MEASURING DAMAGE, NOT LEARNING: STEP-0 BASELINES FOR CONTINUED PRETRAINING

Anonymous authors
Paper under double-blind review

ABSTRACT

Data-recipe comparisons in continued pretraining share one design: continue from one checkpoint under competing rules, rank by held-out loss. Our case is latent-augmented pretraining, which inserts model-generated reasoning into training text *uniformly*: could a per-chunk signal place it better? We ran that comparison carefully — twelve conditions, up to three seeds, two domains, two model scales, a frozen truncation-safe metric — and obtained a clean, seed-stable ordering: uniform beat an equal-budget random control by 0.49 nats/token, and demand-based routing lost to that control by 0.21. Then we scored the warmstart itself. **Every condition, uniform included, is worse than the checkpoint it started from**, so the ordering ranks damage, not learning.

A tenfold lower learning rate removes 92% of the headline contrast; with a warm-start that privileges no condition by density, both signs survive at 7–14% of their magnitude. Only replay plus a low rate puts runs back below their start, and **there uniform loses its lead**: random routing is ahead of it, significantly only when uniform alone matches the warmstart density, and no signal beats random. Across three public replications that share no code with ours (216 runs), the learning-rate and two-axis step-0 checks reproduce every time; weight displacement, which tracked our ranking at 0.94, does not generalise. The mechanism, distribution shift and learning-rate annealing, is known (Wang et al., 2025b); what is new is a documented case in which a ranking stable across seeds, domains and scales was a property of the recipe rather than of the allocations. Separately, length-corrected reasoning demand, a posterior-to-prior KL from a fixed model, predicts per-chunk latent benefit (ρ=0.48; incumbent signal 0.14).

Of twenty-five published continued-pretraining data-recipe comparisons, most report the start, but loss-ranked ones more often as a figure point than as a number to difference against. Step 0 on two axes and a rate check cost an evaluation each; we argue they belong in the standard protocol.

1 INTRODUCTION

Latent-augmented pretraining inserts model-generated reasoning traces into training text and trains on the augmented sequence, bootstrapping the model's own reasoning via an EM-style loop. BoLT (Ruan et al., 2025) and thinking-augmented pretraining (Wang et al., 2025a) augment *uniformly*: every chunk receives a latent thought, regardless of whether it benefits. A parallel line *learns* where to spend latent compute (Zhu et al., 2025; Zeng et al., 2026), and others build the latent structure into the architecture or objective (Kong et al., 2025; Zhou et al., 2026a); none measures per-chunk benefit directly. This raises a natural efficiency question — the same question that motivates data selection (Xie et al., 2023b;a) and adaptive compute (Raposo et al., 2024): if latents are expensive, should we not concentrate them where they help most?

To build such a signal we define **reasoning demand** D_i as the KL divergence between a chunk's posterior reasoning distribution (conditioned on the full chunk) and its prior (conditioned on the prefix alone) — the Bayesian surprise (Itti & Baldi, 2006) in latent space. Demand is an information-theoretic, length-aware measure of how much seeing a chunk's continuation changes the model's reasoning.

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

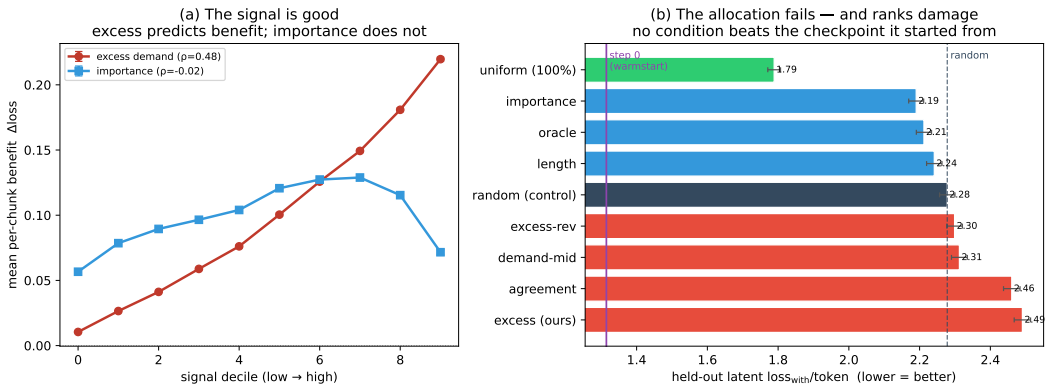

Figure 1: The reversal. *(a)* A length-corrected reasoning-demand signal predicts the per-chunk training benefit of a latent (Δloss; Spearman ρ=0.48), while the importance weight used by prior routing is weakly informative at best ($\rho = -0.02$ raw, 0.14 with length partialled out); mean benefit rises monotonically across the signal's deciles. *(b)* Yet allocating the latent budget by that signal is *worse than an equal-budget random control* on math, and every demand-based condition (red) lies right of random; uniform (all latents) has the lowest loss. The purple line is the warmstart all of them started from — *every* bar lies to the right of it, which is why Section 4 reads this panel as a ranking of damage rather than of learning, and why these magnitudes are the recipe-inflated first-pass values corrected in Table 1. **A signal can predict per-chunk benefit well and still be worse than random at allocation — and the comparison that shows it can be measuring something other than learning.**

Our starting hypothesis was that such a signal should beat uniform allocation under a fixed budget. It does not, and the more useful finding is why our first answer to that question was wrong by an order of magnitude.

Section 2 characterises where demand settles under the EM bootstrap; Appendix A turns demand's EM trajectory into a null-calibrated check for collapse; Section 3 shows that a length-corrected variant predicts the measured per-chunk benefit of a latent at $\rho = 0.48$ while the importance weight used by prior routing reaches 0.14 at best. Those three results are computed from a fixed model scoring a fixed corpus, and nothing later in the paper disturbs them.

Section 4 is where the trouble is. Comparing allocation rules by continued pretraining from a shared warmstart — the standard design — gave us a clean result: uniform beat an equal-budget random control by 0.49 nats/token, demand-routing lost to that control by 0.21, both significant across three seeds and reproduced at 2.5× model scale (the first also on a second domain; the second does not transfer, Section 4). We then scored the warmstart itself, which the design does not require and which we had not done. Every condition was worse than it. The comparison had been ranking how much each allocation *damages* a checkpoint, and we had been reading it as how much each teaches.

Two inexpensive checks separate the two readings, and we recommend both as defaults. Scoring step 0 costs one evaluation — for us 4.5 GPU-minutes against 12 GPU-hours of training — and answers whether training helped at all. A learning-rate check is the second (rankings' sensitivity to the rate is known (Wang et al., 2026)): lowering the rate tenfold removed 92% of our headline effect. (Rank-correlating $\|\theta - \theta_{\text{warmstart}}\|$ with the loss gave 0.94 here, but its sign did not generalise across public replications, so we report it as an observation, not a check.) Rebuilding the experiment with a warmstart that privileges no condition by density and a lower rate leaves both pivotal signs intact at 7–14% of the original size, and no single alternative recipe we tried lets training improve the checkpoint; 50% replay shrinks the damage by 88% and, with the low rate, is the one setting that does.

The contribution is therefore threefold: reasoning demand as a per-chunk predictor of latent benefit, with an identity that says what the signal measures (benefit plus inference error), though not how well it will predict; a corrected and much smaller estimate of what per-chunk allocation buys; and a documented case in which the standard design for this question, run without a step-0 baseline, produced a stable ranking whose winner does not keep its lead under recipes where training helps. The last is what we would most like the field to take from this paper: the design is common and the baseline nearly free.

2 REASONING DEMAND

Setup and definition. A chunk is $x_i = (P_i, S_i)$, a prefix and a suffix; a latent thought is a token sequence z; the model defines a prior $p_\theta(z \mid P)$ and a posterior $q_\theta(z \mid P, S)$.

Definition 1 (Reasoning demand)**.** $D_i = D_{\mathrm{KL}}(q_\theta(\cdot \mid P_i, S_i) \,\|\, p_\theta(\cdot \mid P_i))$*, the Bayesian surprise (Itti & Baldi, 2006) in latent space: the information gained about the model's reasoning by observing the suffix.*

With K posterior samples $z^{(k)} \sim q_\theta(\cdot \mid P_i, S_i)$ we use the Monte-Carlo estimator $\hat{D}_i = \frac{1}{K}\sum_k[\log q_\theta(z^{(k)} \mid P_i, S_i) - \log p_\theta(z^{(k)} \mid P_i)]$, unbiased before the clip at 0 that we apply; its resampled-rank stability is $\rho = 0.97$ at $K{=}8$.

Definition 2 (Excess demand)**.** $\tilde{D}_i = D_i - \hat{f}(\ell_i)$*, where* $\hat{f}(\ell) \approx \mathbb{E}[D \mid \ell]$ *is an isotonic regression of demand on suffix length.*

Where demand settles. At the EM bootstrap's optimum $\mathbb{E}[D]$ equals the conditional mutual information $I_{q_t}(z; S \mid P)$, a non-negative floor that is zero only if the posterior ignores the suffix; this is a reading of the ELBO-surgery identity (Hoffman & Johnson, 2016; Tomczak & Welling, 2018; Alemi et al., 2018), nothing later depends on it, and Appendix B states it with its assumptions.

3 EXCESS DEMAND PREDICTS PER-CHUNK BENEFIT

We test whether excess demand predicts the benefit of a latent to a *fixed* model: per chunk, the held-out suffix-NLL improvement (Δloss, without the latent minus with it), scored on an *identical* with/without span (Section 4) so the comparison is free of truncation artifacts. On the full 200K-chunk corpus ($K{=}4$ posterior samples per chunk rather than the diagnostic set's $K{=}8$, so the estimator is noisier; that would attenuate the correlation if its error were independent of the benefit's, which we have not shown), with doc-cluster bootstrap 95% CIs:

$$\rho(\tilde{D}, \Delta\text{loss}) = 0.48\ [0.479, 0.489], \qquad \rho(\ell, \Delta\text{loss}) = -0.16,$$
$$\rho(-\log w, \Delta\text{loss}) = -0.02\ (\text{partial } 0.14).$$

Excess demand is a strong per-chunk predictor; the importance weight prior routing uses ($w = p(S \mid P, z)\, p(z \mid P)/q(z \mid P, S)$, BoLT's bootstrap weight) is weak — -0.02 raw, 0.14 with length partialled out — and mainly tracks length. Δloss is a fixed model's prediction benefit from a latent, not the training utility of the chunk; Section 4 tests whether the first predicts the second, and it does not.

What demand is, exactly. Before reading $\rho = 0.48$ as a discovery it is worth writing down what the two quantities are. For the exact Bayes posterior $q^\star(z \mid P, S) = p(z \mid P)\, p(S \mid P, z)/M(S \mid P)$, with $M(S \mid P) = \sum_z p(z \mid P)\, p(S \mid P, z)$ the latent-marginal likelihood, taking logs gives, *pointwise in* z, $\log q^\star(z \mid P, S) - \log p(z \mid P) = \log p(S \mid P, z) - \log M(S \mid P)$, so for a general amortized q

$$D_i \;=\; \underbrace{\mathbb{E}_q[\log p(S \mid P, z) - \log M(S \mid P)]}_{\text{expected latent benefit}} \;+\; \underbrace{D_{\mathrm{KL}}(q_i \,\|\, q_i^\star)}_{\text{amortization gap} \,\geq\, 0}. \tag{1}$$

The benefit we *measure* is not quite the first term: $\Delta_i(z) = \log p(S \mid P, z) - \log b(S \mid P)$ scores the suffix against the model's direct no-latent probability b, not the marginal M, so $D_i - \mathbb{E}_q[\Delta_i] = D_{\mathrm{KL}}(q_i \| q_i^\star) + \log[b(S \mid P)/M(S \mid P)]$ and the last term has no fixed sign. Two consequences. The identity says demand is benefit *plus inference error*; it does not fix the sign of their correlation — $\mathrm{Cov}(D, \Delta) = \mathrm{Var}(\Delta) + \mathrm{Cov}(\text{gap}, \Delta)$, and the second term is unconstrained — so $\rho = 0.48$ is an empirical fact about how the gap co-varies with benefit, not algebra. And the residual we can compute is the gap only up to $\log(b/M)$, which we have not scored: on the 200K corpus $\mathbb{E}[\hat{D}] = 51.84$ nats against $\mathbb{E}[\Delta] = 13.24$, so 74.5% of demand is an arithmetic residual (< 0 on 1.5% of chunks). It mixes four terms we cannot separate: the amortization gap, $\log(b/M)$, the difference between $\mathbb{E}_q[\Delta]$ and the benefit of the one stored latent (which the bootstrap resampled toward $q^\star$), and Monte-Carlo error in $\hat{D}$. A marginal term of $+0.07$ nats/token, the size of the whole measured benefit, would halve it, so we give it no interpretation. The same substitution accounts for the incumbent signal's

weakness: $-\log w = \log[q/q^\star] - \log M(S \mid P)$ pointwise, so the benefit term cancels and what remains is baseline suffix difficulty — $\rho(-\log w, \text{loss}_{\text{without}}) = 0.89$ against $\rho(-\log w, \Delta) = 0.20$ on the unnormalised total target (the stored weight is normalised per suffix word). The contrast in Figure 1a is consistent with what the two signals are (`results/demand_identity.json`).

Four disclosures about this measurement. First, the target is Δloss *per token*, and the choice matters: against the unnormalised *total* the picture inverts — $\rho(\ell, \Delta\text{loss})$ becomes $+0.13$ rather than -0.16 and raw demand (0.530) overtakes excess demand (0.513) — so the length-correction motivation is conditional on the per-token target, which we use because a latent's value should be measured per unit of context. Second, the isotonic machinery buys less than it appears: dividing demand by suffix token count gives $\rho = 0.458$ and by word count 0.429, against excess demand's 0.484 (residualised on word count) and raw demand's 0.348; residualising on *latent* length is unnecessary ($\rho(D, \text{latent length}) = 0.222$ falls to 0.072 for $\tilde{D}$). Third, and least flattering, partialling length out of raw demand gives $\rho = 0.488$, marginally *above* 0.484, so the isotonic construction's value is in the CMI reading, not in the measurement. Fourth, 25.9% of latents hit the 400-token generation cap, where the truncated KL understates the full one in expectation; on the uncensored subset (n=132,859) $\rho = 0.4836$ against 0.4840 overall. Excess and oracle top-50% sets share 68% of members (Jaccard 0.52); on the *signal* axis excess demand is clearly the best per-chunk predictor (Figure 4, Appendix E).

4 The Allocation Experiment, and What It Actually Measures

The measurement result licenses the obvious hypothesis: route the latent budget to high-excess chunks and the trained model should improve. We test it with the design this literature uses — and then test the design itself.

Protocol. We compare allocation conditions by continued pretraining from a shared warmstart checkpoint with a matched 600-step budget, `latent_type=routed` so a relabeled corpus trains directly, and leakage-free held-out evaluation (3000 chunks drawn from *unselected* documents; the scorer consumes documents in order until the chunk budget is reached, so the 3000 chunks come from 370 distinct documents, with one document contributing 192 of them). Because chunks within a document share nested prefixes, all Phase-4 intervals below are *document-clustered* bootstrap CIs (resampling documents, not chunks); the i.i.d. intervals we originally computed are 1.3–$3.1\times$ too narrow. Three kinds of interval appear, each labelled: *document-clustered* (over held-out documents, one run), *across-seed* t (over training runs, evaluation set fixed) and *two-component* (both); only the last two say whether another run would agree. The cluster map must replicate the scorer's chunk filter (a document's first chunk is never scored), a mistake we made and corrected. Significance is tested at the *document* level (a clustered bootstrap of the paired mean difference, with per-document paired t and signed-rank tests alongside), Holm-corrected within each per-seed family; every contrast we call significant survives at $p < 0.01$ (largest adjusted p, 1.4×10^{-3}). The chunk-level Wilcoxon p-values we first reported ignored the clustering and remain in the artifacts for comparison only. The decisive metric is held-out latent-aware suffix NLL/token (lower is better), scored over an *identical* with/without span per chunk — the *common-span* metric: both prompts are truncated to the same suffix token span and the same kept prefix, differing only by the latent segment, so all 3000 chunks enter the comparison free of truncation bias (under the naive drop-prefix policy the latent overflows the window on long chunks, leaving only a biased short-chunk subset).

The first pass, and why we believed it. Table 4 (Appendix D) reports twelve conditions under this protocol. The ordering is clean: `uniform` is lowest at 1.789, the equal-budget random control sits at 2.278, and `excess_50` is worse than that control by 0.201 pooled over three seeds. Every contrast is significant under a document-clustered bootstrap; the ordering reproduces across three training seeds and at 2.7B parameters, and uniform's lead reproduces on FineWeb-Edu; the metric is frozen and truncation-safe. We took it to mean that uniform coverage teaches more than any per-chunk rule, and that demand-routing actively hurts — and we could defend that reading against every objection we knew how to raise, because each of those checks asks whether the ordering is *stable*, and none asks whether it is about learning.

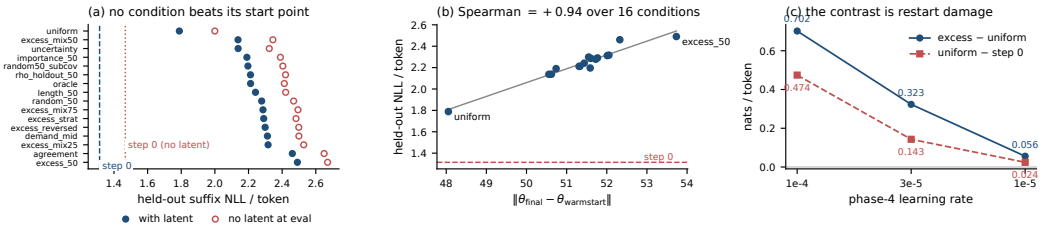

Figure 2: The allocation comparison ranks damage. *(a)* All sixteen conditions on the metric they were compared with (filled) and on the latent-free half of the same evaluation (open); the dashed lines are the warmstart's score on each. Every condition is worse than its start point on both, so the ordering is not a latent-format artifact. *(b)* Here the ordering tracks displacement: distance travelled from the warmstart rank-correlates with held-out loss at $+0.94$ (inconsistent across public grids). *(c)* Dropping the phase-4 learning rate tenfold removes 92% of the headline contrast and 95% of the damage, which is consistent with the contrast being mostly restart damage.

All conditions degrade the warmstart. Scoring the warmstart checkpoint itself on the same metric, with no phase-4 training at all, gives 1.314 — better than every condition in Table 4, including uniform (1.789; paired difference $+0.474$, document-clustered CI $[+0.440, +0.510]$, worse on 98.6% of chunks). The same holds at 2.7B (warmstart 1.299 vs. uniform 1.804). **Six hundred steps of continued pretraining at this recipe makes the model worse on this metric under every allocation we tried**, so Table 4 ranks damage, not teaching (Figure 2a). Nor is it a property of the latent-aware metric: on the *latent-free* half of the same evaluation (suffix NLL given the prefix alone) the warmstart (1.467) again beats all sixteen conditions we trained (named in `results/delta_theta_audit.json`), every document-clustered interval excludes zero, and the two metrics rank the conditions at $\rho = 0.99$. The degradation is in the language model, not in its handling of the latent format.

The comparison also crosses a corpus boundary: the warmstart was trained on the GPT-seeded latents BoLT uses to *initialise* its bootstrap, every phase-4 condition on the EM iteration-1 corpus (the 75% redesign resamples the GPT-seeded source). Iteration-1 latents might therefore simply be weaker training data; we tested it directly. Continuing from the warmstart on the *GPT-seeded corpus it was itself trained on* — where provenance cannot differ — still degrades it, $1.314 \rightarrow 1.365$. Corpus provenance therefore does not explain the step-0 result. Two further facts point the same way: the held-out set comes from the same iteration-1 cache as the training data, so its latents are out-of-distribution for the warmstart and in-distribution for its competitors, and the latent-free rescoring uses no latent at all and still gives the warmstart sixteen wins out of sixteen.

Two features of the recipe could produce this, and neither is a property of any allocation. First, *the warmstart was trained with* `latent_type=random`*, which inserts a latent into every chunk* — so `uniform` is the only phase-4 condition whose data distribution matches what the model had just been trained on, and every routed condition introduces distribution shift proportional to how far its selection departs from "all chunks". Second, the warmstart ended a cosine schedule annealed to near zero over 4069 steps, and phase-4 restarts at lr$=10^{-4}$ with fresh warmup; restarting a converged checkpoint at full learning rate perturbs it, and 600 steps is not enough to recover.

Together these are also consistent with the other anomalies we report: that held-out loss degrades monotonically with training steps at fixed allocation, that 96% of the between-condition span survives with no latent at evaluation, that test-time latent benefit is nearly constant across conditions, and the drift correlation below. **We therefore do not claim that uniform allocation trains a better model.** Separating an allocation effect from these two artifacts requires a warmstart whose latent density coincides with no tested condition, and a learning rate that does not restart a converged checkpoint at full scale. We ran both (below).

The corrected experiment. The two confounds above — a warmstart whose 100% latent density only `uniform` matches, and a fully annealed cosine restarted at 10^{-4} — are both properties of the recipe, not of any allocation; the second is also where we departed from BoLT, whose continual setting carries model and optimizer forward from a pre-decay checkpoint at a constant 3×10^{-5} with no re-warming (Ruan et al., 2025). We changed both and re-ran the pivotal conditions. The warmstart

Table 1: The same three allocations under three recipes (common-span metric, n=3000, 600 steps, lr=10^{-5}). *(a)* The corrected experiment: 75%-density warmstart, no replay — every condition ends above its step-0. *(b)* 50% replay of the warmstart's own corpus from the 100%-density warmstart; `uniform`'s mixture (100% latent) is not density-matched to the 50% rules (75%). *(c)* As (b) from the 75%-density warmstart, so every mixture is equidistant from its start (87.5% vs. 62.5%). In (b) and (c) every run improves on step-0; "50% replay" is a per-document source weight, which gives the 50% rules 53–57% replay tokens against uniform's 43–48%. Losses are means over seeds; "vs. random" intervals are across-seed t in (a) and two-component (document + seed) in (b, c).

	condition (seeds)	loss/tok ↓	vs. step-0	vs. `random_50`
(a)	*step 0*	*1.357*	—	—
	uniform (3)	**1.382**	+0.025	−0.033 [−0.036, −0.031]
	random_50 (3)	1.416	+0.059	—
	excess_50 (3)	1.444	+0.088	+0.029 [+0.027, +0.031]
(b)	*step 0*	*1.3145*	—	—
	uniform (5)	1.3113	−0.0032	+0.0026 [+0.001, +0.004]
	random_50 (3)	**1.3086**	**−0.0059**	—
	excess_50 (3)	1.3116	−0.0029	+0.0030 [+0.001, +0.005]
(c)	*step 0*	*1.3569*	—	—
	uniform (3)	1.3542	−0.0027	+0.0013 [−0.001, +0.004]
	random_50 (3)	**1.3529**	**−0.0041**	—
	excess_50 (3)	1.3563	−0.0006	+0.0034 [+0.001, +0.006]

is retrained at 75% latent density, equidistant from `uniform` (100%) and from every 50% rule so that no condition is privileged by density match (equal density is not equal distance in every other respect); phase-4 runs at 10^{-5}, the least damaging rate in a sweep (Figure 2c); and the new warmstart is scored as step-0 inside the same pipeline. One disclosure: that sweep was scored on the *same* held-out set Table 1a reports, so the rate is test-set-informed; rescoring the checkpoints on a held-out set from a different corpus slice, never used in the selection, reproduces ordering and magnitudes to within 0.008 (0.467/0.140/0.023 against 0.474/0.143/0.024). Table 1a reports the result, and it is what we ask readers to take as the paper's allocation finding. Running the same corrected design on FineWeb-Edu replicates its central feature: uniform 2.134, `random_50` 2.247, `excess_50` 2.243 against a step-0 of 2.125 — *all three still above their start point*, by +0.009, +0.122 and +0.118. The step-0 result is therefore not specific to math or to the original recipe; it holds across two domains and both no-replay recipes.

Three things follow. First, under this recipe **the ordering holds**: uniform < random < excess, all three contrasts significant, with no condition privileged by density. Second, **the magnitudes in Table 4 are inflated about an order of magnitude**: uniform − random goes −0.489 → −0.033 and the demand harm +0.212 → +0.029 (seed 42). Third, **no condition improves on its start**, so the ordering says which allocation loses least on this metric, not which teaches most. ("Damage" means net loss on the evaluated axis, not the absence of gains on another.) Table 4 keeps its breadth, but only two of its orderings, and none of its magnitudes, are confirmed here.

Recipes that do not rescue it. The natural objection to the previous paragraph is that we simply chose a bad recipe. We tested three alternatives, each with `uniform` — the least damaging allocation, so a failure here is a statement about the design and not about routing — and each scored against the step-0 of *its own* starting checkpoint, with success fixed in advance as landing significantly below it. None does: a non-converged start goes 1.378 → 1.381 (+0.003, document-clustered interval spanning zero), removing the schedule restart gives 1.314 → 1.394 (+0.079), and 5× the tokens gives 1.314 → 2.210 (+0.895); Table 3 in Appendix D carries the intervals.

The sharpest form of the objection is older than we first credited: re-warming "first increases the loss" and "in the longer run" improves it (Gupta et al., 2023), re-warm, re-decay and *replay* are the standard recipe (Ibrahim et al., 2024), and the curve decomposes into distribution shift and learning-rate annealing (Wang et al., 2025b) — precisely the two artifacts we identified. So a short run may stop inside an expected rise. Arm C tests it: five times the tokens ends 0.90 above start, and repeating the arm on a 3× larger corpus (0.56 epochs, fresh tokens, ruling out repetition) makes it *worse* (3.441 vs. 2.210). The recovery does not arrive within 5× our budget.

Each eliminates an explanation: the non-converged checkpoint sat 0.064 above the converged one and closed none of it; removing the schedule restart made damage *worse* (our first reading was backwards — the annealing had been reducing damage by lowering the rate late). The lever that does matter is *replay*, which Ibrahim et al. (2024) prescribe against exactly this degradation and which we had not tried. Mixing the warmstart's own corpus back in shrinks the damage steeply in the replay ratio — $+0.474$, $+0.223$, $+0.123$, $+0.056$ at $0/10/25/50\%$ — and then flattens ($+0.0475$ at 75%, $+0.051$ at 100%), where 100% replay *is* the positive control. So full replay removes 89% of the damage in this sweep. Replay also changes repetition and the gradient–rate interaction, so the figure is an intervention effect, not a causal decomposition. The low rate, on its own and without replay, leaves $+0.024$ for `uniform` (original warmstart). We did not run the grid that would separate the two remedies' effects from their interaction, but they *compose*: at 50% replay and 10^{-5} the run ends 0.0032 *below* the checkpoint it started from — five training seeds, every one below, across-seed interval $[-0.0034, -0.0029]$ — the first setting we found in which the comparison measures learning rather than damage. So we ran the allocation comparison there, from the original warmstart and from the 75% one, where no mixture is closer to its start by density (Table 1b,c). All twenty runs improve on their start, and **uniform's lead does not survive**: random is ahead of it by $+0.0026$ (two-component 95% CI $[+0.001, +0.004]$), and by $+0.0013$ ($[-0.001, +0.004]$) once density is matched; the 50% rules' shorter documents also carry more replay tokens (53–57% vs. 43–48%). Demand-routing trails random in both ($+0.0030$, $+0.0034$). On a held-out split never used for any choice, (b) replicates and (c) identifies neither contrast; across three random masks (the original and two fresh) random stays ahead of uniform in all nine runs, though the across-mask interval spans zero; and filling in the rate $\times$ replay 2×2 puts uniform ahead in all three damaging cells and behind only where both remedies combine, with demand-routing behind random in all four (Appendix J). Throughout, nothing beats random where training helps; the ranking is recipe-dependent, and we do not claim crossing step 0 is what reordered it.

An observation: displacement. Every condition continues from the same warmstart for the same 600 steps, so a condition that moved the weights less would score better on a metric the warmstart already wins. It does (Figure 2b): $\|\theta_{\text{final}} - \theta_{\text{warmstart}}\|$ rank-correlates with held-out loss at Spearman $+0.94$ over all sixteen conditions (Pearson $+0.97$, $p < 10^{-9}$), within parameter groups too (feed-forward $+0.97$). Uniform moved least and `excess_50` most, in Table 4's order. It does not generalise, and we tried: across three public grids at the damaging rate the correlation is $+0.92$, -0.65 and $+0.58$, with no pattern we would defend (Appendix I). Here it tracked our ordering twice: $+0.94$ over sixteen conditions, and $+0.87$ over the nine of the corrected experiment at 10^{-5}, still above step-0 (`results/delta_theta_corrected.json`).

We first reported this as an identification problem — drift might *cause* the loss ordering, or a poorly matched allocation might cause the drift. Step 0 does not identify the cause; it removes one reading. The second requires allocation quality to be visible as learning, and no condition improves on its start on this metric, so drift is not a symptom of learning here; which of allocation and drift is upstream of the other stays open, and both are damage. Varying the step budget at fixed allocation gives the direction, not the cause: at `uniform` and lr=10^{-4}, held-out loss degrades at $\approx$1.09 nats per 1000 steps between 600 and 1800, and the 3000-step arm of Table 3 degrades monotonically (within each arm; schedules differ). Displacement is a correlate of the damage here, not a second measurement of it (`results/delta_theta_audit.json`).

Nor are uniform-trained models better at *using* latents: test-time latent benefit spans only 0.180–0.211 across the nine conditions while the with-latent loss spans 0.702, so what the allocation moves is general language-model quality (Appendix E).

Robustness to scale and domain. Three independent scalings leave the demand harm intact (Appendix G): tripling compute, scaling the model $2.5\times$ (three seeds), and tripling the data. The two contrasts behave differently. The demand harm is scale-invariant ($+0.201 \to +0.213 \to +0.178$), while uniform's budget edge decays with model scale ($-0.483 \to -0.376$, pooled) and becomes unidentifiable with the step budget. A 2×2 over corpus size and step budget, three seeds in its 1800-step row (Appendix F), puts that decay at $+0.43$ for $3\times$ the steps at *both* corpus sizes, while the contrast at 1800 steps is no longer distinguishable from zero (-0.061 $[-0.122, +0.001]$ at 200K; $+0.014$ $[-0.158, +0.185]$ at 600K); a single-seed reversal we first reported there did not survive seeding (Appendix F). Repeating the experiment on FineWeb-Edu separates them again: the

budget finding transfers (uniform best, beating the equal-budget control by 0.391 across three seeds, across-seed interval $[-0.434, -0.349]$) while the signal finding does not appear at all. Seeding the pivotal contrast three times gives -0.044, $+0.016$, -0.017: mean -0.015, across-seed 95% interval $[-0.090, +0.060]$, signs disagreeing. **The single-seed cross-domain sign flip we first reported does not replicate**: demand-routing's harm is specific to math ($+0.201$, three seeds), on general web text no effect of the signal is identified, and under the corrected recipe the FineWeb contrast shrinks elevenfold to -0.004, inside that seed band. In neither domain does any signal beat uniform under no-replay recipes (Appendix E).

The broader survey. Appendix D reports the full twelve-condition sweep the corrected experiment was distilled from — an oracle on measured per-chunk benefit, an RHO-Loss-style cross-fit selector, length, agreement, importance, the demand spectrum, and two mechanism probes. Its two pivotal contrasts are what Table 1a confirms — the other nine conditions were not re-run under the corrected recipe — and its magnitudes are recipe-inflated and should not be quoted. Three of its orderings constrain the mechanism: routing by the *measured* benefit — an oracle no realisable signal beats on its own terms — still loses to uniform, so under this recipe the failure is in the allocation objective, not in estimation error, and a generalisation-aware variant does not rescue it; spreading the same signal across documents removes most of its harm (one seed), suggesting allocation geometry; and routing by the *variance* of the demand estimate (a variance heuristic) beats the benefit oracle on the one seed we ran it — the direction we would seed first.

5 Discussion

Two checks we would like to see reported. Neither is expensive: a fixed count of evaluations. *Score step 0.* One evaluation of the starting checkpoint answers what the comparison cannot: did any condition improve? When the corpus is new to the checkpoint, score *two* axes — the training distribution and one it already knew — because adaptation gain on the first can mask damage on the second: continuing Pythia on WikiText-103, all twelve conditions improve on it while all twelve lose ground on FineWeb-Edu (Appendix I). When the answer is no, every between-condition gap is a difference in damage, and the natural verbal gloss — "allocation A teaches the model more than B" — is unsupported no matter how many seeds it survives. Ours survived three seeds, a second domain and $2.5\times$ scale while wrong in exactly this way. *Check the recipe before reading the ranking.* Lowering the rate tenfold removed 92% of our contrast and 86–92% of the between-condition spread in three public grids, and under the recipe that improves on step 0 a different condition led; a ranking that has been seen only under one recipe is a claim about that recipe. Displacement ($\|\theta - \theta_{\text{init}}\|$ against the metric) is an optional observation: $\rho = 0.94$ here, but no consistent sign across public grids.

What survives, and what it is worth. After addressing both artifacts — a sensitivity analysis, not a proof that no confound remains — under the damaging recipe every sign we re-ran holds at 7–14% of its original size: uniform still beats an equal-budget random control (-0.033), demand-routing still loses to it ($+0.029$), both with across-seed intervals clear of zero. They are small enough that we would not have built a paper around them; the uncorrected versions were large enough that we did. The mechanism suggested (one seed, not re-run) is coverage: concentrating the budget on a narrow high-divergence slice fares worse than spreading it, and `excess_strat`, the same signal applied by within-document rank at full coverage, removes 94% of the harm. What we can no longer defend is uniform as *the better teacher*: it is the least damaging condition under the damaging recipe and not first under either recipe where training helps.

Why this is not only our problem — and how far that goes. The mechanism is not new: Wang et al. (2025b) decompose the CPT loss curve into distribution shift and learning-rate annealing, Gupta et al. (2023) report the rise-above-start that re-warming causes, and Zhou et al. (2026b) predict the optimal rate from the checkpoint's own loss. Closest, Wang et al. (2026) show proxy data rankings flip with hyperparameters and recommend a reduced rate, for pretraining from scratch; in continued pretraining the reduced rate alone did not settle our question (one three-condition case, not a test of their procedure) — step 0 showed its ranking was still one of damage (Table 1a), and where training helped, its leader no longer led (b, c). What has not carried over is the consequence for studies that use CPT as a *measuring instrument*: the design makes those factors common-mode across conditions, and every internal check we ran (seeds, domains, scales, clustered bootstraps, a frozen

metric) asks whether the ordering is *stable*, which it was, not whether it is *about learning*. Nor is it ours: rebuilding the experiment on a public checkpoint, corpus and trainer that share no code with ours (three grids, 216 runs; Appendix I) removes 86–92% of the between-condition spread when the learning rate drops tenfold (with fp32 master weights), and in every grid leaves all twelve conditions worse than step 0 on the axis the checkpoint was not trained on.

How widespread that is, we measured rather than asserted (Appendix H). Of twenty-five continued-pretraining data-recipe comparisons, **twenty-four report the starting checkpoint in some form** — our earlier "omitted by default" was wrong and is withdrawn — and what survives is a difference in form. Among the ten papers ranked by held-out loss, four give the start as an explicit value; among the fifteen ranked by downstream accuracy, twelve do; a number, unlike a figure point, can be differenced by any reader. We report that split as suggestive, not established (Fisher $p = 0.087$). None of the ten loss-ranked papers reports a displacement measure. We do not claim particular published orderings are damage rankings; we claim a start shown graphically does not settle what a difference would, and that both checks are nearly free.

Relation to the data-selection literature. Our surviving finding rhymes with RHO-Loss (Mindermann et al., 2022), whose reducible-holdout-loss reframing is the argument that per-example loss is not the right benefit signal; influence estimates in deep networks are themselves fragile (Basu et al., 2021), and memorizing hard examples can hurt generalization (Bayat et al., 2025). Closest in method is Mohri et al. (2026), who carry an unfiltered-pool baseline through every comparison and find that filters winning under a small compute budget lose to no filtering at all once compute grows — an apparent selection benefit that survives only while the control is absent. Our step-0 baseline is that control one level down. Ours is a case where a predictive per-example signal (ρ=0.48) still yields no useful allocation rule. Appendix K covers latent-reasoning pretraining (Kong et al., 2025; Zhu et al., 2025; Zhou et al., 2026a; Zeng et al., 2026).

6 Conclusion

We set out to show that a principled per-chunk signal allocates latent budget better than uniform. It does not. The more transferable finding is what it took to establish that reliably.

Reasoning demand is a valid per-chunk predictor: length-corrected, it predicts measured per-chunk latent benefit at $\rho = 0.48$ where the incumbent importance weight is weak, and its EM trajectory admits a null-calibrated check for collapse.

The allocation result is where the instrument was our own experiment: a shared warmstart gave a clean, seed-stable ordering that reproduced on a second domain and at $2.5\times$ scale, and scoring that warmstart, which the design does not require, showed every condition worse than it. Two recipe artifacts produced most of the gap — a 100%-density warmstart only `uniform` matches and an annealed cosine restarted at 10^{-4}; rebuilt without them both pivotal signs survive at 7–14% of their magnitude.

The substantive conclusion about allocation is a small negative one: where training helps, demand routing — the one signal we tested there — does not beat random, and uniform's lead over random, the paper's original headline, does not survive; the honest effect sizes are an order of magnitude below what the standard design reported. The methodological conclusion is the one we would press: score step 0 on both axes and sweep the rate — the two checks that reproduced in every public grid — and do not read a ranking seen under one recipe as a property of the allocations. When step-0 wins, replay and a lower rate together put our comparison back below its start, so the design is a damage ranking at every setting we tried except replay combined with the low rate; without these checks a comparison can pass every robustness test in this paper and still not mean what it appears to.

7 Limitations

The standard version of our allocation experiment measures damage resistance, not learning; only 50% replay at the low rate improves on the start, and there uniform loses its lead. Those contrasts are small (0.001–0.006 nats, one scale and endpoint, one signal re-tested): they remove uniform's lead without establishing a better rule. Other caveats: Appendix C.

ETHICS STATEMENT

This work studies where to place model-generated reasoning traces in pretraining text. It introduces no new human-subjects data, no personally identifying information, and no new model release. All corpora are public (FineMath-4plus, FineWeb-Edu, WikiText-103) and all models are public checkpoints (TinyLlama-1.1B, Sheared-LLaMA-2.7B, Pythia-160m and -410m); the latent thoughts are model-generated and inherit whatever biases those base models carry, which we do not audit here. The principal societal consideration is compute: the experiments in this paper consumed roughly 400 A100-hours for the TinyLlama and Sheared-LLaMA runs and about 90 more for the clean-room grids, their precision control and the protocol comparisons, and our main practical finding is a *negative* one that should prevent others from spending compute on per-chunk demand-based routing. We report the result that argues against our own starting hypothesis, including several claims we withdraw (Section 4; Appendices F, I and H).

USE OF LARGE LANGUAGE MODELS

LLM assistance was used throughout this work and we disclose it fully. *Writing*: drafting and copy-editing prose, including first drafts of sections, and an adversarial-review pass stress-testing claims against the released artifacts. *Research execution*: implementing and operating the analysis, training-orchestration and plotting code under the authors' direction. *Retrieval and discovery*: locating related work, and fetching and summarising the full text of the papers coded in the reporting-practice survey (Appendix H) — which is why each survey row records a verbatim quote and the source actually read (`results/survey/rows.jsonl`), with the coding rules frozen in the released protocol. *Mathematics*: the proofs in Appendix B, which restate known identities, were drafted with LLM assistance and checked by the authors. *Data*: the latent-thought corpora are themselves LLM-generated — that pipeline is the object of study, and it is described and released in full. No reported number originates in an LLM's text output: every measurement is computed by the released code from the released data, and `scripts/audit_paper_numbers.py` re-checks the 358 quantitative claims it lists against those results (not every sentence); several errors surfaced through that audit and were corrected (an incorrectly transplanted selection-overlap statistic, an over-broad conclusion sentence, and i.i.d. confidence intervals that ignored document clustering).

REPRODUCIBILITY STATEMENT

All quantitative claims are backed by released artifacts. The CMI fixed-point theory (Theorem 1, Corollaries 1–2) is stated with explicit assumptions A1–A3 and proved in full in Appendix B. The reasoning-demand estimator, excess-demand length-correction (isotonic), and all signal computations are deterministic given the released per-chunk logprobs. The allocation experiment fixes the warmstart checkpoint, the 600-step budget, the `latent_type=routed` training recipe, and a leakage-free held-out split (3000 chunks from 370 documents disjoint from the training selection, with zero overlap asserted programmatically; document-clustered bootstrap CIs throughout Phase 4, produced by `scripts/doc_cluster_bootstrap.py`). Each routing condition differs *only* in the per-chunk allocation decision; the oracle routes by measured *training-corpus* per-chunk Δloss (an upper bound on per-chunk-benefit selection, not a held-out generalization oracle). All held-out latent-aware losses use the *common-span* metric — both the with- and without-latent prompts are truncated to an identical suffix token span and identical kept prefix, so every chunk is scored without the truncation bias that the naive drop-prefix policy introduces on long-prefix documents; the metric is frozen with a written specification and CPU unit tests that assert byte-identical scored spans. Diagnostic anchors (Section A, Table 2) are computed on $N{=}10$K chunks with $K{=}8$ Monte Carlo samples. The step-0 check is one extra invocation of the same evaluator every condition already uses ($\approx$4.5 GPU-minutes for 3000 chunks, under 1% of this experiment's training compute); the displacement observation is computed by `scripts/delta_theta_audit.py` from two checkpoint directories (110 s wall-clock on CPU over all 17 checkpoints, no GPU). `scripts/metric_independence.py` reproduces the latent-free rescoring from the per-chunk outputs, and `scripts/audit_paper_numbers.py` re-checks 358 quantitative claims in this paper against the result files they cite; it covers the numbers it lists, not every sentence. Exact configurations, seeds, and the figure/table-generating analysis code are provided in the supplementary

material. *Training details.* Phase-4 runs continue TinyLlama-1.1B in the PAIR/lingua trainer for 600 steps of 4×2048 tokens with 8-step gradient accumulation (39.3M tokens), AdamW ($\beta = (0.9, 0.95)$, weight decay 0.01) at a peak rate of 10^{-4} (original runs) or 10^{-5} (corrected and protocol runs), 60 warmup steps and cosine decay; FSDP2 mixed precision computes in bf16 but keeps fp32 master weights and optimizer state, and the optimizer is re-initialised from the warmstart's weights (BoLT's continual setting carries it over). The partially accumulated gradient is clipped at norm 1.0 after every micro-batch rather than once per step; the logged norm exceeds 1.0 on 2–3% of logged steps. The clean-room trainer is described in Appendix I.

REFERENCES

Alexander A. Alemi, Ben Poole, Ian Fischer, Joshua V. Dillon, Rif A. Saurous, and Kevin Murphy. Fixing a broken ELBO. In *International Conference on Machine Learning (ICML)*, pp. 159–168, 2018.

Samyadeep Basu, Philip Pope, and Soheil Feizi. Influence functions in deep learning are fragile. In *International Conference on Learning Representations (ICLR)*, 2021. arXiv:2006.14651.

Reza Bayat, Mohammad Pezeshki, Elvis Dohmatob, David Lopez-Paz, and Pascal Vincent. The pitfalls of memorization: When memorization hurts generalization. In *International Conference on Learning Representations (ICLR)*, 2025. arXiv:2412.07684.

Thomas M. Cover and Joy A. Thomas. *Elements of Information Theory*. Wiley-Interscience, 2nd edition, 2006.

Sachin Goyal, Pratyush Maini, Zachary C. Lipton, Aditi Raghunathan, and J. Zico Kolter. Scaling laws for data filtering: data curation cannot be compute agnostic. In *IEEE/CVF Conference on Computer Vision and Pattern Recognition (CVPR)*, 2024.

Kshitij Gupta, Benjamin Thérien, Adam Ibrahim, Mats L. Richter, Quentin Anthony, Eugene Belilovsky, Irina Rish, and Timothée Lesort. Continual pre-training of large language models: How to (re)warm your model? *arXiv preprint arXiv:2308.04014*, 2023.

Matthew D. Hoffman and Matthew J. Johnson. ELBO surgery: yet another way to carve up the variational evidence lower bound. In *NeurIPS Workshop on Advances in Approximate Bayesian Inference*, 2016.

Adam Ibrahim, Benjamin Thérien, Kshitij Gupta, Mats L. Richter, Quentin Anthony, Timothée Lesort, Eugene Belilovsky, and Irina Rish. Simple and scalable strategies to continually pre-train large language models. *Transactions on Machine Learning Research*, 2024. arXiv:2403.08763.

Laurent Itti and Pierre F. Baldi. Bayesian surprise attracts human attention. In *Advances in Neural Information Processing Systems 18 (NIPS 2005)*, pp. 547–554, 2006.

Deqian Kong, Minglu Zhao, Dehong Xu, Bo Pang, Shu Wang, Edouardo Honig, Zhangzhang Si, Chuan Li, Jianwen Xie, Sirui Xie, and Ying Nian Wu. Latent thought models with variational bayes inference-time computation. In *Proceedings of the 42nd International Conference on Machine Learning (ICML)*, pp. 31419–31436, 2025.

Sören Mindermann, Jan Brauner, Muhammed Razzak, Mrinank Sharma, Andreas Kirsch, Winnie Xu, Benedikt Höltgen, Aidan N. Gomez, Adrien Morisot, Sebastian Farquhar, and Yarin Gal. Prioritized training on points that are learnable, worth learning, and not yet learnt. In *International Conference on Machine Learning (ICML)*, 2022. arXiv:2206.07137.

Kaixiang Mo, Yuxin Shi, Weiwei Weng, Zhiqiang Zhou, Shuman Liu, Haibo Zhang, and Anxiang Zeng. Mid-training of large language models: A survey. *arXiv preprint arXiv:2510.06826*, 2025.

Christopher Mohri, John Duchi, and Tatsunori Hashimoto. A bitter lesson for data filtering. *arXiv preprint arXiv:2605.19407*, 2026.

Niklas Muennighoff, Alexander M. Rush, Boaz Barak, Teven Le Scao, Aleksandra Piktus, Nouamane Tazi, Sampo Pyysalo, Thomas Wolf, and Colin Raffel. Scaling data-constrained language models. In *Advances in Neural Information Processing Systems (NeurIPS)*, 2023.

David Raposo, Sam Ritter, Blake Richards, Timothy Lillicrap, Peter Conway Humphreys, and Adam Santoro. Mixture-of-depths: Dynamically allocating compute in transformer-based language models. *arXiv preprint arXiv:2404.02258*, 2024.

Yangjun Ruan, Neil Band, Chris J. Maddison, and Tatsunori Hashimoto. Reasoning to learn from latent thoughts. *arXiv preprint arXiv:2503.18866*, 2025.

Ben Sorscher, Robert Geirhos, Shashank Shekhar, Surya Ganguli, and Ari S. Morcos. Beyond neural scaling laws: beating power law scaling via data pruning. In *Advances in Neural Information Processing Systems (NeurIPS)*, 2022.

Jakub M. Tomczak and Max Welling. VAE with a VampPrior. In *International Conference on Artificial Intelligence and Statistics (AISTATS)*, pp. 1214–1223, 2018.

Jiachen T. Wang, Tong Wu, Kaifeng Lyu, James Zou, Dawn Song, Ruoxi Jia, and Prateek Mittal. Can small training runs reliably guide data curation? rethinking proxy-model practice. In *International Conference on Learning Representations (ICLR)*, 2026.

Liang Wang, Nan Yang, Shaohan Huang, Li Dong, and Furu Wei. Thinking augmented pre-training. *arXiv preprint arXiv:2509.20186*, 2025a.

Xingjin Wang, Howe Tissue, Lu Wang, Linjing Li, and Daniel Dajun Zeng. Learning dynamics in continual pre-training for large language models. In *Proceedings of the 42nd International Conference on Machine Learning (ICML)*, pp. 64525–64548, 2025b.

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 (NeurIPS)*, 2023a. arXiv:2305.10429.

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 (NeurIPS)*, 2023b. arXiv:2302.03169.

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

Eric Zelikman, Georges Harik, Yijia Shao, Varuna Jayasiri, Nick Haber, and Noah D. Goodman. Quiet-STaR: Language models can teach themselves to think before speaking. In *Conference on Language Modeling (COLM)*, 2024. arXiv:2403.09629.

Boyi Zeng, Yiqin Hao, He Li, Shixiang Song, Feichen Song, Zitong Wang, Siyuan Huang, Yi Xu, ZiWei He, Xinbing Wang, and Zhouhan Lin. Pretraining with token-level adaptive latent chain-of-thought. *arXiv preprint arXiv:2602.08220*, 2026.

Xiangxin Zhou, Zichen Liu, Haonan Wang, Chao Du, Min Lin, Chongxuan Li, Liang Wang, and Tianyu Pang. Variational reasoning for language models. In *International Conference on Learning Representations (ICLR)*, 2026a. arXiv:2509.22637.

Yongwei Zhou, Juncheng Diao, Junlin Shang, Peiguang Li, and Rongxiang Weng. Predictable scaling laws of optimal hyperparameters for LLM continued pre-training. *arXiv preprint arXiv:2606.05610*, 2026b.

Rui-Jie Zhu, Zixuan Wang, Kai Hua, Tianyu Zhang, et al. Scaling latent reasoning via looped language models. *arXiv preprint arXiv:2510.25741*, 2025.

A Diagnostics: Self-Distillation and Collapse

Demand's trajectory across EM iterations is a window onto distillation health. On N=10K chunks (K=8), $\mathbb{E}[\hat{D}]$ falls from 51.9 to 41.8 between iterations 1 and 2 (-19.6%, chunk-level Wilcoxon $p < 10^{-300}$) with per-chunk rank persistence $\rho = 0.89$.

A null-calibrated collapse diagnostic. Reading "variance contracts faster than the mean" as collapse is mechanically confounded: under uniform multiplicative shrinkage $\hat{D}^{(t+1)} = c\,\hat{D}^{(t)}$ — no diversity loss at all — variance contracts as c^2 while the mean contracts as c. We therefore calibrate against that null, $\Delta\mathrm{Var}_{\text{null}} = c^2 - 1$. Empirically $c = 0.804$, so the null predicts -35.3% against an observed -39.0%: an excess contraction of 3.7 points. With rank persistence 0.891 and a 0% clip rate, this check finds no evidence of collapse at iteration 2 — which is weaker than evidence of its absence, since uniform shrinkage $c < 1$ preserves ranks while driving demand toward zero. **The contribution is the null, not a detection**, and we state its limits plainly: a full criterion would also require the measured I_{q_t} to fall below an independent intrinsic-CMI estimate, and we have no such estimator — so what we contribute is the multiplicative null and the rank-persistence check, two conditions rather than three. Two EM iterations also cannot distinguish decay to a non-zero floor from decay to zero; that reading rests on the identity in Appendix B, not on these points. Figure 3 (Appendix B) shows the trajectory.

B Theoretical Framework: Proofs

This appendix gives the full assumptions and proofs underlying Section 2. Empirical anchors are from N=10,000 chunks, K=8 MC samples, EM iterations 1–2, TinyLlama-1B, FineMath.

B.1 Statements

These are the formal statements summarized in Section 2.

Theorem 1 (Latent-span optimum is the aggregated posterior)**.** *At EM iteration t, consider the population next-token cross-entropy of the latent span alone, $\mathcal{L}(p) = \mathbb{E}_{(P,S)\sim\mu}\,\mathbb{E}_{z\sim q_{\theta^{(t)}}(\cdot|P,S)}[-\log p(z \mid P)]$. Under realizability and optimization to the population optimum* (A1)*, the unique minimizer is the aggregated posterior $p^\star(\cdot \mid P) = m_{\theta^{(t)}}(\cdot \mid P)$.*

Corollary 1 (Static CMI identity)**.** *With the prior exactly at the latent-span optimum $p^\star = m_t$ and the posterior frozen at q_t,*

$$\mathbb{E}[D] \;=\; I_{q_t}(z;\, S \mid P)\,,$$

the conditional mutual information between latent and suffix given prefix, under the model-induced joint $\mu(P)\mu(S \mid P)q_t(z \mid P, S)$.

Corollary 2 (Within-chunk monotonicity, and what it does not imply)**.** *For a fixed chunk, appending suffix tokens cannot decrease the CMI: $I(z; S \mid P)$ is non-decreasing in suffix length by the chain rule. This does* not *derive the cross-sectional correlation $\rho(D_i, \ell_i)$ across different chunks; the observed positive correlation ($\rho(\hat{D}, \ell) = 0.46$) is* consistent with *but* not predicted by *the within-chunk result.*

B.2 Assumptions

Assumption 1 (Realizability + optimization, A1)**.** *The model family is expressive enough to represent the aggregated posterior $m_{\theta^{(t)}}(\cdot \mid P)$ for every prefix P, and optimization reaches the population optimum on the latent span.*

Assumption 2 (Latent-span dominance, A2 — limitation)**.** *The deployed M-step minimizes full-sequence cross-entropy over the concatenation $(P, [\text{SOL}], z, [\text{EOL}], S)$ with one shared θ (no latent-span masking). Theorem 1 analyzes only the latent-span term. A2 assumes that at the optimum the suffix-reconstruction term $\mathbb{E}[-\log p_\theta(S \mid P, z)]$ and the prefix term do not displace the prior's z-marginal from the latent-span optimum — equivalently, the prior's z-marginal is effectively unconstrained by the other loss terms. This is an assumption, not a derived property, and is where the theory's contact with the deployed system is weakest.*

Assumption 3 (Bounded posterior drift, A3)**.** *Monotone-decrease claims for the trajectory $\mathbb{E}[D^{(t)}]$ additionally require that the posterior's drift across an EM step is dominated by the prior's progress toward m_t. We do not prove this; it is an empirically testable assumption.*

B.3 Proof of Theorem 1

Fix P. The inner objective is

$$\mathbb{E}_{S|P}\,\mathbb{E}_{z\sim q_t(\cdot|P,S)}[-\log p(z \mid P)] = \mathbb{E}_{z\sim m_t(\cdot|P)}[-\log p(z \mid P)] = H\big(m_t(\cdot \mid P),\, p(\cdot \mid P)\big),$$

the cross-entropy of p against m_t, where the first equality uses that the z-marginal under $\mu(S \mid P)q_t(z \mid P, S)$ is exactly $m_t(z \mid P) = \mathbb{E}_{S|P}[q_t(z \mid P, S)]$. Cross-entropy over an unconstrained conditional distribution is uniquely minimized at $p = m_t(\cdot \mid P)$ by Gibbs' inequality. Averaging over $P \sim \mu(P)$ preserves the pointwise optimum. □

Scope. Theorem 1 characterizes the latent-span term in isolation. Under A2 it transfers to the deployed M-step; without A2 it identifies the *direction* the latent-span gradient pulls the prior, not the realized optimum. The prior conditions on P alone and targets the suffix-average $m(\cdot \mid P)$, never the per-instance posterior $q(\cdot \mid P, S)$.

B.4 Proof of Corollary 1

Under the joint law $\mu(P)\,\mu(S \mid P)\,q_t(z \mid P, S)$, the marginal of z given P is exactly $m_t(z \mid P)$. With the prior at $p^\star = m_t$,

$$\mathbb{E}_{S|P}[D] = \mathbb{E}_{S|P}\big[D_{\mathrm{KL}}\big(q_t(\cdot \mid P, S) \,\|\, m_t(\cdot \mid P)\big)\big] = I_{q_t}(z; S \mid P{=}P),$$

the expected KL between the conditional $q_t(z \mid P, S)$ and its marginal $m_t(z \mid P)$, which is the definition of conditional mutual information under that joint (Cover & Thomas, 2006). Taking the outer expectation over $P \sim \mu(P)$ gives $\mathbb{E}[D] = I_{q_t}(z; S \mid P)$. □

Interpretation. Expected demand at the optimum equals the information the suffix carries about the model's latent reasoning under the model-induced joint — a non-negative floor, zero only in the degenerate case $q(z \mid P, S) = q(z \mid P)$. Under A2 failure, the realized prior differs from m_t and $\mathbb{E}[D]$ acquires an additional non-negative gap term.

B.5 Proof of Corollary 2

For a fixed chunk write $S = (S_{1:j}, S_{j+1:\ell})$. By the chain rule for mutual information,

$$I(z; S \mid P) = I(z; S_{1:j} \mid P) + I(z; S_{j+1:\ell} \mid P, S_{1:j}) \;\geq\; I(z; S_{1:j} \mid P),$$

since CMI is non-negative for arbitrary distributions. Thus appending suffix tokens cannot decrease the CMI. □

What this does not establish. The monotonicity concerns nested prefixes of a *single* chunk's suffix under one fixed joint distribution; the procedure that re-forms $q(z \mid P, S)$ for each suffix length needs the posteriors at different lengths to be marginals of one joint, which we assume rather than show. It does not derive the cross-sectional correlation $\rho(D_i, \ell_i)$ across different chunks: long chunks may systematically carry lower per-token latent-relevant information — the information-density variation that Definition 2 exploits. Observed $\rho(\hat{D}, \ell) = 0.46$ (iter 1) is positive and *smaller* than the single-sample importance weight's $\rho(-\log w, \ell) = 0.71$, whose extra length dependence comes from the per-token log-likelihood sum rather than information content.

B.6 Loop dynamics and self-consistent fixed points

Theorem 1 and Corollary 1 characterize a single M-step against a frozen posterior. In the full loop q also moves. If the loop converges to $\theta^\star$ with $p_{\theta^\star}(\cdot \mid P) = m_{\theta^\star}(\cdot \mid P)$, then $\mathbb{E}[D] = I_{q_{\theta^\star}}(z; S \mid P)$ exactly (Corollary 1 at $\theta^\star$); the value of the floor depends on where q lands, and finite capacity adds

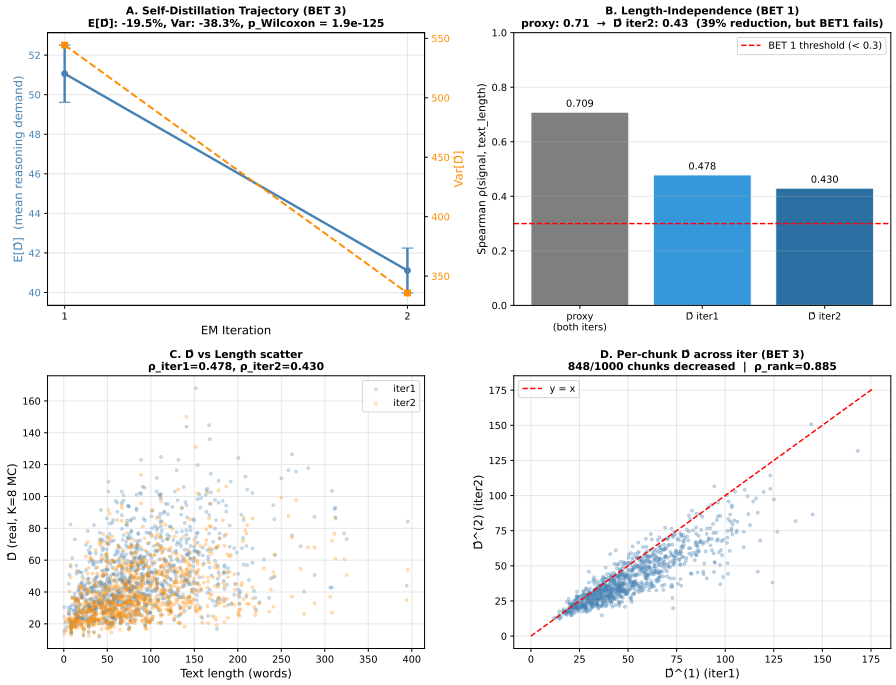

Figure 3: Self-distillation diagnostics over EM iterations 1→2 (N=10K). Mean demand decreases 19.6% (consistent with, not evidence of, a non-zero floor); the variance contraction (−39%) barely exceeds the multiplicative null (−35.3%) and rank persistence stays high (ρ=0.89): no evidence of collapse under this check, which cannot by itself establish its absence.

a non-negative excess-KL term $\mathbb{E}_P[D_{\mathrm{KL}}(m_t \| p_t)]$ to the population quantity (the clipped Monte-Carlo estimator $\hat{D}$ adds its own bias). The empirical trajectory ($\mathbb{E}[\hat{D}]$: $51.94 \to 41.77$, -19.6%, chunk-level $p < 10^{-300}$; rank persistence 0.89) is consistent with movement toward a non-zero floor, though two iterations cannot establish one.

B.7 Excess demand: empirical properties

Table 2 records the properties of $\tilde{D}$ on N=10K chunks with per-iteration-independent isotonic fits.

Table 2: Excess demand properties (N=10K, isotonic length baseline).

Property	Value	Reading
$\rho(\tilde{D}, \ell)$	-0.047	length-free by construction
Cross-iteration persistence (in-sample $\hat{f}$)	0.857	—
Cross-iteration persistence (2-fold cross-fit)	0.857	not a shared-nuisance artifact
Permutation null (identity shuffled)	0.001 ± 0.011	persistence is real signal
$\rho(\tilde{D}, -\log w)$	0.209	decoupled from importance weight
Top-50% Jaccard vs. length routing	0.329	selects different chunks than length
MC clip rate ($\hat{D}_{\text{unclipped}} < 0$)	0.00%	no clipping bias at K=8

C Limitations in Full

The allocation experiment measures damage resistance, not learning. Every condition we trained without replay ends above the warmstart it started from, including under the corrected recipe (Table 1a). We can say which allocation degrades the checkpoint least; we cannot say which teaches it more. We searched for a regime in which training does improve its start point — a non-converged start, no schedule restart, 5× the tokens — and none of those does; only 50% replay at the low rate

does, by 0.003 nats over five seeds (Appendix J), and there uniform loses its lead (Table 1b,c): random routing is ahead of it, significantly only when uniform is not density-matched, and neither routed signal beats random (App. J gives an independent-split check). That comparison has three seeds per routed condition, one scale and 0.001–0.003-nat contrasts, so it removes the damage-regime winner without establishing a better rule; other recipes, BoLT's continual variant among them, may order the conditions differently again. Our redesign removes two specific artifacts; it does not establish that no others remain.

Scale. Our runs use 600 steps ($\sim$39M tokens), far below BoLT's 13B-token regime (its fixed-data setting re-trains from the base TinyLlama each iteration at 10^{-4}; its continual setting continues a pre-decay checkpoint, optimizer included, at a constant 3×10^{-5} without re-warming). Uniform's margin over random shrinks as either data or parameters grow and is no longer identified at 1800 steps (the single-seed sign change at $3\times$ data did not survive seeding), so we present the budget contrast as a small-*budget* effect and do not extrapolate it; the signal contrast (demand-routing worse than random) holds at every scale we test. A billion-token confirmation at BoLT's scale ($\sim$hundreds of GPU-days) is the remaining open scale. The model-scale test holds the latent corpus and D_i fixed at their 1.1B values, isolating consumer-model scale; whether D_i re-estimated by a larger posterior would rank chunks differently is untested.

Seeds. The pivotal conditions carry three training seeds with a two-level hierarchical bootstrap; five exploratory conditions carry two and are marked as such (Table 4). The 2.7B run carries three. Table 1a, the corrected allocation result, now carries three seeds as well, with across-seed intervals clear of zero and a seed spread of only 0.001 — an order of magnitude below the spread at lr$=10^{-4}$, as expected when the runs move less. The remaining side experiments are **single-seed**: the two exploratory allocation axes. Seed variation is not negligible here — at 2.7B, seed 42 alone would have overstated both effects. We flagged the cross-domain sign flip as the case most at risk, its native-warmstart magnitude (-0.019) being smaller than the seed spread we measure on math (0.024). Seeding it confirmed the concern and went further: across seeds $42/43/44$ the FineWeb contrast is -0.044, $+0.016$, -0.017 (mean -0.015, across-seed CI $[-0.090, +0.060]$), so the sign itself is not stable and the flip is withdrawn. We record this because it is the paper's own instance of the failure it describes: three chunk-level intervals, each excluding zero, disagreeing with one another about the sign. The math harm ($+0.20$) and budget effect (-0.48) are an order of magnitude above that spread.

Latent-corpus provenance. The warmstart — and the 75% warmstart of the corrected experiment, which resamples the same source — was trained on `finemath_4plus_..._with_gpt_latent_thoughts_latents`, the GPT-seeded corpus BoLT uses to initialise its bootstrap. Every phase-4 condition, and the held-out set, come from the EM iteration-1 cache. The step-0 comparison therefore crosses a change of latent generator as well as a change of allocation, and we cannot separate "continued pretraining degrades this checkpoint" from "iteration-1 latents are weaker training data than the GPT-seeded ones" with the runs we have. The design that would separate them is a warmstart trained on iteration-1 latents, which we did not run. What we can say is that the confound does not run in the convenient direction: the held-out latents come from the same iteration-1 cache the conditions train on, so they are out-of-distribution for the warmstart and in-distribution for its competitors, and the latent-free metric — no latent in the prompt at all — reproduces the same $16/16$ result.

Theory–system gap. Theorem 1 characterizes the latent-span term in isolation; its transfer to BoLT's shared-θ M-step rests on assumption A2 (Appendix B), the weakest point of contact with the deployed system, which we do not prove.

Downstream signal. GSM8K/MATH are near floor at 600 steps, so we rely on held-out NLL — scored both with and without a latent in the prompt — rather than task accuracy.

Domain transfer. The primary FineWeb-Edu conditions continue-pretrain from the math-warmstarted base for a same-init comparison with math. A *native* FineWeb warmstart gives -0.019, which we previously read as reproducing the flip; with the seed spread now measured it is one more draw from a distribution centred near zero, not independent confirmation. The FineWeb *budget* contrast, by contrast, is seeded and holds: -0.391 across seeds $42/43/44$, interval $[-0.434, -0.349]$.

Table 3: Three recipes, none of which lets continued pretraining improve its start point. Document-clustered 95% CIs.

Recipe	start	after	Δ	verdict
non-converged start (step 2500)	1.378	1.381	+0.003	not distinguishable
constant lr, no warmup	1.314	1.394	+0.079	worse
5× token budget (3000 steps)	1.314	2.210	+0.895	far worse

D THE TWELVE-CONDITION SURVEY

The sweep from which Section 4's corrected experiment was distilled. Every magnitude here is inflated by the two recipe artifacts analysed in Section 4; two orderings — uniform ahead of random, random ahead of high-demand routing — and not the numbers are what the corrected experiment confirms.

Result. Table 4 reports twelve conditions: uniform, ten budget-constrained routing rules, and a generalization-aware `rho_holdout_50` baseline (all 3000 held-out chunks; paired differences pooled over three training seeds with a two-level hierarchical bootstrap). Three findings hold. First, **uniform allocation is the least damaging** (1.789), below every budget-constrained routing rule — though above the untrained warmstart (1.314), so "best" here means "degrades the checkpoint least". Second, **this is a budget effect, not a signal effect**. Uniform applies latents to 100% of chunks and the routed conditions to 50%; the equal-budget *random* control (`random_50`, 2.278) is itself beaten by uniform by 0.483 nats/token ($[-0.554, -0.412]$) — the one large, seed-stable effect. Pooled intervals here carry *two* error components: seed-to-seed variation, which averages out over S seeds, and evaluation-set variation, which does not, because the seeds share one held-out set. Our first implementation resampled chunks independently inside each seed and so divided the evaluation term by $\sqrt{S}$; the intervals we report are recomputed with the terms added and the evaluation term document-clustered, which widens them four- to ninefold and changes one verdict. Against the same control, importance beats random by 0.077 $[-0.120, -0.035]$ and the oracle by 0.051 $[-0.084, -0.017]$ — both a fraction of uniform's margin. The oracle contrast needed a third seed to be identified at all: on the two seeds Table 4 carries, its across-seed interval spanned zero; the third was run for the contrast alone (`results/panel_experiments/summary.json`, E4). Third and most telling, **the demand signal is actively harmful here**: high-demand routing (`excess_50`) is significantly *worse* than random by 0.201 ($[+0.154, +0.248]$, sign-consistent across all three seeds), and *no* demand-based condition beats random. **The oracle still underperforms uniform** (2.212): it routes by each chunk's *measured* per-chunk Δloss on the *training* corpus — the best attainable per-chunk-benefit selection, but *not* a held-out generalization oracle (Appendix K) — yet even this upper bound loses to treating all chunks equally. **A generalization-aware selector fares no better**: an RHO-Loss-style `rho_holdout_50` condition — routing by a holdout-reducible benefit computed with cross-fit models that never trained on the scored chunk (Appendix K) — lands at 2.212, not distinguishable from the oracle (paired difference −0.002 over the two seeds the oracle carries; two-component 95% CI $[-0.086, +0.082]$) and still worse than uniform by 0.428 (two-component CI $[0.361, 0.494]$; all three seeds sign-consistent). Making the benefit signal generalization-aware does not rescue per-chunk routing. The demand→generalization relationship trends negative (high-demand `excess_50` 2.491 worst among the primary conditions; mid/low-demand rows single-seed, exploratory). Under this damaging recipe no tested signal beats uniform; among the seeded conditions none beats the importance baseline (2.190), and the single-seed `uncertainty` condition (2.138) that does is treated as exploratory below.

Mechanism: concentration, not bad benefit. It is tempting to explain the harm by saying high-demand chunks have *illusory* or non-transferable per-chunk benefit. The data reject this. Using the cross-fit holdout models above (each scores chunks it never trained on), we find that the per-chunk latent benefit of high-demand chunks *generalizes across models*: excess demand predicts the *holdout* benefit as strongly as the training benefit (Spearman 0.44 vs 0.48), the two benefits correlate at ρ=0.74, and the top demand decile has the *highest* holdout benefit (0.37). The latents on high-demand chunks genuinely help — even a model that never trained on them. Under this damaging recipe the failure is one of *allocation*, not benefit quality: concentrating the fixed training budget on

Table 4: Held-out allocation comparison (continued pretraining, matched 600-step budget; uniform applies latents to all chunks, routed conditions to 50%). Held-out latent $\text{loss}_{\text{with}}$/token on the common-span metric (all n=3000 chunks, seed-42 shown; three-seed pooled paired differences in text); lower is better. Uniform is least damaging. The equal-budget `random_50` control isolates signal from budget: the high-demand condition is *worse than random* and no demand condition is better, while importance/length/the oracle beat it only marginally. The oracle (top 50% by *measured training-corpus* per-chunk Δloss — not a generalization oracle) still loses to uniform, as does `rho_holdout_50` (an RHO-Loss-style generalization-aware selector, not distinguishable from the oracle). **Magnitudes here are inflated about an order of magnitude by two recipe confounds; see Table 1a for the corrected experiment, which re-ran three of these conditions and preserves both pivotal signs at 7–14% of these sizes. The first row is the warmstart checkpoint scored without any phase-4 training, and it beats every condition** — see "All conditions degrade the warmstart" in the text. The remaining rows are therefore an ordering over how much each allocation *degrades* that checkpoint under this recipe, not over how much each teaches. "Seeds" is the number of training seeds scored for that row; rows with < 3 are exploratory. $\pm$ is the *marginal* document-clustered SEM of the condition mean. Note these marginal SEMs are much larger than the uncertainty on the *differences*: chunks are common to all conditions, so contrasts are paired and their document-clustered CIs are $\sim 3\times$ tighter (e.g. seed-42 uniform $-$ random $= -0.489$, CI $[-0.522, -0.457]$). Do not read overlapping $\pm$ ranges as indistinguishable conditions.

Condition	Signal type	Seeds	Latent $\text{loss}_{\text{with}}$/tok $\downarrow$
warmstart (step 0)	*no phase-4 training*	—	*1.314*
uniform	100% latents (BoLT default)	3	$\mathbf{1.789 \pm 0.075}$
uncertainty	MC spread of $\log(q/p)$ (variance)	1	2.138 ± 0.087
importance_50	bootstrap importance weight w, top 50%	3	2.190 ± 0.088
oracle	measured total train-corpus Δloss	2	2.212 ± 0.090
rho_holdout_50	holdout-reducible benefit (RHO-Loss style)	3	2.212 ± 0.087
length_50	suffix length	2	2.242 ± 0.091
random_50	random (equal-budget control)	3	2.278 ± 0.093
excess_strat	excess demand, within-document rank	1	2.291 ± 0.093
excess_reversed	*low* demand	2	2.299 ± 0.094
demand_mid	mid demand	2	2.313 ± 0.098
agreement	prior–posterior agreement	2	2.461 ± 0.092
excess_50 (ours)	*high* demand	3	2.491 ± 0.097

the high-demand tail generalizes worse than spreading it uniformly (or even randomly), plausibly because the model then trains its latent use on a narrow, high-divergence slice of the distribution. Within the equal-budget 50% family — the only conditions whose training streams are comparable — the two worst conditions are also the two lowest *training* losses (agreement 0.133, excess 0.140, vs. random's 0.238; means over the final 200 steps, since single-step values are noisy — agreement's last step reads 0.100), the signature of fitting a narrow slice. We do not compare against uniform's training loss (0.654), which is not commensurable: it trains on 100% latents and therefore on a different token stream. The agreement signal, our most principled "transferable latent" candidate, is among the worst (2.461, well below random), confirming the failure is not for want of a smarter per-chunk rule — it is that per-chunk benefit, however real and transferable, is the wrong thing to concentrate on. Figure 1b visualizes nine of these conditions.

A direct test. The account above makes a causal prediction, so we test it by removing concentration while holding the signal fixed. `excess_strat` scores each chunk by its excess-demand rank *within its own document*, so the same top-50% cut takes the better half of every document (100% document coverage vs. 95.8%) at an identical token cost and step count. The intervention removes almost all of the harm: demand-routing goes from $+0.212$ (seed 42) worse than the equal-budget random control to $+0.012$, a 94% reduction (`excess_strat` $-$ `excess_50` $= -0.200$, doc-clustered CI $[-0.214, -0.186]$). This is the strongest evidence we have (one seed, damaging recipe) that the failure is one of *allocation geometry* rather than signal quality.

Two qualifications matter. First, de-concentrating makes the signal *harmless*, not useful: `excess_strat` $-$ `random_50` $= +0.012$ with a doc-clustered CI of $[-0.002, +0.026]$ that *includes zero*, so a spread demand signal is statistically indistinguishable from random and still loses to uniform by 0.50. A signal with ρ=0.48 predictive validity converts, at best, into parity with coin-flipping. Second, the two conditions do not select identical sets (their scores correlate at

ρ=0.77), so part of the gap is "which chunks" rather than purely "how spread"; the intervention bounds the concentration effect rather than isolating it exactly.

Allocating by uncertainty, not by benefit, beats the benefit oracle. Every condition above — including the oracle — ranks chunks by an estimate of *expected benefit*. Optimal stratified sampling does not: Neyman allocation weights a stratum by its standard deviation, not its mean. By analogy only — Neyman allocation minimises estimator variance, and nothing guarantees it is a good rule for choosing training data — we add a variance-based heuristic condition that ignores expected benefit entirely and routes by the Monte-Carlo spread of $\log(q/p)$ across the K posterior samples — how *uncertain* the demand estimate is. This signal already existed in our released per-chunk data and had never been used for allocation.

It is the best routed condition in Table 4 (2.138). It beats the equal-budget random control by 0.140 (doc-clustered CI $[0.126, 0.155]$) and, more pointedly, it beats the *measured-benefit oracle* by 0.074 ($[0.061, 0.089]$). A signal that never estimates benefit outperforms one given the true benefit. On one seed under the damaging recipe, this contradicts the natural reading of the previous paragraph — that the oracle upper-bounds per-chunk allocation — and suggests a sharper statement: the oracle bounds *benefit-ranked* allocation only, and benefit-ranking may not be the best use of a per-chunk budget.

It does not, however, disturb the paper's thesis. Uncertainty routing still loses to uniform by 0.349 ($[0.322, 0.377]$) under this damaging recipe, so no per-chunk signal we tested beats uniform there. We report it as a single-seed exploratory condition and as the most promising direction we know of for anyone who must route under a hard budget.

Equal budget equalizes tokens, not data. A latent costs r=2.19$\times$ its suffix, so at a fixed token budget the routed arms see $1.52\times$ more unique text than uniform. That asymmetry is real and fully accounted for in Appendix E, but it is not what drives the result: matching unique-chunk coverage instead of tokens still retains 83% of uniform's advantage, so coverage is not what produces it; this does not separate latent elaboration from warmstart density match.

Robustness. The pivotal contrast is sign-consistent across three seeds ($+0.212/+0.203/+0.189$; pooled $+0.201$, two-component document-plus-seed CI $[0.154, 0.248]$), as is the budget effect ($-0.489/-0.482/-0.478$). Downstream GSM8K/MATH sit near floor at 600 steps and we do not use them as evidence for the allocation ranking. Appendix E also sets out why we read the oracle's failure under the damaging recipe as a property of the allocation *objective* rather than of our training scale.

E ALLOCATION: EXTENDED RESULTS

The ordering is not an *evaluation*-format artifact. Uniform trains on 100% latent-formatted sequences and our metric puts a latent in every val prompt, so one might worry the ranking simply rewards format familiarity. It does not. Sweeping the eval-time latent fraction β from 1 down to 0 by mixing each chunk's with- and without-latent score, uniform's margin over `random_50` (seed 42) moves only from -0.489 to -0.469, and the demand harm from $+0.212$ to $+0.202$. Removing the evaluation latent changes the margin by 4.3%: even when *no* latent appears at evaluation, the ordering and nearly all of its magnitude survive (`results/phase4_eval/eval_format_sweep.json`). This addresses the *evaluation* format only. It does not address the *training* distribution, where the match is real and consequential: the warmstart was fit with a latent in every chunk, so `uniform` alone continues on the distribution the model had just seen. That confound is discussed above and is not ruled out by this sweep.

Document clumping is not the operative narrowness. The most obvious *observational* proxy for narrowness does *not* by itself explain the ranking. Measuring how far each selection clumps within documents, the excess top-50% set is three times more clumped than random (23.0% vs 7.6% of selected chunks sit in documents that are $\geq 80\%$ selected) — but the *importance* selection is clumpier still (38.5%) while being the *best* seeded routed condition. Document-level concentration is therefore a real property of demand routing but not a sufficient statistic for its harm; whatever narrowness matters is semantic rather than positional, and isolating it is open work.

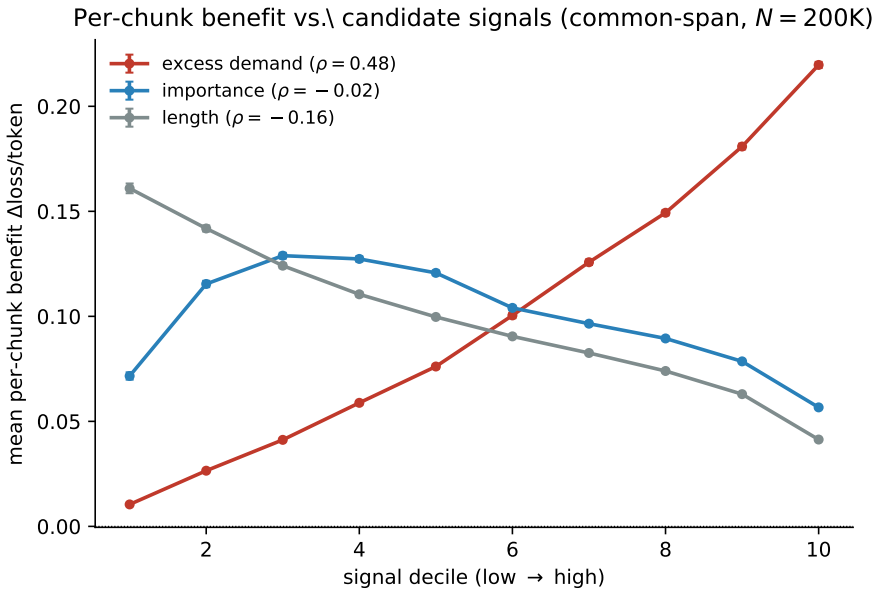

Figure 4: Per-chunk benefit (Δloss) vs. candidate signals on the 200K-chunk corpus (common-span metric). Excess demand attains $\rho = 0.48$, while the importance weight is weakly informative at best ($\rho = -0.02$ raw, 0.14 once length is controlled).

Material summarized in Section 4; the full numbers follow.

Cross-domain (FineWeb-Edu). To test whether the findings are math-specific, we repeat the experiment on a second domain — FineWeb-Edu general web text — generating fresh latents and training five conditions (uniform/random/length/excess/reversed) from the same warmstart, scored under the *same* common-span latent-aware metric on 3000 held-out chunks (span-match $\approx 100\%$; the earlier report of a truncation confound here was an artifact of the naive scorer, which the common-span metric removes). The *budget* finding transfers cleanly: uniform is again decisively best (latent NLL/token 3.719 vs ≥ 4.02 for every routed condition), beating the equal-budget random control (4.107) by 0.388 at seed 42 (document-clustered 95% CI $[0.377, 0.400]$) and by 0.391 pooled over seeds $42/43/44$ (across-seed interval $[-0.434, -0.349]$), and no per-chunk signal beats uniform under the no-replay recipes. The *signal* finding does not transfer, and our first reading of why was wrong. At seed 42 the FineWeb contrast is significantly *negative* (excess $-$ random $= -0.044$, chunk-level 95% CI $[-0.058, -0.032]$) — the opposite sign to math — and we reported that as a domain-dependent sign flip. Replicating it at two further seeds gives $+0.016$ ($[+0.005, +0.026]$) and -0.017 ($[-0.031, -0.003]$). Every one of the three intervals excludes zero and they disagree with each other about the sign. Treating the seed as the unit of replication, the contrast is -0.015 with a 95% interval of $[-0.090, +0.060]$: indistinguishable from no effect.

We therefore withdraw the cross-domain sign flip. The native-FineWeb warmstart run, which we had read as independent confirmation (excess $-$ random $= -0.019$), sits inside this band and is better understood as a fourth draw from a distribution centred near zero. What remains is a weaker but cleaner statement: demand-routing's *harm* is specific to math ($+0.201$ across three seeds, across-seed interval well clear of zero), on general web text no effect of the per-chunk signal is identified, and in neither domain does any signal beat uniform under the no-replay recipes. The *budget* finding is untouched by this and is seeded on FineWeb as well ($-0.388/-0.376/-0.410$ across seeds $42/43/44$).

This is also the paper's own instance of the failure it describes one level up. Three tight, individually significant, chunk-level intervals supported a confident claim about domain structure; the check that dissolved it was two more training runs. Chunk-level intervals answer "would another sample of held-out text agree" and are silent on "would another training run agree", which is the question a claim about domains actually rests on.

What "equal budget" does and does not equalize. The comparison holds the *token* budget fixed, not the amount of unique text seen, and these are not the same thing. Measured directly on the training corpora, a latent costs $r{=}2.19\times$ the suffix it annotates, so a chunk costs 582 tokens under uniform but only 382 under a 50% routing rule. Within the same 39.3M-token budget that is $67{,}513$ unique chunks for uniform against $102{,}866$ for `random_50` — the routed conditions see $\mathbf{1.52}\times$ **more unique text**. Two consequences follow. First, uniform's advantage at 200K is obtained *despite* a 35% coverage handicap, which makes the value of latent elaboration at this corpus size larger, not smaller, than the raw contrast suggests. Second, this is a natural candidate explanation for the loss of uniform's lead at $3\times$ data (Appendix G): elaboration and coverage are traded against each other, and as unique data grows the trade could turn unprofitable.

We therefore ran the control. Holding steps and tokens at 600 and instead *subsampling* the routed arm's corpus to $67{,}676$ chunks makes both arms draw the same number of distinct chunks; the routed arm then repeats them $1.53\times$. Uniform's advantage largely survives: it falls only from -0.489 (seed 42) to -0.408 (doc-clustered CI $[-0.435, -0.381]$), retaining 83%, so coverage is not what produces the margin; this does not separate latent elaboration from warmstart density match. Consistently with a data-constrained reading, the subsampled control also beats the full-corpus `random_50` by 0.082 ($[0.065, 0.099]$) — at this budget, repeating less data beats spreading over more.

We report the accounting because "equal budget" equalizes tokens and not data, and readers should know the routed arms see $1.52\times$ more unique text. But the budget effect is not an artifact of that gap. One caveat: this control replaces the step-count confound that invalidated our first attempt (giving uniform $1.53\times$ the steps moved it $+0.53$ nats along a steep steps-versus-loss curve, swamping the contrast) with a smaller one, repetition; it separates coverage from step count, not from mild repetition. Full accounting: `results/token_accounting.json`, `results/matched_cov/h1_corrected.json`. The budget also counts training tokens only. Demand scoring costs K posterior generations and a prior/posterior scoring pass per chunk ($K{=}4$ on the 200K corpus), which we reused from the bootstrap's own pipeline and did not time end to end; a practitioner adopting routing from scratch pays it, so our efficiency statements are bounded to the training budget and say nothing about the total cost of a routed pipeline.

Why scale would not obviously rescue the premise. A natural objection is that the 600-step budget is too small to conclude. Our reply is conditional. The oracle routes by the measured per-chunk Δloss — the quantity any per-chunk-*benefit* signal aims to maximize, so no realizable such signal can route by it more accurately. That this optimum still loses to uniform says the per-chunk-benefit *objective* is misaligned with what training rewards under this damaging recipe; if that misalignment is structural, larger training changes the magnitude of the uniform–oracle gap and not its sign. But a fixed model's prediction benefit is not an oracle for post-training generalization, the 1800-step cells already leave uniform's edge indistinguishable from zero, and the demand harm did not transfer to FineWeb, so we present this as the reading we find most likely rather than as established: the corrected recipe has not been run at larger scale, and that is the experiment that would settle it.

Robustness (three seeds). The result reproduces across three training seeds. The pivotal contrast — excess demand *worse* than the equal-budget random control — is sign-consistent and tightly clustered at $+0.212/+0.203/+0.189$ nats/token (seeds $42/43/44$; pooled $+0.201$, two-component document-plus-seed 95% CI $[0.154, 0.248]$; the earlier hierarchical interval $[0.186, 0.216]$ divided the evaluation term by the seed count and is superseded), and the budget effect is likewise stable (uniform $-$ random $= -0.489/-0.482/-0.478$). The uniform-best, demand-routing-worst ordering is therefore not a single-seed artifact. Downstream GSM8K/MATH accuracy is *underpowered* here (harness limit $= 250$ examples/task, single seed) and uninformative at this scale (all near floor: GSM8K 0–0.8%, MATH 2–3%); the 600-step models sit below task floor, so we do not use downstream accuracy as evidence for the allocation ranking.

F The Budget Effect Decays with the Step Budget

This appendix has been rewritten twice, and the history is worth stating because it is the same failure the paper is about, one level down.

Table 5: The budget-axis 2×2, one control (`random_50`), common-span metric; N_{uniq} uses rounded per-chunk cost (the direct count at 200K/600 is 67,513). The 200K/600 cell is seed 42 of the main experiment plus two re-trained seeds (mean of per-seed contrasts -0.494; the main experiment's pooled estimate is -0.483) and both 1800-step cells carry three seeds, with across-seed intervals (t, 2 d.f.); the 600K/600 cell is a single seed. Neither 1800-step cell has an identified sign, so the sign change we previously reported between them is not established.

Corpus	Steps	N_{uniq}	seeds	uniform − random	interval
200K	600	67,516	3	-0.494	$[-0.521, -0.466]$
600K	600	67,516	1	-0.414	—
200K	1800	200,003	3	-0.061	$[-0.122, +0.001]$
600K	1800	202,549	3	$+0.014$	$[-0.158, +0.185]$

Version 1 (single axis). Because a latent costs r=2.19× its suffix, the uniform arm affords $N_{\text{uniq}} = B/(c(1+r))$ unique chunks at token budget B and every 50% rule affords $\approx 1.52\times$ more. Plotting the measured gap against N_{uniq} collapsed our three scale experiments onto one decreasing curve crossing zero near 181K–209K chunks. We offered that as a falsifiable prediction.

Version 2 (two axes). Completing the 2×2 appeared to refute the single axis: at seed 42, the two 1800-step cells had N_{uniq} within 1.3% of each other yet differed by 0.129 and changed sign, which we read as corpus size contributing a second term.

Version 3, and what we now report. Seeding the 1800-step row three times dissolves that reading too. Table 5 gives the current state.

What survives. One effect, and it is large. Tripling the step budget moves the contrast by $+0.433$ at 200K (now three seeds at both ends) and $+0.428$ at 600K — the same number at two corpus sizes, and far larger than any interval in the table. **Uniform's advantage is no longer identified as the step budget grows**: decisive at 600 steps, an interval spanning zero at 1800 under either corpus.

What does not. Whether it goes on to *reverse* is not established. The 600K/1800 cell that read $+0.055$ at seed 42 gives -0.066 at seed 43 and $+0.053$ at seed 44; its across-seed interval spans $[-0.158, +0.185]$. Nor is the corpus term: the gap between the two 1800-step cells falls from 0.129 at seed 42 to $+0.074$ across seeds, with the two intervals overlapping almost entirely. We can no longer say that corpus size moves the contrast at fixed step budget, and we withdraw the sign change.

Why we report the history. Three times in this work a tight, individually significant, single-seed result supported a confident structural claim — a cross-domain sign flip, a single unique-data axis, and then a corpus term — and three times two more training runs dissolved it. Every one of those claims had a chunk-level interval excluding zero. That statistic answers "would another sample of held-out text agree"; each claim actually rested on "would another training run agree", which it does not address. This is the step-0 lesson in a different coordinate: a robustness check can be impeccable and still be checking something other than what the claim needs. A fourth instance, of a different kind, is in Appendix I: a correlation that three seeds agreed on dissolved when the number of *conditions* was doubled, because seeds were never the narrow sample there — and its weakened form then failed to agree across three public grids.

Caveats. The 200K/600 cell now carries three seeds, seed 42 plus two re-trained (-0.494, $[-0.521, -0.466]$, against the main experiment's pooled -0.483); the 600K/600 cell is still one, so the 600K column's decay inherits a single-seed endpoint. The 600K cells are scored on a held-out set drawn from the 600K corpus and the 200K cells on the original one; rescoring 600K/600 on both moves the contrast by 0.010, so the val set is not driving the pattern. All magnitudes here are recipe-inflated in the sense of Section 4. The decay is consistent with the data-constrained scaling regime (Muennighoff et al., 2023), where the value of augmenting data falls as unique data grows, and with Goyal et al. (2024) on selection rules weakening as high-quality data is repeated under growing compute. Sorscher et al. (2022) predict instead a reversal in *which* rule is right — hard examples when initial data is abundant, easy ones when it is scarce — which

our budgets cannot confirm either way. (`results/budget_axis_2x2/summary.json`, `results/seed_hardening/summary.json`.)

G Robustness to Scale: Full Results

These three scalings are summarized in Section 4; the full numbers follow. All three are scored on the same frozen common-span metric (n=3000, zero span mismatches) as every other result in this paper, so no comparison in the paper mixes metrics.

Scale robustness ($3\times$ compute). Retraining the three pivotal conditions at 1800 steps ($\sim$118M tokens, $3\times$ the budget) and scoring them on the *same* common-span metric (n=3000) preserves the ordering uniform (3.092) < importance_50 (3.107) < excess_50 (3.254). Absolute losses rise uniformly ($\sim+1.3$ nats/token vs. Table 4). We previously attributed that rise to overtraining over the fixed corpus; that explanation is wrong, because at 1800 steps the uniform arm still covers only $\approx$1.01 epochs of the 200K-chunk corpus. The rise is instead part of a monotone degradation of held-out loss with training steps in this setup ($\approx$1.09 nats per 1000 steps at fixed allocation), which we flag as an identification problem in Section 4 rather than explain away. The paired contrasts nonetheless stay large and significant: excess exceeds uniform by 0.163 (paired 95% CI $[0.147, 0.178]$) and even the practical importance signal is significantly worse than uniform ($+0.015$). The demand-routing-worst position is therefore not a small-compute artifact; uniform's lead over random is (Appendix F).

Model scale ($2.5\times$ parameters). We repeat the three pivotal conditions with a 2.7B consumer model (Sheared-LLaMA-2.7B) in place of the 1.1B one, holding the latent corpus *and* the demand signal fixed at their 1.1B values, so that only the model consuming the allocated latents changes. The base vocabulary is byte-identical to the 1.1B tokenizer, so per-token NLL is directly comparable, and the common-span metric matches spans for all 3000 chunks at both scales (zero span mismatches, zero latent overflows). We run **three training seeds** (42/43/44) and pool with the same two-level document-clustered bootstrap used at 1.1B. The ordering replicates in every seed: uniform (1.804/1.770/1.784) < random_50 (2.127/2.175/2.183) < excess_50 (2.367/2.391/2.367). Both pivotal contrasts keep their sign in all three — uniform beats equal-budget random by a pooled -0.376 (two-component document-plus-seed CI $[-0.500, -0.251]$) and demand-routing remains *worse* than that same control by $+0.213$ ($[+0.134, +0.293]$).

Comparing pooled to pooled, the budget advantage *narrows* with model scale ($-0.483 \rightarrow -0.376$), echoing the data-scale finding below, while the harm from demand-routing is essentially *unchanged* ($+0.201 \rightarrow +0.213$). Seed variation matters here: seed 42 alone would have suggested a much sharper narrowing (-0.323) and a growing harm ($+0.240$), and both readings shrink once all three seeds are in. The stable statement is that the demand signal's anti-usefulness is scale-invariant while uniform's budget edge softens. We caution that this varies consumer-model scale only; whether D_i re-estimated by a 2.7B posterior would rank chunks differently is a separate question we do not test here.

Data scale ($3\times$ corpus, matched epochs). The decisive scale test grows the *data*, not the epochs: 1800 steps over a fresh 600K-chunk corpus ($3\times$ the data at the *same* $\sim$1-epoch exposure as the 600-step/200K run), evaluated on a leakage-free 600K held-out set. We score it on the same frozen common-span metric as everything else (n=3000, zero span mismatches, zero latent overflows); an earlier version of this experiment used the legacy truncation-safe subset (n=671) and we report the clean numbers here.

The core result is robust: excess remains *worst* (3.227) and significantly worse than the equal-budget random control (3.049) by $+0.178$ (paired 95% CI $[+0.160, +0.197]$), and no signal-based condition beats random — importance is also worse ($+0.105$).

One headline does not survive this budget, and our reading of how it fails has itself been corrected twice. On the clean metric at seed 42, uniform came out significantly *worse* than the equal-budget random control ($+0.055$, paired 95% CI $[+0.024, +0.085]$), and we reported that uniform's advantage *reverses* at $3\times$ data. Replicating the cell at two further seeds gives -0.066 and $+0.053$: across seeds the contrast is $+0.014$ with an interval of $[-0.158, +0.185]$. The reversal is withdrawn. What the three seeds do support is that uniform's advantage is no longer identified at this budget: -0.489 at 600

steps (seed 42), an interval spanning zero at 1800 that still admits effects of either sign (Appendix F). What does *not* change with data is the signal result: demand-based routing is worse than random at both scales.

H REPORTING-PRACTICE SURVEY

Three of the objections we anticipated reduce to one: is the failure mode we describe a property of this literature, or of our own experiment? Asserting the former without measuring it would be an $n{=}1$ claim, so we coded a sample. The protocol (inclusion criteria, coded fields, rules of evidence) is fixed in `docs/survey_protocol.md` and the per-paper coding with quotes in `results/survey/rows.jsonl`.

Criteria. A paper is included if it (i) compares two or more data recipes by training and comparing a metric, (ii) continues from an already-trained checkpoint — from-scratch comparisons are excluded, since there step 0 is a random init and the baseline is vacuous — and (iii) appeared from 2023 onward. Papers were selected by keyword search before their coding was known, and none was dropped after coding. Every row was coded from fetched full text; nothing from titles.

Table 6: Twenty-five continued-pretraining data-recipe comparisons, coded for whether the starting checkpoint appears on the metric that ranks the recipes. *yes* = an explicit value; *partial* = a figure point or a qualitative remark only. *LR sens.* codes any learning-rate sweep or sensitivity analysis. The rule separates loss-ranked papers (top) from downstream-ranked.

arXiv	paper	ranks by	step-0	LR sens.
2308.04014	Gupta et al., how to (re)warm…	held-out loss	partial	yes
2402.17400	Investigating Continual Pretra…	held-out loss	yes	no
2403.08763	Ibrahim et al., simple and sca…	held-out loss	yes	yes
2403.16952	Data Mixing Laws (continual-PT…	validation loss	partial	no
2406.01375	D-CPT Law (domain mixture-rati…	validation loss	yes	no
2406.14971	Llama3-70B domain adaptation (…	held-out loss	partial	no
2407.17467	CMR Scaling Law (CPT mixture r…	held-out loss	partial	no
2408.02600	BioMamba (biomedical DAPT)	held-out loss	yes	no
2509.25380	TREC data curriculums (CPT sec…	held-out loss	partial	yes
2603.16177	The Finetuner's Fallacy	held-out loss	partial	partial
2311.08545	ETS/ETA-DACP (FinPythia)	downstream	yes	partial
2402.07625	AutoMathText (autonomous data…	downstream	yes	no
2406.14833	Mitigating the Stability Gap i…	downstream	yes	yes
2407.07263	Reuse, Don't Retrain (NVIDIA 1…	downstream	yes	yes
2501.00656	OLMo 2 (microanneal mixture ab…	downstream	yes	partial
2502.19279	CritiQ (data quality criteria)	downstream	yes	no
2504.09687	Domain-Adaptive Continued PT o…	downstream	yes	no
2505.02881	Rewriting Pre-Training Data (m…	downstream	yes	no
2506.07658	Raw Corpora to Domain Benchmar…	downstream	partial	no
2506.12966	Data Quality in Bilingual LM T…	downstream	partial	no
2507.15640	Data Mixing Agent (continual P…	downstream	yes	no
2510.05858	DACP (phone-conversation summa…	downstream	yes	partial
2510.10071	ADEPT (CPT via adaptive expans…	downstream	yes	partial
2512.21515	Perplexity-Aware Data Scaling…	downstream	yes	no
2601.17858	MergeMix (mid-training data mi…	downstream	no	partial

Result, including the part against us. Twenty-five papers met the criteria (Table 6), six were screened out as from-scratch, and two could not be resolved from the text we could fetch. **Twenty-four of the twenty-five report the starting checkpoint in some form** — sixteen as an explicit value, eight only graphically or qualitatively, one not at all. Our earlier framing, that these baselines are omitted by default, is not supported and we have withdrawn it. Two narrower findings survive, and they are of different strengths.

First, the *form* of the baseline tracks the ranking metric. Of the ten papers ranked by held-out loss, four give the start as an explicit value; of the fifteen ranked by downstream accuracy, twelve do. The

loss-ranked shortfall is a figure point or a sentence ("the red point represents the perplexity before starting"; "similar performance compared to the model before continual pretraining") rather than a number to difference against — enough to see a trend and not enough to notice that *every* condition lost, which is the specific way our own comparison misled us. But $4/10$ against $12/15$ is Fisher $p = 0.087$, so this is a pattern we can point at and not one the sample establishes. We state it that way deliberately: a paper about over-reading an ordering should not over-read its own. Two of the four explicit cases also show what the convention buys. Ibrahim et al. (2024) report a 0%-replay run ending at 3.56 against a 2.17 start, a condition visibly below its own step 0; and the D-CPT law (arXiv:2406.01375) does not merely report the initial loss but fits it as a parameter, $L_g^0 = 2.8602$.

Second, **none of the ten loss-ranked papers reports a displacement diagnostic**. We searched the full text of each of the ten (the rows coded `no` in `reports_displacement`) for any weight-change measurement, and none contains one. The remaining fifteen rows were coded before the field existed and are marked `not_asked` rather than counted as absent. This finding is not an artifact of our sample either: neither of two independent surveys of this literature (Mo et al., 2025, and arXiv:2510.23081) mentions a weight-change diagnostic. This measurement does appear to be absent from the practice; we do not recommend it as a check (Appendix I).

Limits. Twenty-five papers found by keyword search is a convenience sample, not a systematic review, and the loss-ranked cell holds ten. We report denominators rather than percentages for that reason, and the one comparison we do make between cells carries its p-value above. The displacement field (not a table column) is reported over the ten papers actually checked, never over all twenty-five. The sample also cannot speak to papers whose comparisons we could not classify: two are listed in `unresolved.jsonl` rather than silently dropped.

I Clean-Room Replication

Every training run in the body shares one stack: one trainer, one warmstart family, one tokenizer, one corpus lineage. A reader is entitled to ask whether the step-0 pathology is a property of continued pretraining or of our recipe. We therefore rebuilt the smallest version of the experiment with nothing in common but the question.

Setup. A public converged checkpoint (Pythia-410m, trained on the Pile), a public corpus (FineWeb-Edu), HuggingFace Transformers in place of our trainer, the NeoX tokenizer in place of Llama's, and plain PyTorch. Twelve selection rules — `all` (the uniform arm), `random25/50/75`, `shortest25/50`, `longest25/50`, `highppl25/50`, `lowppl50` and `midppl50` — each continue from the same checkpoint for 400 steps on the same token budget, so only *which* documents supply those tokens differs. Two learning rates (10^{-4}, 10^{-5}) and three seeds: 72 runs. Six of the rules were added after a first pass at six, because six conditions could not make either displacement correlation significant and that correlation was then a claim in our abstract. Two held-out sets are scored, at step 0 and at the end: FineWeb-Edu, the training distribution, and WikiText-103, standing in for what the checkpoint already knew — the upstream/downstream split Gupta et al. (2023) and Ibrahim et al. (2024) both report. Code and per-run artifacts are in `cleanroom/` and `results/cleanroom/`.

Precision, found after submission review. The trainer as first written loaded the checkpoint in bf16 and handed those weights to AdamW, so weights, gradients and optimizer state were all stored in bf16 — unlike our main runs, whose FSDP2 mixed precision keeps fp32 master weights and optimizer state. In bf16 an update smaller than the weight's rounding step is lost: after 20 steps at 10^{-5} on Pythia-160m, 94% of parameters had not changed at all, against 0.004% with fp32 master weights (`results/cleanroom/precision_smoke/`). We therefore reran every grid with the *same* bf16 checkpoint promoted to fp32 before training, bf16 autocast for the arithmetic, and fp32 weights and optimizer state (`run.py -precision fp32master`): all 72 runs for both FineWeb grids and seed 42 for WikiText. Table 8 sets the two side by side (`results/cleanroom/precision_control.json`). Three things survive and one does not. The learning-rate check survives, weaker: a tenfold lower rate removes 86–92% of the between-condition spread rather than 81–99%, and at 10^{-5} conditions differ by up to 0.08, not 0.004. Damage on the axis the checkpoint was not trained on survives in every grid ($12/12$ at 10^{-4}), and is larger. The displacement correlation's inconsistency survives ($+0.92$, -0.65, $+0.58$). What does not survive

is the headroom story of the 410m FineWeb grid below: with fp32 updates all twelve conditions are worse than step 0 in-distribution as well, not seven. The bf16 numbers are kept in Table 7 and in the paragraphs that follow because they are what the first version reported; where they differ from fp32, fp32 is the result.

Table 7: Clean-room replication (original bf16-storage grid; see Table 8 for the fp32 control): Pythia-410m continued on FineWeb-Edu, 400 steps, twelve selection rules, three seeds (72 runs), means with across-seed 95% intervals on the in-distribution delta. *fineweb* is the training distribution, where this checkpoint has real headroom; *upstream* is WikiText-103, which it already knew. Sorted by in-distribution loss within each learning rate.

condition	epochs	fineweb NLL	Δ vs step-0	upstream Δ	$\|\Delta\theta\|$
lr $= 10^{-4}$ *(the standard restart)*					
`all`	0.56	2.9015	-0.0034 $[-0.0082, +0.0014]$	$+0.0504$	16.52
`random50`	1.13	2.9023	-0.0025 $[-0.0082, +0.0031]$	$+0.0459$	16.66
`random75`	0.75	2.9027	-0.0022 $[-0.0069, +0.0025]$	$+0.0476$	16.56
`lowppl50`	1.02	2.9037	-0.0011 $[-0.0020, -0.0002]$	$+0.0962$	16.40
`longest50`	0.69	2.9048	-0.0001 $[-0.0031, +0.0029]$	$+0.0688$	16.54
`highppl50`	1.25	2.9066	$+0.0018$ $[-0.0015, +0.0050]$	$+0.0407$	17.04
`midppl50`	1.07	2.9082	$+0.0033$ $[-0.0100, +0.0167]$	$+0.0833$	16.68
`longest25`	0.93	2.9213	$+0.0164$ $[-0.0547, +0.0876]$	$+0.1003$	16.22
`random25`	2.27	2.9290	$+0.0241$ $[+0.0174, +0.0309]$	$+0.1068$	16.98
`highppl25`	2.86	2.9633	$+0.0584$ $[+0.0556, +0.0612]$	$+0.1034$	18.06
`shortest50`	3.10	2.9717	$+0.0669$ $[+0.0578, +0.0760]$	$+0.1208$	17.77
`shortest25`	9.58	3.3706	$+0.4658$ $[+0.3776, +0.5540]$	$+0.5218$	21.46
lr $= 10^{-5}$					
`random50`	1.13	2.8759	-0.0290 $[-0.0295, -0.0284]$	-0.0241	0.77
`shortest50`	3.10	2.8760	-0.0288 $[-0.0294, -0.0282]$	-0.0271	0.91
`random75`	0.75	2.8762	-0.0287 $[-0.0293, -0.0280]$	-0.0234	0.76
`all`	0.56	2.8762	-0.0286 $[-0.0289, -0.0283]$	-0.0235	0.76
`random25`	2.27	2.8762	-0.0286 $[-0.0292, -0.0280]$	-0.0248	0.78
`midppl50`	1.07	2.8763	-0.0285 $[-0.0293, -0.0278]$	-0.0212	0.76
`highppl50`	1.25	2.8764	-0.0284 $[-0.0288, -0.0280]$	-0.0298	0.80
`longest50`	0.69	2.8767	-0.0281 $[-0.0290, -0.0272]$	-0.0186	0.75
`lowppl50`	1.02	2.8768	-0.0281 $[-0.0283, -0.0278]$	-0.0191	0.75
`highppl25`	2.86	2.8771	-0.0277 $[-0.0284, -0.0271]$	-0.0348	0.88
`longest25`	0.93	2.8776	-0.0273 $[-0.0278, -0.0267]$	-0.0173	0.73
`shortest25`	9.58	2.8798	-0.0250 $[-0.0255, -0.0245]$	-0.0308	1.22

The learning-rate diagnostic replicates, and more sharply. Table 7 carries the full grid. Dropping the learning rate tenfold removes 99% of the between-condition spread in bf16 ($0.469 \rightarrow 0.004$ nats/token) and 92% with fp32 master weights ($1.038 \rightarrow 0.082$); the figure in our own experiment was 92%. At 10^{-4} a study would report a clean ordering spanning 0.47 with `all` best and `shortest25` worst; at 10^{-5} all twelve conditions lie inside 0.004 of each other in bf16 — partly because many updates were rounded away — and inside 0.08 in fp32. Whatever that ordering is about, it is not about which documents teach the model more.

The damage replicates on what the checkpoint already knew. At 10^{-4}, **all twelve conditions are worse than step 0 on the upstream set** (bf16; $+0.041$ to $+0.522$; mean damage doubles with fp32 updates), and eleven of the twelve across-seed intervals exclude zero. That is the paper's claim, reproduced in a stack that shares no code with ours.

What did not replicate in bf16, and does in fp32. On the *in-distribution* set only seven of twelve bf16 conditions are worse than step 0; with fp32 master weights all twelve are. This is the regime difference, and we built the experiment to expose it: Pythia was trained on the Pile and has never seen FineWeb-Edu, so continuing it there has real headroom, and adaptation gain partly masks restart damage. Our own warmstart had no such headroom, which is consistent with every no-replay condition there landing above its start. So "every condition is worse than step 0" is not a law of continued pretraining; it is what one should expect when the checkpoint has little left to gain from

the corpus — and the only way to know which regime one is in is to score step 0, on both axes. With fp32 updates this grid no longer shows the masking — the damage is visible on both axes — but the WikiText grid does, in either precision: all twelve conditions improve on the corpus trained on while all twelve lose ground on FineWeb-Edu.

Note also what the in-distribution ordering at 10^{-4} is made of: the two conditions that traverse their subset most often, `shortest25` and `shortest50`, account for the whole bf16 spread at $+0.466$ and $+0.067$, while five of the remaining ten sit within 0.005 of each other (seven within 0.007). The ranking a paper would publish is a concentration effect and noise.

Displacement, and a reversal that did not survive. At 10^{-4}, displacement correlates with held-out loss (bf16) at $\rho = +0.69$ (permutation $p = 0.019$; Fisher-z 95% CI $[+0.16, +0.91]$; leave-one-condition-out range $+0.59$ to $+0.85$; per-seed $+0.88$, $+0.82$, $+0.53$) — the same direction as our $+0.94$ (CI $[+0.83, +0.98]$), and significant at twelve conditions. In bf16, at 10^{-5}, where every run improves on its start, the correlation is $\rho = -0.07$ ($p = 0.84$; CI $[-0.63, +0.54]$; per-seed -0.08, -0.06, -0.04): not distinguishable from zero at twelve conditions, and not shown to be zero either — at $n = 12$ even a true zero gives an interval wider than ± 0.5, so no equivalence claim is available. With fp32 master weights one condition ends above step 0 at this rate and the correlation is $+0.61$ ($p = 0.036$). The two rates do differ (exact paired permutation of the rate label within condition, on within-rate ranks: $p = 0.006$).

We report this paragraph twice over, because the first version was wrong in the way this paper is about. At six conditions the low-rate cell read $\rho = -0.66$ ($p = 0.16$), every seed agreed on the sign, and we wrote that the correlation *reverses* — that its sign was itself a regime detector — and put that in the abstract. Doubling to twelve conditions moved it to -0.07. Three seeds agreeing inside a six-point sample was not evidence of a reversal; it was the same over-reading of a tight-looking ordering that the step-0 baseline exists to catch, committed by us, in the appendix that argues against it. We then wrote that what survives is presence: displacement predicts loss where runs are damaged and stops predicting where they learn. Two more public grids (below) took that away as well — at the damaging rate the correlation is $+0.69$, -0.46 and $+0.59$ across the three in bf16, and $+0.92$, -0.65 and $+0.58$ with fp32 master weights. We withdraw the sign of this correlation as a regime diagnostic altogether. The twelve-condition permutation $p = 0.019$ should also be read as exploratory rather than confirmatory: the six further rules were chosen after the six-condition result had been seen, and the two further grids, run afterwards with the hypothesis fixed, are the confirmation — which did not confirm it.

What we can still say for the observation is narrower. Our corrected experiment runs at the same 10^{-5} with every run still above its step-0, and there the correlation is $+0.87$ over nine runs. Those nine points are three conditions $\times$ three seeds, so as condition-level evidence this is $n = 3$ — each seed's three conditions rank exactly, which three conditions make easy — and we attach no interval or p-value to it. Within one experiment it tracked our ordering twice. It is not, on the evidence of three public grids, a rule about regimes.

Equal budget, unequal data. The clean room reproduces the budget asymmetry too. Holding tokens fixed does not hold data fixed: `shortest25` traverses its quarter of the corpus 9.58 times and `shortest50` its half 3.10 times, while `all` sees 0.56 of one epoch — and the two that traverse most are exactly the two that move furthest (bf16 $\|\Delta\theta\| = 21.46$ and 17.77) and end worst. Concentration of budget, not the content of the signal, is again what the ordering tracks.

Two more public grids. To take the claims outside a single model and corpus we ran the same twelve rules with Pythia-160m on FineWeb-Edu and with Pythia-410m on WikiText-103, 72 runs each (Table 8). The learning-rate check reproduces in both (87% and 86% of the spread removed with fp32 master weights; 98% and 84% in bf16), and in every grid at the damaging rate all twelve conditions lose ground on the axis the checkpoint was not trained on. Both FineWeb grids lose on the training axis too with fp32 updates ($12/12$), and the 160m grid does even in bf16 — the paper's original finding, in a public stack. WikiText was chosen as a no-headroom corpus and turned out not to be one: every condition improves on it, so the no-headroom regime remains mapped only by our own warmstart. The displacement correlation is where the generalisation ends: in bf16 -0.46 at 160m (CI $[-0.82, +0.17]$, $p = 0.14$) and $+0.59$ on WikiText (three seeds, $p = 0.048$); with fp32 master weights -0.65 ($p = 0.022$) and $+0.58$, and nothing across the three we would call a pattern; the six rules added

later to the 410m grid behave like the first six (their bf16 correlations are $+0.71$ and $+0.83$). We do not test differences between these correlations: the grids share the same twelve rules and partly the same data, so they are not independent samples. The other axis is no more consistent (bf16): at 410m and 10^{-5} displacement correlates with the WikiText loss at -0.94 (stable under leave-one-out), against -0.36 and $+0.59$ in the same cell of the other two grids. Full intervals, permutation tests and leave-one-condition-out ranges: `results/displacement_rho_robustness.json`. It tracked our ordering; it is not a regime detector or a recommended check.

Table 8: Three public clean-room grids in both precisions: bf16 storage (the original grids) and fp32 master weights and optimizer state from the same starting values. *train* is the held-out set of the corpus trained on; *other* is the axis the checkpoint was not trained on here (WikiText-103 for the FineWeb grids, FineWeb-Edu for the WikiText grid). "worse" counts conditions above their own step-0 out of twelve; ρ is the Spearman correlation of displacement with training-axis loss over condition means at 10^{-4}; the last column is the drop in between-condition spread from 10^{-4} to 10^{-5}. Seeds are those present in both precisions; over its three bf16 seeds the WikiText grid gives $\rho = +0.59$ and 84%.

				worse at 10^{-4}		worse at 10^{-5}			spread
base	corpus	storage	seeds	train	other	train	other	ρ	removed
410m	FineWeb	bf16	3	7/12	12/12	0/12	0/12	$+0.69$	99%
410m	FineWeb	fp32	3	12/12	12/12	1/12	1/12	$+0.92$	92%
160m	FineWeb	bf16	3	12/12	12/12	0/12	0/12	-0.46	98%
160m	FineWeb	fp32	3	12/12	12/12	2/12	12/12	-0.65	87%
410m	WikiText	bf16	1	0/12	12/12	0/12	12/12	$+0.44$	81%
410m	WikiText	fp32	1	0/12	12/12	0/12	12/12	$+0.58$	86%

Limits. Twelve conditions, 400 steps and models of 160m–410m are small, and WikiText did not give us the no-headroom regime we wanted. Doubling the conditions made one correlation significant and dissolved another, and two more grids then took the survivor away — a fair warning about how much any single grid is worth. What this experiment establishes is narrow and, we think, the part that most needed establishing: the learning-rate artifact and the upstream damage are not ours.

J Do the Two Remedies Compose?

Replay shrinks the damage to about $+0.05$ at 75–100%; a low rate without replay leaves $+0.024$. If the two remedies are not redundant, combining them should do better than either, and the corner of the grid should reach the checkpoint it started from; the grid is not complete enough to separate their effects from their interaction. Table 9 is that grid.

Table 9: Replay ratio $\times$ learning rate, `uniform`, 600 steps, common-span metric, against a step-0 of 1.3145. Loss falls with the replay ratio at each rate except between 75% and 100% at 10^{-4} (1.3620 vs. 1.3654), and only the corner crosses zero.

configuration	loss	vs. step-0
50% replay, lr=10^{-5}	**1.3113**	$\mathbf{-0.0032}$
25% replay, lr=10^{-5}	1.3169	$+0.0024$
0% replay, lr=10^{-5}	1.3388	$+0.0243$
75% replay, lr=10^{-4}	1.3620	$+0.0475$
100% replay, lr=10^{-4}	1.3654	$+0.0509$
50% replay, lr=10^{-4}	1.3707	$+0.0562$
25% replay, lr=10^{-4}	1.4376	$+0.1231$
10% replay, lr=10^{-4}	1.5372	$+0.2227$
0% replay, lr=10^{-4}	1.7888	$+0.4743$

They compose, and the corner learns. At 50% replay and 10^{-5} the condition ends 0.0032 *below* the checkpoint it started from, with a document-clustered interval of $[-0.0044, -0.0019]$, better on 58.3% of chunks. It is the only cell of this grid that improves on its start point.

What it is worth, and what it is not. The margin is 0.003 against the 0.474 of damage the two remedies removed, so this is not a recipe that makes continued pretraining productive here — it is the point at which the comparison stops measuring damage. That is the whole claim: the design is not inherently a damage ranking, it is a damage ranking at every setting we tried except replay combined with the low rate, and two known remedies move it across the line. A reader who adopts our two checks and finds step-0 winning now has somewhere to go.

Seeded. The margin is small enough to sit in the band where three of our own single-seed claims have already reversed, so we replicated it — twice. Across seeds $42/43/44$ the crossing configuration gives 1.31128, 1.31128, 1.31115 against a step-0 of 1.31450; seeds $45/46$, run after this appendix was first written, add 1.31136 and 1.31162. Pooled: mean -0.00316, across-seed interval $[-0.00338, -0.00294]$, **all five below**. The seed range grew with the sample — 1.4×10^{-4} at three seeds, 4.7×10^{-4} at five — a reminder that tight early spreads flatter; but the worst seed is still 0.0029 below step-0 and the interval stays clear of zero. At 10^{-5} with half the stream replayed the model barely moves and different seeds land in nearly the same place. The training trajectories are not close (step-10 loss 1.513 vs 0.928); only the destinations are.

The allocation comparison where training helps. We ran `random_50` and `excess_50` under the corner recipe, three seeds each (`scripts/protocol_3way.sh`; `results/protocol_3way/`), and then all three conditions again from the 75%-density warmstart with 50% replay of *its* corpus (`scripts/protocol_3way_ws75.sh`; `results/protocol_3way_ws75/`), because in the first version only `uniform`'s mixture (100% latent) matched the warmstart. Table 1b,c gives both. Every one of the twenty runs improves on its step 0. On the evaluation set, demand-routing loses to random in both recipes ($+0.0030$, $+0.0034$), and random is ahead of uniform in all six seeds — significantly when uniform's mixture matches the start ($+0.0026$), not when density is matched ($+0.0013$, two-component $[-0.001, +0.004]$), so part of the first gap was density. Uniform and demand-routing are not distinguishable in either recipe.

An independent split. The replay and rate were chosen on that evaluation set, so we fixed the two contrasts in advance and rescored every protocol checkpoint on `phase4_val_600k`, a held-out set from a different corpus slice never used for any choice (`scripts/protocol_split_check.sh`; `results/protocol_split_check/summary.json`). Every run again improves on its own step 0 (eighteen of eighteen). Recipe (b) replicates: uniform $-$ random $+0.0030$ (across-seed $[+0.001, +0.005]$) and demand $-$ random $+0.0029$ ($[+0.001, +0.005]$). Recipe (c) does not sharpen: uniform $-$ random $+0.0014$ ($[-0.002, +0.005]$) and demand $-$ random $+0.0011$ ($[-0.002, +0.004]$), one seed of each slightly negative. So the density-matched demand harm seen on the evaluation set is not established. What both splits support is narrower and, for this paper, sufficient: where training helps, every allocation improves on its start, nothing beats random routing, and uniform, the damage-regime winner, is no longer first. We do not read these differences as a recipe.

The rate $\times$ replay grid. The corner recipe changes two things at once, so we filled in the 2×2 of learning rate (10^{-4}, 10^{-5}) and replay (0%, 50%) from the original 100%-density warmstart for all three conditions, three seeds per cell, reusing cells already run under the identical recipe (`scripts/lr_replay_grid.sh`; `results/lr_replay_grid/summary.json`). Table 10 gives the result. Only the cell with both remedies improves on its start. Uniform leads random in the three cells that damage the start and trails it only in that one; each remedy alone shrinks uniform's lead from 0.48 to 0.01–0.03 without changing its sign. Demand-routing trails random in all four cells, every across-seed interval clear of zero, which makes it the most stable allocation result in the paper. With four cells, and the sign change confined to the one cell that also crosses step 0, this describes where the ranking changes, not why.

Other random masks. Every protocol run used one stored `random_50` mask (mask seed 42); training seeds varied, the mask did not. We drew two fresh masks (mask seeds 43, 44; each agrees with the first on 50% of chunks, as independent masks should) and trained each with training seeds 42–44 under recipe (b) (`scripts/protocol_masks.sh`;

Table 10: Rate × replay from the 100%-density warmstart, 600 steps, three seeds per cell; contrasts are paired by seed with across-seed 95% t intervals. "improve" counts runs below their step-0 (1.3145).

replay, rate	uniform vs. step-0	uniform − random	excess − random	improve
0%, 10^{-4}	$+0.470$	-0.483 $[-0.497, -0.469]$	$+0.201$ $[+0.171, +0.231]$	0/9
0%, 10^{-5}	$+0.024$	-0.030 $[-0.032, -0.028]$	$+0.027$ $[+0.024, +0.030]$	0/9
50%, 10^{-4}	$+0.057$	-0.014 $[-0.022, -0.006]$	$+0.028$ $[+0.027, +0.029]$	0/9
50%, 10^{-5}	-0.0033	$+0.0026$ $[+0.002, +0.004]$	$+0.0030$ $[+0.001, +0.005]$	9/9

`results/protocol_masks/summary.json`). All nine runs improve on step 0 and all nine are ahead of uniform: by $+0.0026$, $+0.0032$ and $+0.0064$ for the three masks. The first mask was the most conservative of the three. Treating the mask as the unit, the across-mask interval ($[-0.001, +0.009]$, three masks) still spans zero, because which chunks a random rule picks moves the result by up to 0.004 — more than the training seed does.

What "50% replay" means in tokens. The data loader picks a source per document and only then tokenises, so the $0.5/0.5$ source weights split documents, not tokens or chunks. Rendering 3000 sampled documents per source exactly as the routed loader does (`scripts/replay_token_accounting.py`; `results/replay_token_accounting.json`): `uniform` documents carry 5661 tokens, the 50% rules' 3797–3799 (half their chunks have no latent), and the replay corpora 5132 (100%) and 4252 (75%). The realised replay token share is therefore 47.6% with `uniform` and 57.5% with either 50% rule from the 100% warmstart, and 42.9% against 52.8% from the 75% one. The realised chunk-level latent density of recipe (c) is 88.0% against 62.3%, equidistant from the 75% start as intended. Two consequences: the `excess_50−random_50` contrast is matched in both density and replay exposure, and the `uniform−random_50` contrast in either protocol recipe also differs by about ten points of replay tokens, which on the replay sweep's evidence favours the 50% rules. Uniform losing its lead is therefore not attributable to allocation alone.

K Extended Related Work and Positioning

We position ThinkGap against the nearest lines of work and state the delta in each case. The through-line: prior work either augments *uniformly*, learns allocation *end-to-end*, or selects data by a *generalization-aware* criterion; we instead take an explicit, interpretable per-chunk measurement and ask whether it is a valid *allocation rule* — separating predictive validity from allocation utility.

Latent-augmented pretraining. The bootstrap lineage begins with STaR (Zelikman et al., 2022), which self-trains on rationales filtered by answer correctness, and Quiet-STaR (Zelikman et al., 2024), which drops that correctness filter — rewarding instead any thought that improves next-token prediction — and learns token-level internal thoughts on ordinary web text — spending its thought budget at *every* token, the uniform extreme of the allocation axis we study. BoLT (Ruan et al., 2025) carries the bootstrap into pretraining with an EM loop and likewise augments *every* chunk; we measure *which* chunks a latent helps and show that concentrating the budget there does not improve — and on math harms — the trained model *under the recipe that damages it*; where training improves on its start, uniform loses its lead to random routing (significantly only without density matching) (Section 4), so we read BoLT's uniform default as recipe-dependent rather than as either robust or a missed optimization. Thinking-augmented pretraining (TPT) (Wang et al., 2025a) adds generated reasoning trajectories at scale; our controls separate the effect of *adding* latents (budget) from the effect of *where* they go (signal); both move the model, and which matters more depends on the recipe. Latent-Thought LMs (Kong et al., 2025) build the prior/posterior structure into the architecture and training objective, and Ouro (Zhu et al., 2025) builds iterative latent computation with an entropy-regularised objective for learned depth allocation into pretraining; we instead read the KL between two *model-defined* conditionals as a post-hoc measurement and diagnostic, and deliberately do not claim KL itself as novel.

Learned adaptive compute. Adaptive latent CoT (Zeng et al., 2026) learns token-level halting end-to-end and reports efficiency gains; Thoughtbubbles and mixture-of-depths style routers (Raposo et al., 2024) likewise *learn* where to spend compute. These are not in tension with our result: a *learned* policy can succeed precisely where a *fixed* per-chunk proxy fails, and our finding delimits the boundary — an interpretable measurement that predicts per-chunk benefit is still not a safe static allocation rule. Variational reasoning (Zhou et al., 2026a) unifies reasoning-as-latent-variable training; we work in the reward-free pretraining-bootstrap regime and focus on the aggregated-posterior fixed point and its CMI reading rather than on a training objective.

Proxy runs and reduced learning rates. Wang et al. (2026) ask whether small training runs can guide data curation for pretraining, find that conclusions about data quality flip under small hyperparameter changes because the best configuration is data-dependent, and show across 23 data recipes that proxies trained at a reduced learning rate rank recipes consistently with fully tuned large runs, with a random-feature argument for why. We agree with the diagnosis and our learning-rate results are an instance of it, which is why we do not present learning-rate sensitivity as our finding. What continued pretraining adds is a start point that is already good. A reduced rate makes conditions differ less; it does not say whether any of them improves on the checkpoint, and in our case the reduced-rate ranking (uniform first, Table 1a) was still a damage ranking and did not identify the condition that led once training improved on its start (Table 1b,c). That is a three-condition case on one model, not a test of their procedure; a direct comparison under a regret criterion across many recipes is the natural next experiment.

Data selection and the RHO-Loss analogy. The closest conceptual neighbor is RHO-Loss (Mindermann et al., 2022): prioritizing high training-loss points hurts because such points carry high *irreducible* loss; their reducible-holdout-loss reframing shows per-example loss is the wrong benefit signal. We are not a re-run of RHO-Loss in a new domain. First, our per-chunk signal is not a raw loss but an information-theoretic divergence that *empirically predicts measured per-chunk benefit* (ρ=0.48), so the failure is not that the signal is uninformative — it is that a *valid* per-example predictor is still the wrong allocation rule. Second, we include an *oracle* on the measured per-chunk benefit itself (the best attainable such selector) and show it too loses to uniform, which, under the damaging recipe, points to the per-chunk-benefit *objective* rather than to estimation error. Notably this oracle already scores each chunk's benefit under a *fixed reference model that never trained on the routing selection* — the same reference-model structure as RHO-Loss's holdout term — so its failure to beat uniform is evidence *against*, not for, the hypothesis that a holdout-reducible variant would succeed. We test this directly with a dedicated `rho_holdout_50` baseline: two cross-fit holdout models (each trained on one half of the documents) score the out-of-sample half, giving every chunk a generalization-aware latent benefit $\Delta\text{loss}_{\text{holdout}}$; routing by it top-$50\%$ produces a model that *loses to uniform by* 0.428 *NLL/token* (three seeds, two-component 95% CI $[0.361, 0.494]$) and is not distinguishable from the oracle (paired difference -0.002, two-component CI $[-0.086, +0.082]$ on the two seeds the oracle carries — which does not establish that they are equal). The generalization-aware selector RHO-Loss would prescribe therefore behaves exactly like the training-corpus oracle and, like it, does not beat uniform — settling this comparison empirically, under the damaging recipe, rather than by analogy. Third, the harm is *domain-specific* in a way a pure hard-example story does not predict: it is large and seed-stable on math and not identified on general web text (Appendix E). Fragile influence estimates (Basu et al., 2021), memorization pitfalls (Bayat et al., 2025), distribution-matching selection (DSIR (Xie et al., 2023b)) and proxy-model domain reweighting (DoReMi (Xie et al., 2023a)) are complementary evidence that per-example utility and population generalization diverge; we extend the lesson to EM-bootstrap latent allocation, where the per-example signal is information-theoretically grounded and predictive yet still misaligned.

Bayesian surprise. Reasoning demand is a latent-space instance of Bayesian surprise (Itti & Baldi, 2006). The novelty is not the quantity but its use: we give it a fixed-point characterization (expected demand = CMI at the M-step optimum), a null-calibrated diagnostic, and a rigorous test as an allocation signal — the last of which it fails.