WHERE SHOULD A FIXED BUDGET OF HUMAN DATA GO?
ALLOCATING ACROSS TIME AND MODES IN RECURSIVE TRAINING

Anonymous authors
Paper under double-blind review

ABSTRACT

Recursive training on model-generated text degrades language models, and human-written data arrests that degradation. We study how to spend a finite human-data budget: one fixed lifetime stock of human-origin tokens, allocated across recursive generations and across monitored regions of the human distribution. We formalise this as a constrained allocation problem and decompose it into four budget-matched policies that isolate each decision, spending randomly, on timing alone, on selection alone, or on both jointly. We enforce matching on two axes at once, lifetime human-origin tokens and total optimizer tokens, realising 0.0381% and 0.00% spreads against a 0.2% bound, and we assemble each corpus by displacement so every arm trains on identical data volume. We show that spending any human budget improves held-out regret by 4.04% over a control trained on that same volume, that concentrating the budget by mode is worth a further 9.59%, and that a back-loaded schedule improves endpoint regret by 6.51% on an interval excluding zero while leaving integrated regret unchanged, so the integral hides a timing effect the endpoint reveals. We then reconstruct every arm's realised purchases from the frozen manifests and verify them to the token against every archived chain total. That audit establishes which comparisons the run licenses: the joint policy never left its base schedule, so the joint-versus-selection comparison measures how far two identical policies drift apart, and the value of coupling remains open. It also bounds the two secondary contrasts, which are confounded with archive replay and with terminal spend. We report 25 certified GPT-2 chains and 50 further chains on Pythia-160M and SmolLM2-135M as a descriptive replication.

1 INTRODUCTION

Generative models now write a large share of the text, code, and images that later models are trained on, closing a loop: a model trained on human data produces synthetic examples, a successor trains on a corpus containing them, and the cycle repeats. Some versions of this loop degrade the learned distribution – held-out error rises, diversity falls, low-probability events disappear (Shumailov et al., 2024; Alemohammad et al., 2024; Briesch et al., 2023; Dohmatob et al., 2024b;a; Suresh et al., 2025) – while others stay bounded, the difference being the data workflow rather than the use of synthetic data itself (Gerstgrasser et al., 2024; Kazdan et al., 2025). Keeping human-written data in the mixture is among the most reliable stabilisers (Seddik et al., 2024). We call the failure *recursive-training degradation* rather than "model collapse", which has been given several incompatible definitions.

Human data is not free: it has to be licensed, cleaned, and held out of evaluation, and showing the same archive to the optimizer again consumes training steps a synthetic-only competitor could spend elsewhere. We therefore treat it as a budget – one fixed allowance for the whole run, counted in *optimizer presentations*, so presenting an example three times costs three times and a policy cannot look efficient by replaying a small archive (Appendix E).

Under that allowance a policy must decide when to introduce human examples and which ones. Early spending may limit error propagation, later spending may address degradation observed during

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.

training; random sampling may leave underrepresented regions uncovered, targeted selection may be sensitive to gaps in the monitoring data. We call a region of the human reference distribution a *mode*: one cell of a partition fixed before the experiment. Under a fixed allowance the two decisions are coupled, since tokens spent at generation three are not available at generation eight and tokens spent on one mode are not available for another – which makes "when" and "what" a single decision (Figure 1).

Does it help to decide both *when* and *where* to spend a limited human-data budget, or is picking the right examples enough on its own? Answering this requires policies that differ in one decision at a time, so we compare four budget-matched policies along two axes: whether the per-generation schedule is uniform or non-uniform, and whether spend is targeted at under-covered modes. *Random* varies neither, *schedule-only* the schedule alone along one non-uniform vector fixed before the run, *selection-only* targeting alone, and *joint* both – joint being the only policy whose spend responds to a state signal at run time. The schedule axis therefore contrasts two *fixed* schedules and is not a test of adaptive pacing, which Section 6.6 costs out. A no-rescue control spends nothing; crediting joint for beating only that control, or only random, would be too easy a test, so the comparison that matters is against the better of the two single-axis policies (Section 4; results in Table 1, parameters in Table 3).

Matching both budgets is what makes those comparisons mean anything. A policy holding a constant human fraction can quietly consume far more human tokens than one that runs briefly, and one replaying a small archive can quietly obtain many more optimizer exposures. We therefore hold the lifetime human-token total and the overall optimizer-token total equal across policies. That isolates whether *allocation* matters once training exposure is controlled, and makes a negative answer informative: if a fixed schedule matches an adaptive one under the same accounting, the added machinery is not earning its keep. One *chain* is one complete recursive run under its own seed; chains are the unit of analysis, and generations inside a chain are repeated observations of one trajectory rather than independent samples.

The study was not externally preregistered; Appendix J documents which design choices were fixed before execution and every difference between the planned and the implemented procedure.

Contributions.

1. We formalise spending a fixed human-data budget as a joint decision over recursive time and monitored modes, and decompose it into four policies that isolate each axis under exact matching on both budgets.
2. We execute the comparison on 25 certified chains and show that selection accounts for most of the measured gain and that the schedule we tested moves the endpoint but not the integral, with both contrasts carrying confounds we identify and bound rather than leave to inference.
3. We reconstruct every arm's realised purchases from the frozen manifests and verify them to the token against every archived chain total. The audit establishes what the run licenses: the joint arm never paced, so it bought what selection-only bought; selection replayed 75 short tail articles every generation, so concentration and duplication are not separable here; and the schedule null on the integral is not null at the endpoint.

2 Related Work

Recursive degradation. Repeatedly fitting a model to its predecessors' samples amplifies approximation and sampling error. Shumailov et al. (2024) separate early tail loss from late distributional collapse, Alemohammad et al. (2024) show quality–diversity deterioration in self-consuming image loops, and Briesch et al. (2023) report falling output diversity in recursive language-model loops; theory makes the dependence on estimator, sample size, and horizon explicit (Dohmatob et al., 2024a;b; Suresh et al., 2025). Properties of the human source can change the direction of the shift, not only its size (Kovač et al., 2025; Wang et al., 2026a), so one scalar "collapse" score is not meaningful across studies. We therefore fix one domain, one recursive protocol, and explicit held-out human outcomes in advance, and track the whole trajectory.

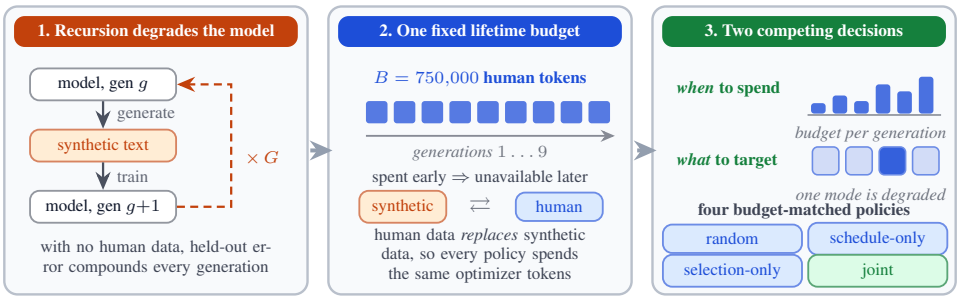

Figure 1: **Budgeted recursive training.** Recursion compounds held-out error; every policy then receives the same lifetime budget B of human-origin tokens, and human examples replace synthetic ones rather than joining them, so a difference between policies is a difference in allocation; that budget forces two competing decisions, *when* to spend and *what* to target, whose crossing gives the four policies of Section 4.

Human data as the stabiliser. Gerstgrasser et al. (2024) find error bounded under accumulation where replacement degrades, Kazdan et al. (2025) that the workflow decides whether degradation is explosive or contained, Seddik et al. (2024) conditions under which collapse is avoided, and Kang et al. (2025) that useful mixture ratios depend on model size and budget; a parallel line edits the synthetic side (Gillman et al., 2024; Zhu et al., 2025; 2026; Yi et al., 2026). All of this establishes that human data helps. None asks how to allocate one exhaustible stock across both generation and mode.

Scheduling and mixtures, the closest competitors. Data-mixture research treats domains as controllable pretraining components, learning static weights (Xie et al., 2023a), predicting performance from mixture proportions (Ye et al., 2025), searching over many small runs (Liu et al., 2025), and lately making the mixture time-dependent (Zhao et al., 2026; Wang et al., 2026b). These are the highest-threat related works, and we state the threat plainly: *varying training time and data domain together is not new*. What is specific here is the conjunction of a recursive human–synthetic chain, exact human-origin optimizer-token accounting, one fixed lifetime stock, pre-committed modes, and matched single-axis controls. That literature also sets a standard we state and do not meet, and Section 6.6 says what our controls cost the comparison.

Selection and tail preservation. Which examples to buy has its own literature (Xie et al., 2023b; Katharopoulos & Fleuret, 2018; Chang et al., 2017; Sener & Savarese, 2018; Ash et al., 2020), including detector-based resampling of likely human text in the recursive setting (Drayson et al., 2025); none of these selectors is claimed as novel here, and our mode score is preregistered and allowed to fail. Because average loss hides degradation on rare groups we also take a tail-retention outcome, following work on worst-group and long-tail robustness (Sagawa et al., 2020; Liu et al., 2021; Sohoni et al., 2020; Menon et al., 2021). That literature carries a warning we build on: a targeted policy observes only its monitor, so if the monitor omits a globally important mode the policy can look locally efficient while making held-out performance worse, and our monitoring-omission intervention is a falsification test for that failure (Appendix H.2).

3 Background and Problem Setting

Generations are indexed $g = 0, \ldots, 9$, so the horizon $G = 10$ counts generation 0, the model trained before any purchase. Sums over purchases run from $g = 1$, and the regret curve is integrated over the same range against generation 0.

The chain. Let $g \in \{0, \ldots, G-1\}$ index one recursive chain against the human reference distribution P_{H}. The first model θ_0 is trained from a pre-committed base partition. At each later generation θ_{g-1} produces a synthetic pool S_g under a fixed prompting and decoding protocol, a constructor combines S_g with the human examples H_g the policy bought, and training restarts from the same pinned pretrained checkpoint, so the recursion runs through the data rather than through accumu-

lated weights. Initialisation, pool size, optimizer, update count, and total optimizer-consumed tokens are held constant across budget-matched policies (Appendix A.3).

Modes and what a policy can see. A pre-committed monitoring map $\mu(x) \in \{1, \dots, M\}$ assigns each eligible human example to one of M modes; here $M = 3$ and the partition is by article length (Appendix A.3). Before training generation g the policy sees $\mathbf{u}_g$, a mode-level lagged statistic, and never the final test partition; R_g denotes the tail-retention outcome instead.

The budget. Let $b_{g,m} \geq 0$ be the human-origin tokens from mode m actually presented to the optimizer while training generation g. Repeated presentation is counted every time; padding is excluded under one fixed rule. Every spending policy satisfies

$$\sum_{g=1}^{G-1} \sum_{m=1}^{M} b_{g,m} = B, \qquad \sum_{g=1}^{G-1} \Big(\sum_{m=1}^{M} b_{g,m} + s_g \Big) = T,$$

where B is the lifetime human budget, s_g the synthetic-origin tokens at generation g, and T the total optimizer-token budget, both fixed before any primary outcome is opened. The first constraint stops a policy winning by buying more human data, the second by training longer. The policies differ only in which arguments of $b_{g,m}$ they adapt: schedule-only chooses the generation totals $b_g = \sum_m b_{g,m}$ but splits each by a fixed neutral rule; selection-only chooses the mode split but follows a fixed vector; joint chooses both; random fixes both. A no-rescue chain is a reference point rather than part of the decomposition, as the accumulation, fixed-fraction, detector-resampling, and oracle-mode conditions would have been had we built them.

Outcome. One independently seeded complete chain is one experimental unit. With L_g^π the held-out human negative log-likelihood for policy π at generation g, regret is taken against the chain's *own* generation-0 value, $r_g^\pi = L_g^\pi - L_0^\pi$, and the chain-level outcome is the trapezoid area under that curve,

$$A^\pi = \sum_{g=1}^{G-1} \tfrac{1}{2} \big(r_{g-1}^\pi + r_g^\pi \big).$$

The reference is within-chain, so A^π carries no cross-arm baseline and a paired contrast inherits no variance from a shared one. Because an integral can cancel a schedule worse early and better late, we report the final-generation regret r_{G-1}^π beside A^π; it was not pre-committed and is descriptive. A pre-committed tail-retention measure, computed on a partition disjoint from the monitoring reference and not identical to the policy's own score, is the confirmatory outcome.

Contrast. The primary contrast is the paired chain-level difference between joint and the strongest eligible non-joint policy, where "strongest" is picked by a preregistered screening rule and cannot be chosen after confirmation outcomes are seen; comparisons are matched on initial assets, chain seed, B, T, horizon, and evaluation data. A monitoring-bias intervention (Appendix A.3) tests whether a targeted policy can satisfy its monitor while harming the true reference distribution.

4 METHOD

4.1 WHAT EACH POLICY MAY ADAPT

All four policies consume the same lifetime human-token budget and the same total optimizer-token budget, so a difference between them is a difference in allocation rather than in how much data they saw. *Both* schedules are fixed before the run: joint is the only arm that reads a state signal, so the schedule axis contrasts two fixed schedules and does not test adaptive pacing.

At generation g a policy observes only the remaining human-token budget, the number of generations left, and a per-mode score computed from a lagged validation reference. It does not observe held-out test data, another policy's outputs, or any primary outcome. A mode with no observations scores zero; if every mode is missing, selection falls back to a seeded random candidate order. The candidate pool is the whole frozen rescue partition at every generation: an example that was bought at generation g is eligible again at $g + 1$, and its tokens are charged again if it is bought again.

Let r denote the remaining human-token budget, s_{base} the base per-generation spend, $s_{\max}$ the per-generation cap, and $F = \max(0,\, G - 2 - g)$ the number of allocation rounds remaining after index g and before terminal reconciliation.

An allocation at index g is assembled into generation $g + 1$'s corpus, so the last active decision is index $G - 2$ and the allocation at $G - 1$ is discarded. At $G - 2$ terminal reconciliation *replaces* the clamp: each arm's allowance becomes its whole remaining balance, so the arms converge on one lifetime total instead of stranding different amounts to indivisibility, which operates on whole articles (mean 4,082 tokens), not 512-token blocks (Table 2).

4.2 THE FOUR POLICIES

Note that $s_{\text{base}} = B/G$ divides by the horizon, not by the number of spending generations, so every arm's last allocation exceeds its nominal rate (Table 2).

Random. Spend $\min(s_{\text{base}}, r)$ each generation; shuffle candidates under the chain seed combined with the generation index, so the shuffle differs at every generation, and take them in order without exceeding budget.

Schedule-only. Spend s_g from a preregistered schedule, selecting in seeded random order under the same seeds as random: the schedule varies, targeting does not. The executed vector is zero for allocation index $g < 5$ and $s_{\max}$ thereafter, with the last generation taking the remainder (Table 2). It reads no state, so it is non-uniform rather than adaptive, and it is back-loaded – the opposite of the front-loading our own motivation in Section 1 calls interesting, which we did not run.

Selection-only. Spend $\min(s_{\text{base}}, r)$ each generation; score candidates by their mode's lagged score, sort descending, break ties by ascending example identifier: targeting varies, pacing does not. The runner supplies the reciprocal of each mode's mean lagged validation NLL, so the mode ranked first has the *lowest* lagged NLL, where the preregistered clipped relative gap would have ranked first the mode that degraded most (Appendix J). Since every candidate in a mode carries the same score and ties break by identifier, the arm buys the same identifier-ordered prefix of the top mode at every generation that mode stays on top (Section 6.3).

Table 3 lists the policy parameters. Unlike combining the fixed schedule with mode-based selection, the joint policy lets per-generation spending respond to monitored mode scores (Appendix G.5).

Joint. Let u be the largest finite mode score and let $\tau_{\text{low}} < \tau_{\text{high}}$ be the urgency thresholds. Desired spend is

$$d(u) = \begin{cases} 0 & u < \tau_{\text{low}} \\ s_{\text{base}} & \tau_{\text{low}} \le u < \tau_{\text{high}} \\ s_{\max} & u \ge \tau_{\text{high}} \end{cases}$$

and realised spend applies a feasibility clamp,

$$b = \text{clamp}\big(d(u),\ \max(0,\, r - s_{\max}F),\ \min(s_{\max},\, r)\big),$$

whose lower bound is a spending floor rather than a reserve: requiring $b \ge r - s_{\max}F$ is the same as requiring the unspent balance $r - b$ to stay within what the F remaining generations can absorb at the cap, so it prevents budget stranding. The upper bound enforces the cap and the remaining budget; candidates are then ranked as in selection-only. Under the executed score the band $[\tau_{\text{low}}, \tau_{\text{high}}) = [0.25, 0.50)$ says the best-covered mode's lagged validation NLL lies in $(2.0, 4.0]$ nats; the reciprocal inverts which end is open. Every held-out NLL this study measured lies inside that range, which is why joint never left its base band (Section 6.3). Two properties of the executed rule follow from the reciprocal substitution. Because u rises as loss falls, the rule spends $s_{\max}$ when the monitored mode's NLL is at or below 2.0 nats and nothing once it passes 4.0, which is the opposite of the rescue direction the preregistered relative gap would have given; and because $u = \max_m s_m$, spend tracks the best-covered mode rather than the most degraded one.

5 EXPERIMENTAL SETUP

The experimental unit is one independently seeded chain run to the pre-committed horizon; generations within a chain are repeated observations of that unit, not independent samples, so every interval is computed across chains. Chains are paired by seed over the preregistered set $\{101, 202, 303, 404, 505\}$, and the quantity estimated is the paired chain-level difference between joint and the strongest eligible non-joint policy, selected by a rule fixed in advance but resolved against the primary outcome. The primary outcome is the regret AUC of Section 3, the confirmatory outcome the pre-committed tail-retention measure on a partition never used for selection, and final-generation regret is reported beside them as a descriptive, post hoc quantity.

Matching both budgets. Every policy in a comparison consumes an identical total optimizer-token count, which the corpus-assembly rule enforces by construction, and a lifetime human-origin count matched as a common ceiling reached up to indivisibility, since a policy cannot spend a fraction of a candidate. The realised spread is reported in Section 6 and is an order of magnitude below the practical effect threshold. A comparison is reported as invalid rather than repaired when it has mismatched paired seed sets, unequal budgets on either axis, nonfinite scores, leakage, or hash mismatches; poor measured performance is not grounds for exclusion. On fixtures the accounting tests recompute token counts from the tokenized batches actually consumed by the optimizer; *on the executed run they do not*, because realised batch records were not retained. Every budget figure below therefore rests on the declared ledger rather than a recount of consumed batches (Appendix E).

Reconstructing what was bought. The per-generation purchase manifests were not archived, but every allocation rule is a deterministic function of the frozen rescue manifest, the chain seed, and the per-mode ranking, all of which are. We re-execute the rules and accept a reconstruction only when it reproduces the chain's archived lifetime total to the token (Appendix D).

The executed configuration, the data-separation regime, and the positive-control reproduction are in Appendices A.2, A.5 and B.

6 RESULTS AND DISCUSSION

6.1 WHAT WE EXECUTED, AND HOW BOTH BUDGETS MATCHED

Five policies crossed with the 5 preregistered seeds gives 25 independently seeded chains, each run to the pre-committed horizon of 10 generations. Every chain completed, and every chain is certified valid with a recorded limitation rather than invalid, on two standing properties of the apparatus (Appendix A.2): the budget ledger is checked against the pre-committed budget rather than recounted from retained batch records, and the per-generation allocation trace was not archived – the second now partly lifted by reconstruction (Section 6.3).

The comparison rests on equal lifetime human-origin tokens *and* equal total optimizer tokens. An earlier execution failed both, so its primary contrast was not reportable; the one secondary contrast matched on the human axis, schedule-only against random, is the one this run replicates. Under the assembly rule adopted here, where purchased human examples *displace* synthetic records rather than being appended, both conditions hold by construction: realised lifetime human spend runs 749,709–749,995 tokens against a 750,000 ceiling, a spread of 0.0381% against 0.2% permitted, and realised total optimizer tokens are identical across all five arms. Displacement also removes the confound the additive rule carried, where the control trained on strictly less data than every spending policy.

6.2 THE PRIMARY COMPARISON IS A NULL, AND AN IDENTITY

The primary contrast is the paired chain-level difference in held-out human NLL-regret AUC between joint and the strongest eligible non-joint policy, lower favouring joint. That policy is selection-only here, identified by lowest mean regret among the four non-joint arms; the selection rule is preregistered, the arm it selects is read from the outcome.

At matched budgets across five paired chains we did not detect a difference between joint allocation and the strongest non-joint policy: paired mean difference 0.0103, 95% Student t interval on four

Table 1: Per-arm results, generated by the analysis pipeline from archived chain artifacts. AUC regret is the area under the generation-wise held-out human NLL-regret curve, measured against each chain's own generation-0 value; lower is better. SD is between-chain standard deviation. Final regret is the same regret at the last generation, r_9, reported beside the integral because an integral cancels a schedule that is worse early and better late; lower is better. Tail retention is the confirmatory outcome at the final generation; higher is better. Every arm is matched on both budgets, so every row is comparable to every other. **Bold** marks the best value among the four spending arms in each outcome column, shared on a tie at displayed precision; the control is not ranked because it spends nothing by construction.

Policy	Chains	Human tokens	AUC regret ↓	SD	Final regret ↓	Tail retention ↑
No rescue (control)	5	0	2.6431	0.0140	0.4770	0.8676
Random	5	749,869	2.5364	0.0080	0.4411	0.8769
Schedule-only	5	749,878	2.5261	0.0194	0.4124	0.8846
Selection-only	5	749,827	**2.2931**	0.0250	**0.3647**	**0.9024**
Joint	5	749,827	2.3034	0.0221	0.3654	**0.9024**

degrees of freedom $[-0.0092, 0.0297]$, or 0.45% of selection-only's mean. Every interval here is of that form; the equivalence region is anchored to random, so the two figures are not on a common scale. That preregistered region is ± 0.0507 in outcome units, and the interval lies entirely inside it, reaching at most 0.0297 from zero.

Against the preregistered region this is a null, and the interval excludes every difference the preregistration calls practically meaningful, not just zero. A power statement computed from the observed spread would put the requirement at about three chains per arm, but it cannot justify the sample size it was computed from. The verdict does not depend on which arm's mean supplies the denominator, but it does depend on the *scale* the region is anchored to: anchored to the incremental gain selection buys over random rather than to random's total regret, the region is ± 0.0049 and the primary interval does *not* fit inside it. Against the preregistered region this is equivalence; against a region scaled to the effect being tested it is inconclusive. Appendix E gives both computations.

The null also has a specific cause, which is that the two policies bought identical data. They share a ranking rule and differ only in how much they spend per generation, so joint reduces *exactly* to selection-only on any chain whose monitored urgency stays inside $[\tau_{\text{low}}, \tau_{\text{high}})$ throughout: the desired spend is then s_{base} at every generation and the clamp never binds. Three pieces of evidence, all from archived artifacts, show that this is what happened. Joint and selection-only reached the same lifetime total, 749,827 tokens, in every one of the five seeds, while the two shuffling arms vary by seed (Table 5). Of the 6,561 desired-spend sequences the urgency rule can produce, exactly 1 reproduces that total, and it is the all-base sequence (Appendix D). And leaving the band would require a validation NLL below 2.0 or above 4.0 nats, outside the range any held-out figure in this study reached. The joint arm therefore bought the same 152 examples as selection-only, in the same generations. The primary contrast compares a policy with itself, so its value measures how far two identical policies drift apart, not the effect of coupling.

The two arms differed only in the training and decoding that followed those purchases. At generation 0, where all five arms train on one corpus under one seed, the spread across arms is 0.00008–0.00054 nats per seed, and the largest per-generation joint-minus-selection gap in any seed is 0.0031–0.0063 nats. Run-to-run variation is small at a single training step and grows as it feeds through decoding and retraining, so the primary interval measures recursive amplification of nondeterminism rather than pacing; the equivalence statements above should be read accordingly.

6.3 What each arm actually bought

Re-executing the allocation rules over the frozen rescue manifests reproduces every archived chain total to the token and pins the GPT-2 selection trace to a single sequence (Table 5, Appendix D). The two shuffling arms bought 171 to 200 distinct examples across all three modes, and no example more than 3 times. Selection-only bought 752 presentations of 152 distinct examples, every one from the tail mode, with 75 of them bought in all 9 spending generations and accounting for 89.8% of the budget – far more replay than the tail pool's size forces (Appendix D).

The selection-versus-random contrast therefore mixes mode concentration, duplication, and within-mode ordering, and the arm called "selection-only" targeted the mode the model was doing *best* on, because the executed score is the reciprocal of NLL rather than the preregistered relative gap (Appendix D).

6.4 What moves the outcome

Because all five arms are matched on both budgets, the secondary contrasts are directly comparable to the primary one and together explain the source of the null. Each is a paired chain-level difference over the same 5 seeds; the percentage beside each is relative to the mean of the arm being improved on. Beside each integrated contrast we give the same contrast on final-generation regret, r_9, whose 2% region is ± 0.0088.

- **Spending human tokens at all helps.** Random against the no-rescue control improves the outcome by 4.04%, 95% CI $[-0.121, -0.092]$, and by 7.53% at the endpoint, CI $[-0.0451, -0.0267]$. Under displacement this arm trains on exactly as much data as the control, so the improvement is attributable to the human origin of part of that data, not its quantity.
- **Which examples are bought matters most, but the contrast is confounded.** Selection-only against random improves the outcome by 9.59%, 95% CI $[-0.269, -0.218]$, at identical total training volume (0.00% difference), and by 17.33% at the endpoint, CI $[-0.0846, -0.0683]$ – roughly twice again what unselective spending is worth. Section 6.3 bounds the reading: the arm replayed 75 examples 9 times each, so mode concentration and duplication are not separated here.
- **Timing does not move the integrated outcome, but does move the endpoint.** Schedule-only against random gives 0.41% of random's mean, on a 95% CI of $[-0.036,\ 0.016]$ in outcome units: an interval containing zero and lying inside the practical equivalence region, at matched human spend and matched totals. This replicates the one clean contrast available from the earlier execution, now with both budgets holding rather than one. On final-generation regret the same contrast is -0.0287, CI $[-0.0357, -0.0218]$, or 6.51% of random's endpoint regret: an interval excluding zero and lying entirely outside the ± 0.0088 region. The back-loaded schedule is worse than uniform while it spends nothing and better once it spends at the cap; the integral cancels the two. This endpoint contrast is confounded by volume as well as timing: under terminal reconciliation schedule-only carries about twice the uniform arms' unspent balance into its final allocation (Table 2), and each generation trains from the pinned checkpoint on that generation's corpus.

Together these explain the primary null and its limits: targeting accounts for most of the improvement we measured, the schedule we tested shows no detectable effect on the integrated outcome, and joint matches targeting alone because in this run it executed targeting alone.

6.5 The confirmatory outcome, and where it disagrees

On the primary contrast the tail-retention outcome agrees more tightly: joint minus the strongest policy is 0.00003, 95% CI $[-0.00100, 0.00105]$, the agreement expected of two arms that bought the same examples.

On one secondary contrast it disagrees: schedule-only against random is 0.00766, 95% CI $[0.00563, 0.00968]$, an interval excluding zero where the integrated outcome shows none. We follow the primary outcome rather than the more favourable of the two metrics: the schedule we tested has no detectable effect on integrated regret, a 6.51% effect at the endpoint, and a small effect on tail retention that excludes zero (Appendix C.3).

Figure 2 in Appendix C.4 plots held-out human NLL against generation: every spending policy ends below the no-rescue control, and the selection-only and joint trajectories lie on top of one another for the whole horizon. This is not a well-powered design. Coefficients of variation run 0.32%–1.09% against a 2% threshold, the paired standard deviation 0.0156 is estimated on four degrees of freedom with its own 95% interval reaching 2.9 times the point estimate, and all 25 chains share a single fixed base subsample (Appendix C.2).

Two further model families. The same five policies, seeds, corpus, and horizon were run on Pythia-160M and SmolLM2-135M, giving 50 more chains. These grids append rather than displace, so total optimizer tokens spread 2.28% and 2.07% across arms and only two contrasts are matched on both axes. Both reproduce the GPT-2 result on SmolLM2-135M (joint minus selection-only -0.0021, CI $[-0.0073, 0.0031]$; schedule-only minus random 0.0026, CI $[-0.0137, 0.0189]$), and the purchase reconstruction finds the same identity. On Pythia-160M the arms differ by 0.13371–0.59061 nats at generation 0 on one corpus and one seed, so that grid measures run-to-run variation more than allocation. None of these chains is certified and none enters the claims (Appendix F).

6.6 Limitations

This is a focused pilot: the certified evidence covers English Wikipedia text and one 124M model at one budget, one horizon, one decoding rule. The modes are constructed rather than given, so a targeted policy can satisfy a misspecified monitor while harming the true distribution (Appendix G.4).

The primary contrast compares a policy with itself. Because the spending rule never left its base setting, the preregistered contrast measures how far two identical policies drift apart, not whether coupling helps (Section 6.3). Testing it needs a spending rule the signal crosses; widening the band or rescaling the score would do that at no extra compute. Pacing did occur on Pythia-160M, but under the executed reciprocal score it paced against the degradation it was meant to answer: spend falls to zero exactly when the monitored mode passes 4.0 nats.

The executed mode score is not the preregistered one. The freeze specified a clipped relative NLL gap, under which a degraded mode scores high; the runner supplied the reciprocal of lagged NLL, under which the best-covered mode does, and both ranking arms sorted on that. The targeting result is therefore about concentrating spend on the lowest-NLL mode, which here is the tail mode, not about targeting under-coverage.

Eight further limitations are stated in full in Appendices G.3 and G.4: targeting is confounded with duplication; the one schedule we tested is back-loaded; the baseline resolves against the data; five seeds bound the paired variance loosely; three of seven planned policies were not built; equivalence does not demonstrate inferiority; the earlier budget failure was fixed structurally rather than diagnosed; and the effect threshold was never externally reviewed.

This work does not claim that synthetic data inevitably causes collapse, that human data is always superior, or that varying time and mode together is new in general.

7 Conclusion

We asked whether it helps to decide both *when* and *where* to spend a limited human-data budget, or whether picking the right examples is enough. Under budgets matched on both axes the joint policy matched selection-only, and reconstructing what each arm bought shows why: joint's spending rule never left its base setting, so the two arms bought the same examples, on GPT-2 and on SmolLM2-135M alike. The comparison measures how far two identical policies drift apart over ten generations of retraining, not whether coupling helps. **The question this work set out to answer is still open**, and testing it needs a spending rule the monitored signal actually crosses. The equivalence region is also wide relative to the effect, and five chains bound the variance loosely.

The secondary contrasts locate the effect that does exist: spending a human budget at all is worth 4.04% against a control trained on identical data volume, concentrating it on 152 tail examples a further 9.59%, and the one non-uniform schedule we tested 0.41% on the integral while improving endpoint regret by 6.51%. Neither of the last two is clean: selection is confounded with replay, the endpoint contrast with terminal volume. Displacing synthetic records rather than appending human ones is what made the budget comparison valid in the first place; setting thresholds the monitored signal never crossed, on a score whose sign was inverted, is what made the primary comparison uninformative. What would move the question forward is a spending rule the signal crosses, a selection rule that depletes its pool, a score with the preregistered orientation, and a front-loaded schedule.

AI USE STATEMENT

In this work, we used generative AI tools for manuscript drafting and editing across several revisions, and for assistance with analysis and plotting code. We did not use generative AI tools to originate the research question, to design the experiments, or to produce, alter, or interpret any experimental result: every number in this paper is emitted by a generator script from archived run artifacts, and none is typed by hand. All AI-assisted text and code was reviewed by the authors. We take responsibility for the final content of this work, including text, claims, and artifacts produced with the aid of generative AI.

REPRODUCIBILITY STATEMENT

Every result number in this paper is a macro emitted by a generator script from immutable run artifacts; a bare decimal in a section is treated as a defect. The generators, the runs they read, and the archived artifact hashes are described in Appendix I; the executed configuration, including the schedule vector, the urgency transform, and the training settings actually used, is in Appendix A.2; corpus construction and the frozen partition are in Appendix A.6 and Appendix A.5; the estimator, the interval procedure, and the practical-equivalence region are in Appendix C.2. Deviations between what was preregistered and what executed are listed item by item in Appendix J rather than summarised, and the allocation rules are re-executed over the frozen rescue manifests in Appendix D, which refuses to emit anything unless every archived chain total is reproduced to the token.

ETHICS STATEMENT

This study trains small language models on publicly available English Wikipedia text and involves no human subjects, no personal data, and no model release. Its subject matter is the degradation of models trained on their own output; we report an equivalence and a null rather than a method, and Section 6.6 and Appendix G.4 state the conditions under which the reported comparisons do not license a recommendation.

REFERENCES

- Sina Alemohammad, Josue Casco-Rodriguez, Lorenzo Luzi, Ahmed Imtiaz Humayun, Hossein Babaei, Daniel LeJeune, Ali Siahkoohi, and Richard G. Baraniuk. Self-consuming generative models go MAD. In *International Conference on Learning Representations*, 2024. URL https://openreview.net/forum?id=ShjMHfmPs0. 1, 2
- Jordan T. Ash, Chicheng Zhang, Akshay Krishnamurthy, John Langford, and Alekh Agarwal. Deep batch active learning by diverse, uncertain gradient lower bounds. In *International Conference on Learning Representations*, 2020. URL https://openreview.net/forum?id=ryghZJBKPS. 3, 27
- Martin Briesch, Dominik Sobania, and Franz Rothlauf. Large language models suffer from their own output: An analysis of the self-consuming training loop. *arXiv preprint arXiv:2311.16822*, 2023. URL https://arxiv.org/abs/2311.16822. 1, 2
- Haw-Shiuan Chang, Erik Learned-Miller, and Andrew McCallum. Active bias: Training more accurate neural networks by emphasizing high variance samples. In *Advances in Neural Information Processing Systems*, volume 30, 2017. URL https://proceedings.neurips.cc/paper/2017/hash/2f37d10131f2a483a8dd005b3d14b0d9-Abstract.html. 3, 27
- Elvis Dohmatob, Yunzhen Feng, and Julia Kempe. Model collapse demystified: The case of regression. In *Advances in Neural Information Processing Systems*, volume 37, 2024a. URL https://proceedings.neurips.cc/paper_files/paper/2024/hash/53dbd7e34fab703a639964e2d3ee9e84-Abstract-Conference.html. 1, 2
- Elvis Dohmatob, Yunzhen Feng, Pu Yang, Francois Charton, and Julia Kempe. A tale of tails: Model collapse as a change of scaling laws. In *Proceedings of the 41st Interna-*

tional Conference on Machine Learning, volume 235 of *Proceedings of Machine Learning Research*, pp. 11165–11197, 2024b. URL https://proceedings.mlr.press/v235/dohmatob24b.html. 1, 2

- George Drayson, Emine Yilmaz, and Vasileios Lampos. Machine-generated text detection prevents language model collapse. In *Proceedings of the 2025 Conference on Empirical Methods in Natural Language Processing*, pp. 29657–29673, 2025. doi: 10.18653/v1/2025.emnlp-main.1506. URL https://aclanthology.org/2025.emnlp-main.1506/. 3, 26, 28
- Matthias Gerstgrasser, Rylan Schaeffer, Apratim Dey, Rafael Rafailov, Henry Sleight, John Hughes, Tomasz Korbak, Rajashree Agrawal, Dhruv Pai, Andrey Gromov, Daniel A. Roberts, Diyi Yang, David L. Donoho, and Sanmi Koyejo. Is model collapse inevitable? breaking the curse of recursion by accumulating real and synthetic data. In *Conference on Language Modeling*, 2024. URL https://openreview.net/forum?id=5B2K4LRgmz. 1, 3, 26
- Nate Gillman, Michael Freeman, Daksh Aggarwal, Chia-Hong Hsu, Calvin Luo, Yonglong Tian, and Chen Sun. Self-correcting self-consuming loops for generative model training. In *Proceedings of the 41st International Conference on Machine Learning*, volume 235 of *Proceedings of Machine Learning Research*, pp. 15646–15677, 2024. URL https://proceedings.mlr.press/v235/gillman24a.html. 3, 26
- Feiyang Kang, Newsha Ardalani, Michael Kuchnik, Youssef Emad, Mostafa Elhoushi, Shubhabrata Sengupta, Shang-Wen Li, Ramya Raghavendra, Ruoxi Jia, and Carole-Jean Wu. Demystifying synthetic data in LLM pre-training: A systematic study of scaling laws, benefits, and pitfalls. In *Proceedings of the 2025 Conference on Empirical Methods in Natural Language Processing*, 2025. doi: 10.18653/v1/2025.emnlp-main.544. URL https://aclanthology.org/2025.emnlp-main.544/. 3
- Angelos Katharopoulos and Francois Fleuret. Not all samples are created equal: Deep learning with importance sampling. In *Proceedings of the 35th International Conference on Machine Learning*, volume 80 of *Proceedings of Machine Learning Research*, pp. 2525–2534, 2018. URL https://proceedings.mlr.press/v80/katharopoulos18a.html. 3, 27
- Joshua Kazdan, Rylan Schaeffer, Apratim Dey, Matthias Gerstgrasser, Rafael Rafailov, David L. Donoho, and Sanmi Koyejo. Collapse or thrive: Perils and promises of synthetic data in a self-generating world. In *Proceedings of the 42nd International Conference on Machine Learning*, volume 267 of *Proceedings of Machine Learning Research*, pp. 29469–29494, 2025. URL https://proceedings.mlr.press/v267/kazdan25a.html. 1, 3, 26
- Grgur Kovač, Jérémy Perez, Rémy Portelas, Peter Ford Dominey, and Pierre-Yves Oudeyer. Recursive training loops in LLMs: How training data properties modulate distribution shift in generated data? In *Proceedings of the 2025 Conference on Empirical Methods in Natural Language Processing*, pp. 32290–32309, 2025. doi: 10.18653/v1/2025.emnlp-main.1643. URL https://aclanthology.org/2025.emnlp-main.1643/. 2
- Evan Z. Liu, Behzad Haghgoo, Annie S. Chen, Aditi Raghunathan, Pang Wei Koh, Shiori Sagawa, Percy Liang, and Chelsea Finn. Just train twice: Improving group robustness without training group information. In *Proceedings of the 38th International Conference on Machine Learning*, volume 139 of *Proceedings of Machine Learning Research*, pp. 6781–6792, 2021. URL https://proceedings.mlr.press/v139/liu21f.html. 3, 28
- Qian Liu, Xiaosen Zheng, Niklas Muennighoff, Guangtao Zeng, Longxu Dou, Tianyu Pang, Jing Jiang, and Min Lin. RegMix: Data mixture as regression for language model pre-training. In *International Conference on Learning Representations*, 2025. URL https://proceedings.iclr.cc/paper_files/paper/2025/hash/5f67d864aae6115374fed7beddd119e0-Abstract-Conference.html. 3
- Aditya Krishna Menon, Sadeep Jayasumana, Ankit Singh Rawat, Himanshu Jain, Andreas Veit, and Sanjiv Kumar. Long-tail learning via logit adjustment. In *International Conference on Learning Representations*, 2021. URL https://openreview.net/forum?id=37nvvqkCo5. 3, 28
- Shiori Sagawa, Pang Wei Koh, Tatsunori B. Hashimoto, and Percy Liang. Distributionally robust neural networks for group shifts: On the importance of regularization for worst-case generalization. In *International Conference on Learning Representations*, 2020. URL https://openreview.net/forum?id=ryxGuJrFvS. 3, 26, 28

- Mohamed El Amine Seddik, Suei-Wen Chen, Soufiane Hayou, Pierre Youssef, and Merouane Debbah. How bad is training on synthetic data? a statistical analysis of language model collapse. In *Conference on Language Modeling*, 2024. URL https://openreview.net/forum?id=4b55130ce33429ab600e5cde927dfe10a36328ba. 1, 3
- Ozan Sener and Silvio Savarese. Active learning for convolutional neural networks: A core-set approach. In *International Conference on Learning Representations*, 2018. URL https://openreview.net/forum?id=H1aIuk-RW. 3, 27
- Ilia Shumailov, Zakhar Shumaylov, Yiren Zhao, Nicolas Papernot, Ross Anderson, and Yarin Gal. AI models collapse when trained on recursively generated data. *Nature*, 631:755–759, 2024. doi: 10.1038/s41586-024-07566-y. URL https://www.nature.com/articles/s41586-024-07566-y. 1, 2
- Nimit S. Sohoni, Jared A. Dunnmon, Geoffrey Angus, Albert Gu, and Christopher Ré. No subclass left behind: Fine-grained robustness in coarse-grained classification problems. In *Advances in Neural Information Processing Systems*, volume 33, 2020. URL https://proceedings.neurips.cc/paper_files/paper/2020/hash/e0688d13958a19e087e123148555e4b4-Abstract.html. 3, 28
- Ananda Theertha Suresh, Andrew Thangaraj, and Aditya Nanda Kishore Khandavally. Rate of model collapse in recursive training. In *Proceedings of the 28th International Conference on Artificial Intelligence and Statistics*, volume 258 of *Proceedings of Machine Learning Research*, pp. 1396–1404, 2025. URL https://proceedings.mlr.press/v258/suresh25a.html. 1, 2
- Yaxuan Wang, Zhongteng Cai, Yujia Bao, Xueru Zhang, and Yang Liu. Observations and remedies for large language model bias in self-consuming performative loop. In *Proceedings of the 64th Annual Meeting of the Association for Computational Linguistics*, 2026a. doi: 10.18653/v1/2026.acl-long.1561. URL https://aclanthology.org/2026.acl-long.1561/. 2
- Yifan Wang, Binbin Liu, Fengze Liu, Yuanfan Guo, Jiyao Deng, Xuecheng Wu, Weidong Zhou, Xiaohuan Zhou, and Taifeng Wang. TiKMiX: Efficient semi-dynamic data mixture via data influence for LLM pre-training. In *Proceedings of the 64th Annual Meeting of the Association for Computational Linguistics*, pp. 5777–5793, 2026b. doi: 10.18653/v1/2026.acl-long.261. URL https://aclanthology.org/2026.acl-long.261/. 3
- Sang Michael Xie, Hieu Pham, Xuanyi Dong, Nan Du, Hanxiao Liu, Yifeng Lu, Percy Liang, Quoc V. Le, Tengyu Ma, and Adams Wei Yu. DoReMi: Optimizing data mixtures speeds up language model pretraining. In *Advances in Neural Information Processing Systems*, volume 36, 2023a. URL https://proceedings.neurips.cc/paper_files/paper/2023/hash/dcba6be91359358c2355cd920da3fcbd-Abstract-Conference.html. 3
- Sang Michael Xie, Shibani Santurkar, Tengyu Ma, and Percy Liang. Data selection for language models via importance resampling. In *Advances in Neural Information Processing Systems*, volume 36, 2023b. URL https://proceedings.neurips.cc/paper_files/paper/2023/hash/6b9aa8f418bde2840d5f4ab7a02f663b-Abstract-Conference.html. 3, 26, 27
- Jiasheng Ye, Peiju Liu, Tianxiang Sun, Jun Zhan, Yunhua Zhou, and Xipeng Qiu. Data mixing laws: Optimizing data mixtures by predicting language modeling performance. In *International Conference on Learning Representations*, 2025. URL https://openreview.net/forum?id=jjCB27TMK3. 3
- Bingji Yi, Qiyuan Liu, Yuwei Cheng, and Haifeng Xu. Escaping model collapse via synthetic data verification: Near-term improvements and long-term convergence. In *VerifAI: AI Verification in the Wild, Workshop at ICLR 2026*, 2026. URL https://arxiv.org/abs/2510.16657. 3
- Kaiyan Zhao, Zhongtao Miao, Akiko Aizawa, and Yoshimasa Tsuruoka. RegMix-D: Dynamic data mixing via proxy training trajectories. *arXiv preprint arXiv:2606.18663*, 2026. URL https://arxiv.org/abs/2606.18663. 3
- Huminhao Zhu, Fangyikang Wang, Tianyu Ding, Qing Qu, and Zhihui Zhu. Analyzing and mitigating model collapse in reflow methods. In *Conference on Parsimony and Learning*, volume 328 of *Proceedings of Machine Learning Research*, pp. 314–340, 2026. URL https://proceedings.mlr.press/v328/zhu26a.html. 3

Table 2: The executed per-generation allowances, in both index conventions. An allocation made at index g is consumed by generation $g+1$; the allocation at index 9 has no successor and is discarded; at index 8 every spending arm's allowance is raised to its remaining budget. The last row is the realised spend of the selection-only and joint arms only, reconstructed in Appendix D; schedule-only's terminal allocation is likewise its whole unspent balance, which after three generations at the cap is substantially larger than the uniform arms', and Section 6 treats that asymmetry as a confound on the endpoint contrast. Generated from the run script and the terminal-reconciliation rule.

Allocation index g	0	1	2	3	4	5	6	7	8	9
Schedule-only allowance	0	0	0	0	0	150,000	150,000	150,000	150,000	150,000
Schedule-only after reconciliation	0	0	0	0	0	150,000	150,000	150,000	remainder	discarded
Random, selection-only, joint allowance	75,000	75,000	75,000	75,000	75,000	75,000	75,000	75,000	remainder	discarded
Consumed by generation	–	1	2	3	4	5	6	7	8	9
Selection-only and joint, realised	–	74,817	74,817	74,817	74,817	74,817	74,817	74,817	74,817	151,291

- Xuekai Zhu, Daixuan Cheng, Hengli Li, Kaiyan Zhang, Ermo Hua, Xingtai Lv, Ning Ding, Zhouhan Lin, Zilong Zheng, and Bowen Zhou. How to synthesize text data without model collapse? In *Proceedings of the 42nd International Conference on Machine Learning*, volume 267 of *Proceedings of Machine Learning Research*, pp. 79746–79771, 2025. URL `https://proceedings.mlr.press/v267/zhu25d.html`. 3

A Apparatus, configuration, and data

A.1 Experimental Apparatus

Figure 1 shows the recursive procedure, and the same procedure stated in words is this. Recursion compounds held-out error; every policy then receives the same lifetime budget B of human-origin tokens, and human examples replace synthetic ones rather than joining them, so a difference between policies is a difference in allocation; that budget forces two competing decisions, *when* to spend and *what* to target, whose crossing gives the four policies of Section 4. One *chain* is one independently seeded traversal of the loop for the full horizon G. Chains, not generations, are the experimental units: generations within a chain are repeated observations of the same trajectory and are not independent samples.

A.2 Executed Configuration

The pilot configuration was fixed before any chain ran, and every scientific value in it traces to written evidence recorded at the time: the retraining regime, the corpus subsample, the lifetime and per-generation budgets, the ordered seed set, the horizon, the outcome definitions, the contrast, and the exclusion rules. Two further reference configurations remain unregistered and the runner continues to refuse them.

The executed run is the five policies of Section 4 crossed with the five preregistered seeds: twenty-five chains at horizon ten, GPT-2 at a pinned revision, WikiText-103 subsampled to four hundred articles from the pre-committed base-training partition, a lifetime budget of $750{,}000$ human-origin optimizer tokens per chain, and generation prompts drawn from a partition held disjoint from training data. Execution took 8.53 hours across four accelerators.

Budget matching is asserted after execution rather than assumed, over the whole run rather than per shard, and a violation is a non-zero exit rather than a warning. That assertion is what determines whether the primary comparison in Section 6 is reportable at all. On an earlier execution it failed, and nothing from that execution was reported. On the run described here it passes on both budgets.

A.3 Specification

Indexing and the terminal rule. An allocation made at generation g is assembled into generation $g+1$'s training corpus, so purchases are decided at $g = 0, \ldots, G-2$ and the allocation a policy would make at $G-1$ is discarded. At $g = G-2$ every spending arm's allowance is raised to its whole remaining budget, so that arms converge on the same lifetime total instead of stranding

different amounts to indivisibility; each arm still chooses *which* examples by its own rule. Table 2 gives both index conventions.

Everything a reader needs in order to check the claims above, in one place. Values shown in bold brackets are recorded in the released run configuration and are not reproduced in this draft; each is one number, not an experiment.

Model and corpus. GPT-2, 124M parameters, at a pinned revision. WikiText-103 subsampled to the first four hundred base-training articles, tokenized with the GPT-2 tokenizer at revision `607a30d7` into blocks of 512 tokens. The assembled corpus holds about 3,280 records in every generation after the first. The positive control of Appendix B runs on WikiText-2, a different corpus from the one the pilot uses.

Initialisation. Each generation trains a freshly initialised learner from the same pinned pretrained GPT-2 checkpoint, not from the previous generation's weights. The recursion therefore runs through the training data only. Because the starting point is pretrained rather than random, human-origin knowledge re-enters through the initial weights at every generation, which bounds how far these results speak to training from scratch.

Budget and schedule. Horizon $G = 10$ with generations indexed 0 to 9. Lifetime human budget $B = 750{,}000$ optimizer-presented tokens, which is about 4.5% of the tokens the optimizer consumes over the horizon. Base spend $s_{\text{base}} = B/G$ and cap $s_{\max} = 2B/G$. Candidates are indivisible: a policy cannot buy a fraction of an example; they average 4,082 tokens. Monitoring lag is one generation. Seeds are the preregistered set $\{101, 202, 303, 404, 505\}$. The schedule-only vector is zero at allocation indices 0 to $5 - 1$ and $s_{\max}$ from index 5 on; with the terminal rule and the discarded final allocation, Table 2 gives what every arm was allowed and what the ranking arms spent.

Modes. Modes are defined from article length: the `article_length_quantile` definition, frozen on 2026-08-08 over the alternative of Wikipedia category labels, which are not part of the pinned dataset release and cannot be recomputed identically. Thresholds are computed once from the whitespace-token distribution of the training split after exact deduplication and applied unchanged to the validation and test partitions: an article of at most 1,106 words, the bottom decile, is *tail*; one above 2,661 words, the top half, is *common*; the rest are *mid*. So $M = 3$. The rescue partition holds 2,105 common, 1,697 mid and 433 tail candidates; the validation partition 31, 26 and 3; the test partition 31, 19 and 10. The tail-mode set supplied to the confirmatory metric is $\{\text{tail}\}$, named in the configuration rather than inferred from the policy's own score. This is a deterministic, reproducible partition, not a claim that article length captures every meaningful minority region of human text.

The mode score and the urgency thresholds, as executed. The runner scores the validation partition per mode after each generation's training and hands the policies $s_m = 1/\overline{\text{NLL}}_m$, the reciprocal of mode m's mean held-out NLL on the previous generation's model. Selection-only and joint rank candidates by s_m descending and break ties by ascending identifier, so the first-ranked mode is the one with the lowest lagged NLL; a candidate whose mode is absent from the statistics scores zero, and a nonfinite score aborts the chain. Joint's urgency is $u = \max_m s_m$, and its thresholds are $\tau_{\text{low}} = 0.25$ and $\tau_{\text{high}} = 0.50$, which on this scale are a lowest-mode NLL of 4.0 and 2.0 nats. The generation-0 value of s_m is snapshotted as the reference the tail-retention metric divides by. The preregistered score was different: $u_{g,m} = \text{clip}\big((L_{g-1,m} - L_{0,m})/\max(|L_{0,m}|, 10^{-8}), 0, 1\big)$, a relative NLL gap that is zero for an undegraded mode and rises with degradation, under which the thresholds would have read as 25% and 50% relative increases in a mode's NLL. That formula is recorded in the policy configuration and the method freeze and was not the one the runner supplied. Appendix J records the discrepancy.

Outcomes. The primary outcome is the area under the generation-wise held-out human NLL-regret curve, regret being measured against each chain's own generation-0 value. The confirmatory outcome is a pre-committed ratio-based tail-retention measure, chosen over an NLL-gap alternative on reliability and independence grounds, and computed without reference to the policy's own selection signal. It is the mean over tail modes of the current mode score divided by its generation-0

Table 3: Preregistered policy parameters, at the budget the run actually used. Generated from `configs/policy/joint.json`; the lifetime budget B is read from the run artifacts, and the per-generation spends are the preregistered ratios B/G and $2B/G$ at that budget.

Preregistered parameter	Value
Horizon G	10
Lifetime human tokens B	750,000
Base generation spend $s_{\text{base}} = B/G$	75,000
Maximum generation spend $s_{\max} = 2B/G$	150,000
Candidate granularity	one example (indivisible)
Monitoring lag (generations)	1
Desired spend, $u < \tau_{\text{low}} = 0.25$	0
Desired spend, $0.25 \le u < \tau_{\text{high}} = 0.50$	s_{base}
Desired spend, $u \ge 0.50$	$s_{\max}$

reference, each ratio clipped to $[0, 1]$:

$$R_g \;=\; \frac{1}{|T|} \sum_{m \in \mathcal{M}_{\text{tail}}} \min\left(1, \max\left(0, \frac{s_m^{(g)}}{s_m^{(0)}}\right)\right),$$

where T is the tail-mode set and $s_m^{(g)}$ the score of mode m at generation g. It equals one when tail coverage is fully preserved. Reference scores must be positive, and the metric never reads the policy's selection signal. Final-generation regret, r_{G-1}, is reported as a descriptive quantity and was not pre-committed.

Training and decoding. Training uses blocks of 512 tokens with the loss taken on the last 256 tokens of each block, batch size 8, one epoch per generation, and `bfloat16`. Decoding is top-k with $k = 50$, temperature 1.0 and no nucleus truncation or beam search. The global seed is 42. No learning rate is passed to the training command, and the pinned upstream training script sets none, so the optimizer runs at the training library's defaults for the pinned release (`transformers` 4.48.3): AdamW at a peak learning rate of 5×10^{-5}, linear decay to zero over the epoch, no warm-up, and no weight decay. A replication should set these explicitly rather than inherit them. Both the training recipe and the decoding protocol are first-order drivers of recursive degradation.

A.4 Preregistered policy parameters

These were fixed in the Week 2 method freeze using interfaces, fixtures, and synthetic results. No primary outcome influenced any score, schedule, seed, threshold, exclusion rule, or interpretation.

A.5 Data Separation

Each example receives a stable content hash before splitting. Five partitions are held disjoint: base human training data, per-generation rescue candidates, generation prompts, validation data, and final human test data. No final test example may influence prompting, selection, threshold tuning, early stopping, or hyperparameter choice.

A.6 Data Appendix

The domain was pre-committed and executed. The corpus is WikiText-103, subsampled to the first 400 `base_train` articles, tokenized with the GPT-2 tokenizer at revision `607a30d7` into blocks of 512 tokens. The assembled training corpus holds the same number of records in every generation after the first, about 3,280, because rescued human examples displace synthetic records one for one rather than being appended. The record count is what displacement holds fixed; it is constant across arms within a generation, which is what the equal-optimizer-exposure claim rests on.

The released artifacts carry the following, and this appendix indexes them rather than restating values that would then exist in two places:

Table 4: Positive-control perplexity by generation, both arms, generated from the committed per-arm result artifacts.

Generation	Fully synthetic	Human mixed
0	29.5885	29.5903
1	33.3678	29.8961
2	36.0165	30.0915
3	38.3181	30.1769
4	40.3815	30.2397
5	42.2049	30.2691
6	44.0361	30.3092
7	45.8288	30.3122
8	47.6292	30.3334
9	49.2302	30.3143
10	50.8309	30.3320
Degradation ratio	1.7179	1.0251

- Dataset identifier, revision hash, and licence, with the licence audit that selected it over the audited alternatives.
- Preprocessing: tokenizer identity and revision, normalisation, and the exact filter chain, each with a content hash.
- Partition statistics for all five partitions (base training, rescue candidates, prompts, validation, final test), reporting example counts, token counts, and per-partition manifest hashes.
- The mode definition, stated in full in Appendix A.3: $M = 3$ modes by article length, with the tail mode the bottom decile.
- Overlap report: exact and near-duplicate checks across every partition pair, with the near-duplicate threshold declared before it was applied.

Final test data influences no prompt, selection decision, threshold, stopping rule, or hyperparameter. This constraint is enforced by the disjointness tests rather than asserted.

B Positive Control

This control confirms that an unmitigated recursive loop degrades in our apparatus and that a fully human-replenished arm does not. It does not validate the components the pilot's claims rest on: it runs on a different corpus from the pilot, compares $\alpha = 0$ against $\alpha = 1$ rather than a constrained budget, and predates the displacement assembly rule. Read it as a check that the loop reproduces known degradation, not as an end-to-end test of the budgeted displacement pipeline.

Before any treatment comparison, the pipeline is validated against a published result: Drayson et al.'s recursive-training study, at upstream commit `feb8511479a2` (abbreviated; the full hash is recorded in the run's artifact ledger), GPT-2 on WikiText-2 over eleven generations, with the paper's importance-sampling mitigation disabled so that the unmitigated baseline is what is measured.

Both arms complete every planned generation. Per-generation perplexity and the resulting degradation ratios are given in Table 4, generated from the committed run artifacts. The fully synthetic arm ($\alpha = 0$) degrades substantially; the human-mixed arm ($\alpha = 1$) is close to flat. The synthetic arm lies above the mixed arm at every subsequent generation and the two never re-cross, so the expected ordering and degradation direction are recovered.

Two independent executions were performed, on different accelerators and different dependency stacks. Every reported quantity agrees between them to well under one per cent, including the case where the two arms computed their shared baseline generation separately. The per-execution values and their differences are recorded in the released run artifacts rather than restated here.

Agreement with the published values is assessed against a five per cent relative difference. This is an internal engineering tolerance, adopted because the source provides no suitable uncertainty for the endpoint, and it must not be read as proof of exact replication. All deviations from the upstream configuration are enumerated in the accompanying evidence record.

Stage A uses WikiText-2, the upstream default. The pilot design targets a larger corpus from the same family; the two stages therefore use different corpus sizes, recorded as a deviation before execution rather than after.

The reported table is regenerated from saved metrics by the same command that produced them, with every recorded artifact hash verified. Taken together with the executed run, this establishes that the apparatus reproduces recursive degradation on WikiText-2 and runs the full allocation loop end to end under matched budgets. The control validates the apparatus on a corpus the pilot does not use, and it runs at $\alpha = 1$, fully human, so it checks the degradation mechanism rather than the mitigation effect at the pilot's 4.5% operating point.

C Evaluation, statistics, and trajectories

C.1 Evaluation Appendix

One coupling is worth stating plainly. The confirmatory tail-retention measure was chosen to be independent of the policy's selection signal, and the two are computed on disjoint partitions. Under the executed reciprocal score, however, the selector ranks modes by inverse validation NLL and the retention measure is a normalised ratio of inverse test NLL on the tail mode. Both therefore reward the same functional quantity on the same mode, which makes selection-only's advantage on the confirmatory outcome partly structural rather than independent corroboration. A retention measure built on a non-likelihood criterion – lexical diversity, tail n-gram coverage, or downstream accuracy – would not share that form, and we did not compute one.

The tail-retention definition is pre-committed: the ratio-based metric, selected over an NLL-gap alternative on reliability and independence grounds, and recorded before any outcome was opened. Both candidates were implemented and audited for determinism, independence from the policy's selection signal, sensitivity to tail degradation, discriminant validity against non-tail modes, and split-half reliability; the pre-committed choice is the one reported throughout this paper. Mode scores are the reciprocal of mean held-out negative log-likelihood, a monotone-decreasing transform of NLL, and the generation-0 value is snapshotted as the reference every later generation is measured against.

Held-out human negative log-likelihood is computed from target-token log probabilities on the pre-committed test partition. Regret is aggregated across the horizon as the area under the generation-wise curve.

Both tail candidates are computed without reference to the policy's under-coverage score, so no policy can be rewarded for agreeing with its own selection signal. This is a structural property of the interfaces, not a convention. It is weakened in the executed run by the fact that the policies ranked on the same reciprocal-NLL statistic the tail metric is built from, although on a different partition (Appendix J).

C.2 Statistics Appendix

Uncertainty is computed across independent chains. Multiple generations within one chain are repeated observations and do not increase the effective sample size.

The primary contrast is the joint policy against the strongest eligible non-joint baseline, paired by chain seed. Paired differences are summarised by their mean with a two-sided 95% Student t interval on four degrees of freedom, the preregistered seed set being five chains per arm. The smallest scientifically meaningful effect was preregistered before outcomes were opened, at 2% relative, taken against the random arm's mean regret, and the interpretation that follows is selected mechanically from precommitted templates covering favourable, null, harmful, and uncertain outcomes.

Which arm satisfies "strongest eligible non-joint baseline" is read from the outcome rather than fixed in advance; the selection rule is what was preregistered. This is the conservative direction, since it compares the joint policy against the best baseline available to it rather than a convenient one.

C.3 Integrated outcome, endpoint, and power

These passages are summarised in Section 6 and stated here in full.

Why the integral and the endpoint disagree. At 0.87% of the random level it sits below the relative bar the preregistration fixes for the primary outcome, though that bar was set for that outcome. The primary conclusion follows the primary outcome rather than the more favorable of the two metrics: in summary, the schedule we tested has no detectable effect on integrated chain-level regret, a 6.51% effect on final-generation regret, and a small but statistically distinguishable effect on tail retention that does not meet any preregistered bar for being called practically minor.

Two features of the design push in the same direction. First, the primary outcome integrates regret over the whole horizon, so a schedule that is worse early and better late has those effects partly cancelled. Figure 2 shows exactly that shape: the schedule-only trajectory runs above random through the early generations, crosses it between generations five and six, and ends below it. An integrated statistic obscures early-versus-late effects in a study about timing specifically, which is why we report the endpoint alongside it. Second, the tail-retention contrasts are lopsided in a way the reconstruction explains: adding a schedule to random moves tail retention by 0.00766, while adding pacing to selection moves it by 0.00003, a factor of about 250. An earlier draft left three explanations open. The reconstruction in Section 6.3 resolves this: joint never effectively paced, and reduced to selection-only on every GPT-2 chain. Whether pacing contributes nothing once targeting is present, and whether a score computed from 3 tail validation examples would have paced on noise had it paced at all, remain untested.

What five seeds do and do not bound. Figure 2 plots held-out human NLL against generation. Every policy that spends its budget ends below the no-rescue control, and the selection-only and joint trajectories lie on top of one another for the whole horizon, showing the identity established in Section 6.3 directly rather than only asserting it. Coefficients of variation on the primary outcome range from 0.32% to 1.09% against a preregistered effect threshold of 2%, and the paired difference between joint and the strongest policy has a standard deviation of 0.0156 where that threshold corresponds to 0.0507. An earlier reading of these coefficients took the tight spread as evidence that chain count was not the binding constraint; that reading does not hold up. With 5 chains the standard deviation is estimated on four degrees of freedom, and its own 95% interval runs $[0.0093, 0.0448]$: the upper end is 2.9 times the point estimate, and at that value the preregistered region is only 1.1 standard deviations wide, which is not a well-powered design. The power statement is also computed from the observed spread rather than from an independent estimate, so it cannot be used to justify the sample size it was computed from. Separately, all 25 chains share one fixed base subsample, so these estimates cover seed and decoding variability only and say nothing about a different corpus draw, which is the likeliest reason the coefficients look this tight.

C.4 Pilot trajectories

This figure is referenced from Section 6 and carries the trajectory evidence for the claims made there. It sits here rather than in the body only because of the main-text page limit.

D Purchase reconstruction

What the selection contrast cannot separate. The selection-versus-random contrast mixes three things this design cannot separate: concentrating spend on the tail mode, presenting 75 short articles 9 times each, and non-random ordering within a mode. The data support either reading: that targeting helps, or that replaying a small tail archive helps. And the arm called "selection-only" did not target under-covered modes: it targeted the mode the model was doing best on by lagged validation NLL, because the executed score is the reciprocal of NLL rather than the preregistered relative gap. That the mode it fixed on happens to be the tail mode – the one the confirmatory outcome measures – reflects a property of the corpus, not a property of the policy.

Why the joint arm's spend sequence is pinned. Three pieces of evidence, all from archived artifacts, show that this is what happened. First, joint and selection-only reached the same lifetime

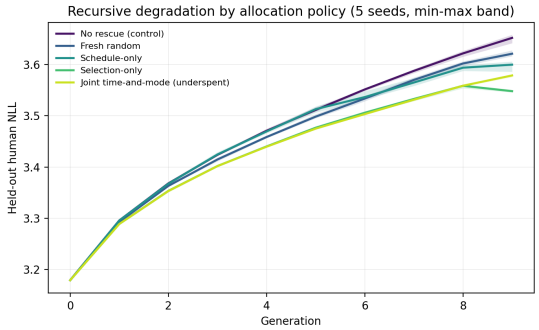

Figure 2: Held-out human NLL against generation for each policy, mean over the preregistered seed set with a min–max band. The generated legend labels "Fresh random" and "Joint time-and-mode" are the random and joint policies named everywhere else in this paper. Every policy that spends its budget ends below the no-rescue control. All five arms consumed identical total optimizer tokens and the four spending arms consumed matched human budgets, so the separation between trajectories is not attributable to differences in training volume. Two features matter for the claims above: selection-only and joint stay within a line width of each other at every generation, as two arms that bought the same examples should, while every other pair separates visibly; and schedule-only crosses random between generations five and six, so the integrated outcome cancels part of a real endpoint difference. Generations are indexed 0 to 9.

total, 749,827 tokens, in every one of the five seeds, while the two shuffling arms vary by seed (Table 5). Second, of the 6,561 desired-spend sequences the urgency rule can produce over the eight base generations, exactly 1 reproduces that total under the mode ranking the selection-only chains establish, and it is the all-base sequence (yes); the 210 alternatives among the 13,120 sequences under a different top mode would require joint's mode ranking to differ from selection-only's at every generation of every seed, starting from a generation-0 model trained on the same corpus under the same seed. Third, the executed urgency is the reciprocal of the best-covered mode's lagged NLL, and leaving the band would require a validation NLL below 2.0 or above 4.0 nats, outside the range any held-out figure in this study reached. The joint arm therefore bought the same 152 examples as selection-only, in the same generations, and the primary contrast compares one allocation with itself.

Reconstructing what was bought. The per-generation purchase manifests of the executed grids were not archived. But every allocation rule is a deterministic function of the frozen rescue manifest, the chain seed, and the per-mode ranking the evaluator handed the policy, and the manifests and seeds are in the repository. We therefore re-execute the rules over the manifest and accept a reconstruction only when it reproduces the archived lifetime total of the chain to the token. For the two shuffling arms that is an exact check per chain. For the two ranking arms the mode ranking is not archived, so every per-generation top-mode sequence (19,683 of them) is enumerated and only those reproducing the total are kept; for joint, every desired-spend sequence its urgency rule can produce (6,561) is enumerated as well. Appendix D gives the procedure and the check ledger.

Table 5 reports the reconstruction of Section 5. Every random and schedule-only chain in all three model families reproduces its archived lifetime total to the token, thirty of thirty, confirming that the re-executed rules match the rules that actually ran. For selection-only on GPT-2, exactly 1 of the 19,683 top-mode sequences reproduces the archived total, and it is the sequence in which the tail mode ranks first at every generation; the same holds on SmolLM2-135M. On Pythia-160M the tokenizer's block sizes let 169 sequences land on the same total, the tail-always sequence among them, so that grid's selection trace is bounded rather than pinned.

That is what the executed rule produces: the tail mode had the lowest lagged validation NLL at every generation, every tail candidate carries that mode's score, and ties break by identifier, so the arm re-bought the same identifier-ordered prefix each time. The tail pool holds 433 candidates and 438,498 tokens, less than the 750,000-token budget, so a policy that concentrates on the tail must

Table 5: What each arm bought over the horizon, reconstructed by re-executing the allocation rules over the frozen rescue manifests and verified against every archived chain total (Appendix D). Presentations count optimizer presentations of human examples; distinct examples count each example once; max. repeats is the largest number of generations in which one example was bought. Ranges are over the five seeds for the shuffling arms and over the sequences that fit the total where the trace is not pinned. Joint is listed with selection-only where its totals are identical in every seed.

Model	Policy	Presentations	Distinct examples	Max. repeats	Modes bought
GPT-2	Random	175–206	171–200	3	all three, by shuffle
	Schedule-only	169–201	167–198	2	all three, by shuffle
	Selection-only and joint	752	152	9	152 tail
Pythia-160M	Random	176–206	172–200	3	all three, by shuffle
	Schedule-only	170–204	167–202	2	all three, by shuffle
	Selection-only	406–763	155–200	9	not pinned (169 sequences fit)
	Joint	lifetime totals differ from selection-only by seed (754,973–755,000); trace not pinned			
SmolLM2-135M	Random	181–203	176–194	3	all three, by shuffle
	Schedule-only	171–203	169–201	2	all three, by shuffle
	Selection-only and joint	732	148	9	148 tail

replay something; the executed policy replayed far more than the pool forces, since it touched 152 of the 433 tail candidates.

Why it is possible. Every allocation rule in Section 4 is a deterministic function of three inputs: the frozen rescue-candidate manifest, which carries each candidate's identifier, mode, and optimizer-token price; the chain seed, combined with the generation index; and, for the two ranking arms, the per-mode score the evaluator handed the policy. The manifest and the seeds are in the repository; the per-generation scores are not. The greedy fill takes candidates in the arm's order and buys each one that still fits, so a purchase sequence is fixed once the order and the allowances are fixed.

The check. A reconstruction is accepted only when it reproduces the archived lifetime human total of the chain to the token. The totals are the one purchase-side quantity every chain result carries, and they are sensitive: shifting one candidate changes a total by hundreds to thousands of tokens. For random and schedule-only the order depends on the seed alone, so the check is exact per chain, and it passes for all thirty chains across the three model families. That is also the check that the rules re-executed here are the rules that ran, since a different shuffle, seed convention, schedule, or terminal rule would miss the totals.

The ranking arms. Selection-only's order depends only on which mode ranks first at each generation, since every candidate in a mode carries that mode's score and ties break by identifier; whether a lower-ranked mode is ever reached depends on the allowance, and no allowance in this study exhausted the top mode. All $19{,}683$ per-generation top-mode sequences are enumerated over the nine spending generations. On GPT-2 and SmolLM2-135M exactly one reproduces the archived total, and it ranks the tail mode first throughout; on Pythia-160M 169 do, so that grid's trace is reported as a range. For joint, the desired spend at each of the eight base generations is one of $\{0, s_{\text{base}}, s_{\max}\}$, followed by the feasibility clamp and the terminal rule, and all $6{,}561$ such sequences are enumerated under the top mode the selection-only chains establish; only the all-base sequence reproduces the total on GPT-2 and SmolLM2-135M. The alternatives under a different top mode are counted and reported rather than dismissed.

What it does not recover. The reconstruction gives the purchase sequence, not the per-mode validation NLL that produced the ranking, so the statement that the tail mode had the lowest lagged NLL is an inference from the executed transform and the pinned ranking, not a reading of an archived score. It also cannot say whether joint's urgency came close to a threshold, only that it never crossed one. The check ledger, listing every chain's reconstructed and archived total and every matching sequence, is released beside the script.

E RECONCILIATIONS

This appendix does the arithmetic that Section 5 and Section 6 refer to. Every input is a number printed elsewhere in this paper.

What the equivalence region is anchored to. The verdict does not depend on which arm's mean supplies the denominator: taken against the strongest policy rather than against random the region is ± 0.0459, and the interval lies inside that too. It does depend on the *scale* the region is anchored to. Anchoring 2% to random's total regret makes the region ± 0.0507, which is about five times the pacing effect the study is trying to resolve and 14% of the whole control-to-best range. Anchoring the same 2% to the incremental gain that selection buys over random gives ± 0.0049, and the primary interval does *not* fit inside that. Against the preregistered region this is equivalence; against a region scaled to the effect being tested it is inconclusive. Appendix E gives both computations.

The two token ledgers. Candidates are tokenized into blocks of 512 tokens, and rescued human examples displace synthetic records one for one, so each generation's assembled corpus holds a constant number of records. If the human ledger also counted whole blocks, every arm's lifetime human total would be the same multiple of 512: the largest such total below the 750,000 ceiling is 749,568. The realised totals in Table 1 are 749,709 to 749,995, and none is a multiple of 512. The two ledgers therefore count different units. The human ledger counts the tokens of the purchased examples themselves; the optimizer ledger counts the fixed-size blocks those examples are packed into, padding included. Both statements in Section 6 are true under that reading, and neither is true under a single common unit: human spend is matched to a ceiling up to indivisibility, and total optimizer exposure is identical because the record count per generation is held fixed by displacement. Read as a fraction of optimizer exposure, the human budget is about 4.5% of the tokens the optimizer consumes over the horizon, which is the operating point every effect size below should be read against.

The preregistered region is 2% of the random arm's mean regret, which is ± 0.0507. Two other anchors are defensible and give different verdicts. Against the strongest arm's mean the region is ± 0.0459 and the primary interval still fits inside. Against the incremental gain that selection buys over random, which is the quantity an allocation study is trying to resolve, the region is ± 0.0049, and the primary interval $[-0.0092, 0.0297]$ does not fit inside it. The preregistered region is also about 14% of the entire measured range from the no-rescue control to the best arm. We report the preregistered verdict because it was preregistered, and we report this sensitivity because a region five times the size of the effect it screens for makes a null easy to obtain.

What five chains say about the variance. The paired standard deviation is 0.0156 on four degrees of freedom. A 95% chi-square interval for the underlying standard deviation is $[0.0093, 0.0448]$; the upper end is 2.9 times the point estimate. At that upper end the preregistered region spans only 1.1 standard deviations, so the design is not comfortably powered across the range the data leave open. The power statement in Section 6 is computed from the observed spread and therefore cannot certify itself. All chains share one fixed base subsample, so none of this covers corpus-draw variability.

Choosing the baseline from the outcome. The comparison baseline is the lowest-mean-regret non-joint arm, a rule fixed in advance but resolved by the data. Choosing the minimum of four noisy arms and then testing against it means the reported two-sided intervals do not have exactly their nominal coverage. We did not apply a formal post-selection inference correction such as polyhedral conditioning. We did simulate how often the rule picks a different arm, which is a different question and is reported below. The effect is likely small here, because selection-only beats the next arm by 9.59% of its own mean against a paired standard deviation of 0.0156, so the choice was effectively forced by the data rather than close; but likely small is not measured, and a selection-adjusted interval is the right fix.

Multiplicity. This paper reports one primary contrast and four secondary ones on the primary outcome, with confirmatory versions of each on tail retention and descriptive versions on final-generation regret. We applied no multiplicity correction, and we did not preregister one. The primary contrast is the only confirmatory test and is unaffected. The secondary contrasts should be read as descriptive: the two large ones, spending at all and choosing which examples to buy, are far enough

from the threshold that any conventional correction leaves them intact, while the schedule contrast is a null on the primary outcome and would only become more so. The contrasts this matters for are the tail-retention and final-generation pacing effects, which exclude zero on uncorrected intervals; across the number of contrasts reported here they should not be treated as confirmatory, and Section 6 does not treat them that way.

What the numbers mean in perplexity. Read off Figure 2, chains start near 3.18 nats (perplexity ≈ 24); the control rises to roughly 3.65 (≈ 38) and the best policy to roughly 3.54 (≈ 34.5). The whole experiment lives inside about four perplexity points at the final generation, and the primary null is a small fraction of that. These are figure readings, not tabulated values.

How much the baseline choice costs coverage. Section 6.6 notes that the comparison baseline is the lowest-mean non-joint arm, a data-dependent choice. We simulated it rather than asserting it is harmless. Drawing each non-joint arm's five-chain mean from a normal with the arm's reported mean and standard error and taking the argmin, selection-only is chosen in $200{,}000$ replicates out of $200{,}000$. The two best arms are separated by 0.2330 in outcome units, which is 16.5 standard errors of their difference, so the selection is effectively deterministic and the reported interval is an ordinary paired t interval with nominal coverage. This argument holds at the observed separation and would not hold if a future run brought two arms close together.

A coincidence we cannot explain. The joint-minus-selection-only and schedule-only-minus-random contrasts are equal to four significant figures and opposite in sign (0.0103 and -0.0103). It follows from the arm means in Table 1, so it is not a duplicated statistic in the code. We have no account of why the means fall that way, and flag it rather than let a reader find it.

The superseded reproducibility-floor argument. An earlier draft argued the two arms differed in allocation because their regrets differ by 0.0103, about 17 times a reproducibility floor of 0.0006 units taken from the positive control (arithmetic retained in Table 6). Two things settle it against that draft: the arms did not differ in allocation (Appendix D), and a single-step floor on another corpus does not bound a recursive chain's drift – the in-pipeline generation-0 spread is 0.00008–0.00054 nats and the gap grows to 0.0063 within the horizon.

Where every derived value comes from. The quantities below are not read from a chain artifact. Each is computed from numbers printed elsewhere in this paper, and Table 6 gives the arithmetic for each one so a reader can recompute it by hand.

F TWO FURTHER MODEL FAMILIES: A DESCRIPTIVE REPLICATION

After the GPT-2 grid was frozen, the same five policies were run on EleutherAI/pythia-160m (Pythia-160M) and HuggingFaceTB/SmolLM2-135M (SmolLM2-135M) over the same 400 WikiText-103 articles, the same five seeds, the same horizon of 10 generations, and the same training and decoding recipe. Each model's budget is the same *share* of its own total optimizer tokens rather than 750,000 tokens: a rule fixed before either run sets that share at 4.50%, the share GPT-2 realised, giving 755,057 tokens for Pythia-160M (1.007 times the GPT-2 count for the same text) and 799,022 for SmolLM2-135M (1.066 times). Rows compare across models as the same share, not the same text.

What holds and what does not. All 50 chains completed, and both grids match on the human axis: realised spend spans 754,789–755,044 tokens for Pythia-160M (0.034%) and 798,691–799,008 for SmolLM2-135M (0.040%), against 0.2% permitted. On the total axis they do not. These grids ran from the main branch, which *appends* purchased examples to the synthetic corpus; the displacing rule the GPT-2 grid used was never merged. Totals therefore spread 2.28% and 2.07% across arms, with the control training on the least data, and the spending arms differ from each other too, since arms concentrating on short tail articles pack differently into 512-token blocks than arms drawing across all modes. Only two contrasts have matched totals – joint against selection-only (-0.01% and 0.00%) and schedule-only against random (-0.02% and -0.02%). Every other row of Table 7 mixes allocation with training volume and is shown for completeness. Only a rerun under displacement removes this; no analysis can.

Table 6: Every derived quantity quoted in this paper, with its arithmetic. Inputs are the arm means of Table 1, the positive-control perplexities of Table 4, and the frozen configuration of Appendix A.2.

Quantity	Value	How it is computed
Incremental margin	0.0049	2% of the selection-only-vs-random gain: $0.02 \times (2.5364 - 2.2931) = 0.00487$
Reproducibility floor (superseded)	0.0018	positive-control generation-0 disagreement: $29.5903 - 29.5885$
in outcome units	0.0006	that gap in NLL, accumulated over the horizon: $10 \ln(29.5903/29.5885) = 0.00061$
observed gap over floor	17	observed joint-vs-selection gap over that floor: $0.0103/0.00061 = 17$
Tail ratio	250	$0.00766/0.00003 = 255$, quoted as "about 250"
σ interval, low	0.0093	95% chi-square interval on σ, $s = 0.0156$ on 4 df: $0.0156\sqrt{4/11.1433} = 0.0093$
σ interval, high	0.0448	$0.0156\sqrt{4/0.4844} = 0.0448$
σ ratio	2.9	$0.0448/0.0156 = 2.9$
Region in σ	1.1	$0.0507/0.0448 = 1.1$
Human fraction	4.5%	$749{,}869/(3280 \times 512 \times 10) = 4.5\%$
Largest whole-block total	749,568	$1464 \times 512 = 749{,}568$, the largest whole-block total below the ceiling
Urgency band in NLL	$(2.0, 4.0]$	reciprocals of the thresholds 0.50 and 0.25 under the executed score $1/\mathrm{NLL}$
Endpoint region	0.0088	2% of random's mean final regret, 0.02×0.4411

A second gap is procedural: partition-provenance manifests were not archived for either model (0 of 25 and 0 of 25), so the validator certifies none of these chains. The numbers come from the same code and the same archived chain results as the GPT-2 numbers, but no value from them enters the claims, and the abstract names the replication without reporting one.

What the two grids show. On SmolLM2-135M the GPT-2 pattern reproduces (Table 7, Figure 3): joint minus selection-only is -0.0021, 95% CI $[-0.0073, 0.0031]$ (0.11% of selection-only's mean), inside a ± 0.0413 region built as GPT-2's was; schedule-only minus random is 0.0026, CI $[-0.0137, 0.0189]$, containing zero; the same contrast on tail retention is small and excludes zero (0.00595, CI $[0.00401, 0.00789]$), as on GPT-2; and at the endpoint it is -0.0220, CI $[-0.0291, -0.0148]$ (6.29% of random's endpoint), again excluding zero. Coefficients of variation run 0.28%–0.70%. The purchase reconstruction reproduces the GPT-2 identity: joint and selection-only reached 798,946 tokens in every seed, the trace is pinned to the tail mode throughout, and the arm bought 148 distinct tail examples with 73 replayed 9 times (Table 5).

On Pythia-160M between-chain variation is an order of magnitude larger, 12.19%–38.75% against a 2% threshold, and no contrast resolves. Joint minus selection-only is 0.9052, CI $[-0.7680, 2.5785]$: the interval contains zero but reaches far outside the ± 0.1473 region, so the outcome is inconclusive rather than an equivalence. Schedule-only against random is likewise inconclusive (0.5773, CI $[-2.4010, 3.5557]$), as is the endpoint contrast (-0.1154, CI $[-0.5048, +0.2739]$). The variance is not seed variance: at generation 0, where all five arms train on one corpus under one seed, they disagree by 0.13371–0.59061 nats per seed, against at most 0.00054 on GPT-2 and 0.00018 on SmolLM2-135M. The Pythia-160M training stack is nondeterministic at a fixed seed on this hardware, and five chains at that variance cannot separate policies. Two facts follow from the reconstruction. Joint's lifetime totals differ from selection-only's by seed (754,973–755,000 against selection-only's total), so on this family the urgency signal did cross a threshold and joint did pace, which it never did on the other two; and 169 top-mode sequences reproduce the selection-only total, so its trace is bounded to 155–200 distinct examples rather than pinned. The direction of that pacing is not ambiguous, however: under the executed reciprocal score, spend falls to zero once the monitored mode passes 4.0 nats, so on the one family where the controller acted it withdrew human data as degradation worsened.

A rerun of these grids under the displacing rule, which needs only the merge of code the GPT-2 grid already ran on, would settle the total-token axis without new method work.

Table 7: Per-arm results on two further model families, generated from archived chain results by the same outcome definitions as Table 1, with final-generation regret beside the integral. Human tokens are matched within each model; total tokens are not, so only joint against selection-only and schedule-only against random are comparable on both axes. **Bold** marks the best value among the four spending arms of each model in each outcome column (lowest AUC and final regret, highest tail retention), shared on a tie at displayed precision; the control is not ranked because it spends nothing by construction. None of these chains is certified.

Model	Policy	Human tokens	Total tokens	AUC regret ↓	SD	Final regret ↓	Tail retention ↑
Pythia-160M	No rescue (control)	0	16,769,536	10.0660	1.9857	2.2566	0.6033
	Random	755,021	16,867,942	7.3662	2.8548	1.4596	0.7141
	Schedule-only	754,914	16,864,154	7.9435	1.5078	1.3442	0.7254
	Selection-only	754,977	17,160,192	**6.5168**	0.7945	1.0104	**0.8074**
	Joint	754,982	17,157,837	7.4220	1.5716	**1.0096**	0.8045
SmolLM2-135M	No rescue (control)	0	17,720,832	2.1529	0.0075	0.3797	0.8755
	Random	798,876	17,818,214	2.0636	0.0145	0.3492	0.8839
	Schedule-only	798,888	17,814,733	2.0661	0.0092	0.3273	0.8898
	Selection-only	798,946	18,095,616	1.8881	0.0082	**0.2923**	**0.9022**
	Joint	798,946	18,095,616	**1.8860**	0.0053	**0.2923**	0.9021

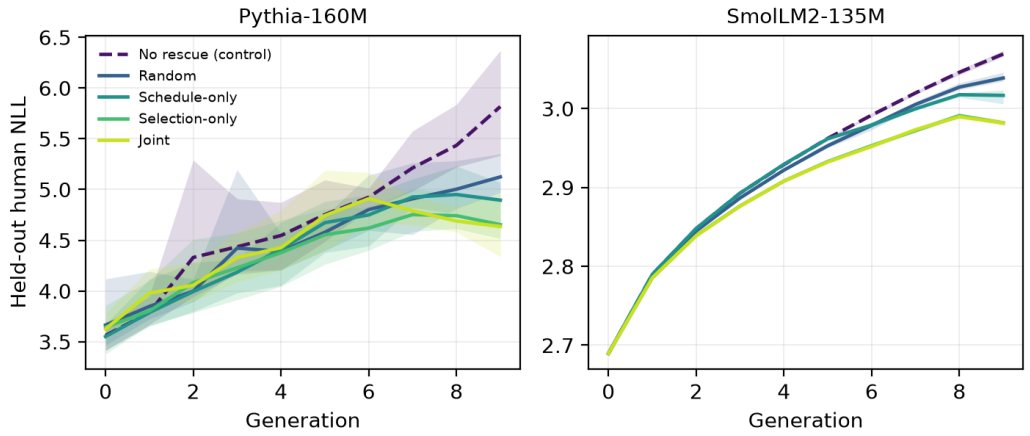

Figure 3: Held-out human NLL against generation on Pythia-160M (left) and SmolLM2-135M (right), mean over the five seeds with a min–max band; the control is dashed. The vertical scales differ, and the two panels share the single legend drawn on the left. On SmolLM2-135M selection-only and joint lie on top of one another as on GPT-2; on Pythia-160M the seed bands overlap for every pair of policies, which is the generation-0 spread this appendix reports rather than an allocation effect.

G Design choices and limitations

G.1 Design choices and what they bound

Six choices in the setup shape what the executed run can and cannot show. Each is stated here with the bound it imposes.

The integrated outcome disadvantages a pacing policy by construction. The primary outcome A^{π} is an unweighted trapezoidal integral over the horizon, so a schedule that is worse early and better late has those two effects partly cancelled algebraically before any estimator sees them. A back-loaded schedule is exactly such a policy. That is why the executed run returns a null on A^{π} and a contrast excluding zero on r_{G-1} for the same arm. A study whose hypothesis is about *timing* should pre-commit endpoint regret, or a discounted cumulative regret $\sum_g \gamma^g r_g$, as a co-primary outcome rather than reporting it descriptively as we do. We report the endpoint beside the integral for this reason, and it carries no multiplicity adjustment.

Budgeting by optimizer presentations does not enforce diversity. Section 1 motivates counting presentations so that a policy cannot look efficient by replaying a small archive. That accounting equalises token exposure and compute, and it does bound total presentations, but it does not prevent repetition: with a candidate pool that is never depleted, an arm may spend its whole budget representing a small set. Selection-only did exactly that. Preventing it requires pool depletion or sampling without replacement, neither of which the executed rule has, and the framing in Section 1 should be read as a statement about accounting rather than about diversity.

Re-initialising from a pinned checkpoint bounds the degradation it measures. Each generation trains from the same pinned pretrained checkpoint rather than continuing from θ_{g-1}. This isolates recursion through the data from optimizer drift through the weights, which is what the allocation question needs. It also re-injects the pretrained model's human linguistic representations at every generation, so the trajectories here are bounded well above what an unanchored self-consuming loop, or one fine-tuning continuously from its predecessor, would produce. Degradation magnitudes in this paper should not be compared directly against those settings.

Every outcome here is a likelihood. Both the primary and the confirmatory outcomes are held-out cross-entropy, one integrated and one as a tail ratio. Cross-entropy on a held-out slice of the same corpus can stay flat while generated text loses lexical and topical variety, and it says nothing about downstream capability. A study claiming that an allocation policy preserves what matters about a model would need diversity measures over generated text and at least one downstream benchmark alongside the likelihood. We report neither, so the claims here are claims about held-out likelihood on one corpus and should not be read as claims about capability retention.

The grid is not a balanced factorial. Section 4 presents two axes, but the two non-uniform arms are not the same kind of object: schedule-only runs an open-loop vector fixed before the run, while joint runs a closed-loop rule reading monitored state. Two cells of a full design are absent – an open-loop schedule combined with mode targeting, and adaptive pacing over randomly shuffled candidates. The executed comparison is therefore open-loop scheduling against closed-loop pacing, not a crossed 2×2, and Appendix G.5 states why joint is not the composition of the other two.

G.2 Limitations of Scope

The validation partition holds 3 tail articles and the test partition 10. A mode score computed from 3 documents carries a standard error large enough that a threshold rule reading it would be responding partly to document-level noise, and a confirmatory retention metric computed on 10 documents inherits finite-sample noise over a ten-generation trajectory. In this run the consequence was latent rather than active, because the executed rule never crossed a threshold; an actuated pacing rule on this partition would have been steering on that noise. The partitions came from an unstratified draw over the corpus, so the fix is to stratify by mode when drawing, not to change the policy.

Section 6.6 states the limitations in full and is not repeated here. The one figure this appendix adds is the model scale: 124M parameters, a screening-scale model whose behaviour cannot establish anything about larger scales, other domains, or other model families.

G.3 Further limitations of the primary run

These five limitations are summarised in Section 6.6 and stated here in full. They are limitations of the GPT-2 grid reported in the body, and are distinct from the scope limitations in Appendix G.2 and the run limitations in Appendix G.4.

Targeting is confounded with duplication. Selection-only ranks candidates deterministically within a mode while random shuffles, and the candidate pool is never depleted. The reconstruction in Section 6.3 makes this measurable rather than merely suspected: 75 tail examples were presented in all 9 spending generations and carry 89.8% of the budget, while random touched 171 to 200 distinct examples with at most 3 repeats. "Targeting helps" and "upweighting a small tail archive helps" are not distinguished by this design. A selection arm that removes bought examples from the pool, or that breaks ties with a seeded shuffle instead of by identifier, would separate the two effects and would require no new method – only a change to the tie-breaking rule.

The pacing contrast (schedule-only against random) is real, but it is the only one we ran. Random and schedule-only share seeded candidate orderings and differ only in the per-generation allowance, so under a uniform schedule they would be identical by construction; the project's failure log once recorded them as such under a fixture schedule and closed the entry when the frozen back-loaded schedule was restored. The executed schedule is published in Table 2, and the two arms reconstruct to the token with different purchase sets, confirming the contrast reflects a real allocation difference rather than an artifact. It is also the only non-uniform schedule tested, and it is back-loaded.

The comparison baseline is chosen from the outcome. The rule is preregistered but resolves against the data, so the reported intervals do not have exactly their nominal coverage. Appendix E argues the effect is small here and notes that we did not measure it.

Five seeds bound the variance loosely, and doubling them was affordable. The paired standard deviation is estimated on four degrees of freedom, and its interval reaches 2.9 times the point estimate (Appendix E). The grid cost 8.53 hours on four accelerators. Doubling the seed set would be the least expensive way to strengthen the equivalence argument, and we did not do so before this submission. It would not repair the Pythia-160M grid: there the five arms already disagree by 0.13371 to 0.59061 nats at generation 0, where they train on one corpus under one seed, against 0.00054 on GPT-2 and 0.00018 on SmolLM2-135M. That family's 12.19%–38.75% between-chain variation is run-to-run nondeterminism at a fixed seed, and more seeds would average it rather than remove it; a determinism audit of that training stack is the necessary first step.

The policies we did not build, and what each one costs the claim. Three of seven planned comparison policies are absent, and each would change what the study can conclude. An accumulation or fixed-fraction arm under matched optimizer steps would say whether budgeted rescue beats simply keeping prior data (Gerstgrasser et al., 2024; Kazdan et al., 2025). A distribution-matching or worst-group selector – importance resampling toward a target (Xie et al., 2023b) or a group-regularised objective (Sagawa et al., 2020) – would say whether our length-quantile heuristic captures anything those do not. A detector-guided resampling arm (Drayson et al., 2025) or a self-correcting loop (Gillman et al., 2024) would say whether spending human tokens beats filtering the synthetic side, which is the alternative a deployment actually weighs. "Strongest eligible policy" therefore means strongest among the four we ran. Two absences matter most. Without the oracle, an equivalence between two policies does not indicate how much of the achievable gain either captures, so no reader should infer that selection-only sits near a ceiling. And we ran one non-uniform schedule, back-loaded, when our own motivation in Section 1 identifies front-loading as the interesting direction; neither a front-loaded schedule nor a genuinely adaptive pacing-only arm was built. The timing null therefore applies only to the one fixed schedule we tested, not to timing as a general question. This was a scoping decision rather than a technical constraint: a front-loaded schedule arm would have cost about the same as any other five-seed arm in this study.

G.4 Limitations of the executed run

The modes are constructed, not given. The map assigns candidates to 3 modes by article length, with the tail at $\leq 1{,}106$ words. Training segments text into 512-token blocks, so document length is a property of Wikipedia curation rather than of what the optimizer consumes, and a length quantile need not correspond to any semantic sub-distribution; the protocol requires only that the partition be auditable and fixed without test access, which length satisfies without making it the right partition. More generally, clusters or labels can merge distinct subpopulations, split a coherent one, or omit an important tail, so a targeted policy can be efficient against its monitor while harming the full distribution. The monitoring-omission intervention probes one form of this but cannot enumerate all misspecification, and mode-level statistics are noisy at small sample sizes: the validation partition holds 3 tail examples, 26 mid and 31 common, so the tail score that ranked every purchase rests on three articles.

Equal tokens is not equal cost. Counting optimizer-consumed tokens is more auditable than counting documents, but equal counts do not imply equal informational value, annotation cost, li-

censing cost, duplication, or compute. This compares four allocation rules at equal token cost, a narrower question than which human-data policy is best under any economic accounting.

These limitations attach specifically to the executed run.

- **A null is not a demonstration of inferiority.** The interval lies inside the practical equivalence region, which licenses the statement that joint allocation does not improve on the strongest policy by a practically meaningful margin, and licenses no statement that it is worse.
- **The earlier failure was fixed structurally, not diagnosed.** Why the additive assembly rule made totals diverge across arms is still unestablished: that run recorded chain totals but not per-generation token accounting, so block packing and divergent generated text remain indistinguishable as causes. Displacement removes the divergence by construction and is validated to that extent, but it is a remedy adopted without the diagnosis.
- **We know what each policy bought and when, by reconstruction rather than by record.** The archived artifacts record lifetime human consumption but not the per-generation, per-mode allocation, which lived only in chain checkpoints released with the compute. Appendix D recovers the sequence from the frozen manifests and the archived totals, and the recovery is exact for the shuffling arms and pinned to a unique sequence for the ranking arms on GPT-2. What it recovers is the purchase; the per-mode validation scores that drove the ranking are not recoverable, so the statement that joint's urgency stayed in band is an inference from the totals and the executed transform, not a reading of the trace.
- **The effect threshold has not been externally reviewed.** It was preregistered before outcomes existed and checked against measured anchors, but never had the independent statistics review the project's own decision log calls for. Because the primary finding is an equivalence, it rests on that threshold more directly than a positive finding would.

G.5 Why joint is not simply the other two composed

Composing schedule-only with selection-only does not produce joint. A tuned schedule paired with mode-aware selection still gives a fixed spending profile: the schedule is still chosen in advance and cannot react to what monitoring reports at run time. Joint is the only policy whose spend at generation g is a function of the monitored state at generation g. Under a fixed lifetime budget this is a reallocation rather than an increase: tokens withheld early stay available, and the clamp's lower bound prevents stranding by keeping the unspent balance within what the remaining generations can absorb. In the executed run the monitored state never left the base band, so the function was constant and joint coincided with selection-only (Section 6.3); the distinction is real in the rule and inert in the data.

H Related work, extended

H.1 The standard our controls do not meet

The scheduling and mixture literature sets a methodological standard we state and do not meet: a pacing-only policy should get the strongest state signal available, and a selection-only policy the strongest fixed schedule. Our schedule-only arm reads no state signal, and our selection-only arm runs a uniform schedule we do not argue is strongest. Only the residual contrast against controls meeting that standard would speak to the value of coupling the two decisions; ours speaks to a weaker comparison.

H.2 Selection and tail-preservation literature

The compressed paragraph in Section 2 is stated here in full.

Which examples to buy has its own literature: density-ratio resampling (Xie et al., 2023b), gradient-contribution importance sampling (Katharopoulos & Fleuret, 2018), prediction-history weighting (Chang et al., 2017), core-set (Sener & Savarese, 2018) and gradient-embedding (Ash et al., 2020) active selection, and, in the recursive setting, detector-based resampling of likely human text

(Drayson et al., 2025). None of these selectors is claimed as novel here; our mode score is pre-registered and allowed to fail. Because average loss hides degradation on rare groups, we also take a tail-retention outcome, following worst-group optimisation (Sagawa et al., 2020), high-loss upsampling (Liu et al., 2021), hidden-subclass discovery (Sohoni et al., 2020), and long-tail logit correction (Menon et al., 2021). That literature carries a warning we build on: a targeted policy observes only its monitor, so if the monitor omits a globally important mode the policy can look locally efficient while making held-out performance worse. Our monitoring-omission intervention is a falsification test for that failure.

I Reproducibility Appendix

Every value reported in this paper is generated by a script and traced to an archived artifact. No number is entered by hand.

- **Positive control.** Upstream repository and exact commit are pinned before execution, together with the resolved framework versions and tokenizer revision actually used. Deviations from the upstream configuration are declared before running, never after observing a result.
- **Run artifacts.** Each chain emits a run manifest and a chain result validated against versioned JSON schemas. A three-state validator classifies each run as valid, valid with limitation, or invalid; blocking failures include budget-equality violations, identity mismatches, and impossible token accounting.
- **Budget matching.** Total optimizer tokens are identical across every policy in a comparison, by construction: rescued human examples displace synthetic records rather than being appended, so each generation's training volume is fixed regardless of what a policy buys. Lifetime human-origin tokens are matched as a common ceiling reached up to indivisibility rather than as exact equality, since candidates are indivisible and a policy cannot spend a fraction of one; the realised spread is reported in Section 6 and is an order of magnitude below the practical effect threshold. Both axes are checked before launch from configuration and again after the run from emitted artifacts.
- **Token accounting.** Counts are denominated in tokens consumed by the optimizer rather than in documents. Repeated presentation of an example costs its tokens once per presentation; gradient accumulation counts every micro-batch; resume neither double-counts nor drops the batch spanning a checkpoint. These properties are asserted by the accounting tests on fixtures. On the executed run they are *not* recomputed from realised batch records, which were not retained: the declared ledger is checked against the pre-committed budget instead. Every chain carries this as a recorded limitation rather than as a silent assumption.
- **Purchase reconstruction.** The per-generation purchase sequence of every arm is rederived from the frozen manifests and seeds by the released script and checked against every archived chain total (Appendix D); the script exits non-zero on any mismatch, and every purchase figure in this paper is emitted by it.
- **Determinism.** Seeds propagate through sampling, generation, initialisation, dropout, and evaluation. Checkpoints carry content digests so silent corruption cannot survive a resume. Bitwise reproducibility across runs is not claimed: the spread across the five arms at generation 0, where they share a corpus and a seed, is 0.00008 to 0.00054 nats on GPT-2 and 0.13371 to 0.59061 nats on Pythia-160M.
- **Generated outputs.** Tables and figures are emitted by scripts and recorded with SHA-256 digests in an artifact manifest, so a reader can verify that a figure is the one the pipeline produced.

J Registration status, item by item

The paper's equivalence argument leans on things having been fixed in advance, so this appendix states what the project repository actually records for each, with dates, rather than leaving a reader to assume a uniform standard. The first chain of the executed grid ran on 2026-08-19.

Preregistered: frozen in the method freeze of 2026-07-31. The four allocation rules of Section 4, including the thresholded joint rule with its feasibility clamp and the thresholds 0.25 and 0.50; the back-loaded schedule shape, zero for the first half of the horizon and the cap thereafter; the mode-score formula, a clipped lagged relative NLL gap; the ordered seed set and the replacement seeds; the primary outcome, NLL-regret AUC; the tail-retention family for the confirmatory outcome; the 2% relative practical-effect threshold; the exclusion rules; and the rule selecting the strongest eligible non-joint baseline. These were frozen with fixture-scale budgets, and the budget scale itself was not among them.

Pre-committed: fixed in configuration between the freeze and execution. The domain and the mode definition, `article_length_quantile` with its thresholds, frozen on 2026-08-08; the ratio-based tail-retention metric over the NLL-gap alternative, frozen the same day; the retraining regime, the four-hundred-article subsample, the lifetime budget of 750,000 tokens with base B/G and cap $2B/G$, the horizon of 10, and the total-token figure, frozen on 2026-08-18 by the project owner rather than by a team vote, on written evidence and with a stated reversal condition; displacement as the corpus-assembly rule, adopted on 2026-08-19 after the first grid's totals diverged. The 2% threshold was re-examined against the first grid's measured anchors on 2026-08-19 and kept; it has still not had the independent statistics review the decision log calls for.

Executed differently from the freeze. The mode score was not implemented as specified in the preregistered plan: the runner supplied the reciprocal of each mode's lagged validation NLL rather than the planned relative-gap score. Both ranking arms used the supplied score to rank candidates. Section 6.3 and Appendix A.3 state the consequences: the top-ranked mode was the lowest-NLL mode rather than the most-degraded one, and joint's band corresponds to an NLL range the run never left.

Recorded as open and later closed. The failure log once recorded random and schedule-only as selecting identically. That entry described a fixture with a uniform schedule, produced by a revert of the frozen policy code, and was closed when the frozen back-loaded schedule was restored; the executed arms reconstruct to the token with different purchase sets (Table 5). The schedule vector is built by the run script from the frozen shape and the frozen cap, and Table 2 prints it.

Not registered. There is no timestamped external registration to cite; the plan document in the repository is marked a draft and says so. The final-generation regret of Section 6 and the purchase reconstruction of Appendix D were added after the outcomes were open, in response to review, and are labelled descriptive throughout.

Why this is stated rather than smoothed over. Three of these items sit directly under load. The threshold decides the equivalence verdict. The mode score decides what targeting targeted, and it was not the frozen one. The schedule decides whether the timing contrast is a contrast at all, and it is now printed. Publishing the executed configuration alongside the run was the single cheapest thing that could raise the paper's evidential standing, and this appendix and Appendix A.3 do it.