	2	3	4	5	6	7	8
d_s	+0.0022	+0.0022	+0.9777	+0.0943	−0.4169	+0.9754	−0.9609	+0.0257
s	9	10	11	12	13	14	15	16
d_s	+0.0419	−0.0073	+0.0039	−0.0106	+0.0854	+0.0329	+0.0658	+0.0106

The reliability prediction is a hit on both halves: at three times the budget the arms differ in the lower tail and hardly at all at the top.

Y_{495} over the 16 seed arguments	method	baseline	ratio or gap
sample variance (standard deviation)	0.07013 (0.2648)	0.10508 (0.3242)	$V_B/V_M = 1.498$
mean of the four lowest runs	0.6321	0.4641	0.1680
mean of the four highest runs	0.9992	0.9955	0.0036

What this does not show. Both predictions carry the registered scope: one depth, a purely terminal reward at 495 updates, seed arguments 1–16, this environment family and this model; the cell at the registered budget keeps its three outcomes. The method arm's mean is above the matrix's ceiling gate of 0.90 (Appendix M); this registration set its own ceiling guard at 0.98, which did not fire, so both predictions are scored as registered, but the gap is read with the method arm near the top of the scale.