CANARY: SEPARATING BEHAVIORAL ANCHORING FROM RISK-GUIDED SELECTION IN CONTINUAL LLM POST-TRAINING

Anonymous authors
Paper under double-blind review

ABSTRACT

Continual post-training often couples a retention objective with a policy for selecting the behaviors it protects, making their contributions difficult to separate. CANARY distinguishes cached stage-start response anchoring from risk-guided anchor allocation. Across six matched seed blocks with Qwen2.5-3B on a fixed tool-use, chemistry, and medical stream, gradient-ranked anchoring improves final stream performance, forgetting, and held-out accuracy over the implemented live-teacher on-policy self-distillation baseline in every block, with mean paired gains of 0.0490, 0.0291, and 0.3091, respectively. Yet scored random and bottom-ranked anchors achieve stronger aggregate results. Across six saved states from one seed, gradient-conflict scores weakly agree with cloned one-microbatch AdamW loss changes, with mean Spearman correlation 0.1009. Ranking by the cloned update yields favorable mean effects relative to both matched selector controls, with joint non-worsening in three of six runs against each. A four-seed risk-screened coverage follow-up improves mean held-out accuracy over random selection but worsens stream performance and forgetting. A separate supervised-replay comparison finds no benefit from adding the tested anchoring branch. These results distinguish the benefit of an anchored training policy, the fidelity of its local risk signal, and the downstream value of using that signal to allocate a fixed budget.

1 INTRODUCTION

Language models can be post-trained sequentially for tool use, scientific reasoning, and specialized domains. Each update must acquire new behavior without unnecessarily damaging what the model already does well. This stability–plasticity problem motivates parameter regularization, output distillation, and rehearsal (Kirkpatrick et al., 2017; Li & Hoiem, 2018; Luo et al., 2025). Supervised replay retains original prompt–target pairs; other replay methods retain historical predictions or synthesize examples (Buzzega et al., 2020; Huang et al., 2024). Their effectiveness depends both on the retention objective and on which examples receive its limited budget.

CANARY examines a simple version of this distinction. At the start of each stage, a reference model generates fixed responses to a reservoir of prior prompts. Training then anchors the model to a selected subset of those responses. The same objective can protect high-risk, low-risk, random, or lexically representative prompts. Our central question is: *when cached-response anchoring improves continual post-training, how much of the observed gain requires risk-guided selection rather than the anchored training intervention itself?*

Maximally Interfered Retrieval (MIR) selects memories predicted to incur the greatest loss increase after an update and reports benefits in its evaluated settings (Aljundi et al., 2019a). CANARY examines this principle in continual LLM post-training through controlled attribution. We separately test three questions: whether an anchored policy improves the base learner; whether a gradient-conflict score agrees with a cloned optimizer proposal; and whether allocating anchors using these signals improves complete training outcomes relative to matched selector controls.

The results make this separation consequential. Gradient-ranked anchoring improves all three endpoints over the base learner in all six paired runs, but random and bottom-ranked anchoring have stronger means. Cloned-update ranking has favorable mean effects with mixed joint outcomes.

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.

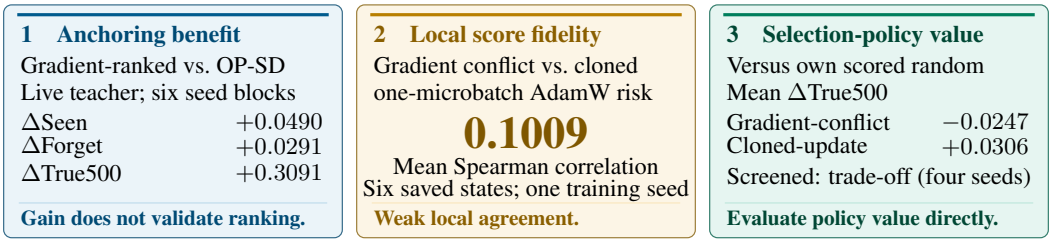


Figure 1: Three distinct tests in CANARY: anchoring benefit, local score fidelity, and full-stream selection value. Gains over the base learner do not establish selection gains, nor does score agreement establish policy value. Positive ΔForget denotes less forgetting. Sections 6 and 7 report all endpoints, variability, and paired outcomes.

Adding a risk screen to prompt coverage instead introduces a trade-off between held-out accuracy and the two stream-based endpoints. The supervised-replay comparison further shows that adding the tested anchoring branch is not beneficial in every training configuration.

We contribute (i) matched full-stream comparisons that separate anchored training from allocation policy; (ii) a saved-state diagnostic distinguishing gradient-conflict ranking from cloned one-microbatch AdamW loss change; and (iii) a fixed-budget risk-screened coverage test with complete seed-level and benchmark-slice reporting. These experiments identify *which comparisons support the measured gains*, while distinguishing each selector's conclusions.

2 Related Work

Continual learning and retention. EWC constrains changes to important parameters (Kirkpatrick et al., 2017); learning without forgetting distills a previous model's outputs (Li & Hoiem, 2018); and episodic-memory methods constrain or modify gradients using retained examples (Lopez-Paz & Ranzato, 2017; Chaudhry et al., 2019). LLM rehearsal includes continual instruction tuning with replay (Scialom et al., 2022), instruction-driven replay (Wang et al., 2024), self-synthesized rehearsal (Huang et al., 2024), and selective attention distillation (He et al., 2024). KPIG masks key instruction parts to guide replay (He et al., 2025). Dark experience replay preserves historical logits (Buzzega et al., 2020); sequence-level distillation uses generated response sequences (Kim & Rush, 2016). CANARY caches stage-start responses, with an empty-output fallback (Section 3). Whether prior selection benefits extend to this setup remains empirical.

Interference-aware retrieval and set selection. MIR retrieves memories whose loss is predicted to increase after a prospective update, in both experience- and generative-replay settings (Aljundi et al., 2019a). Gradient-based sample selection uses gradient geometry to construct an episodic memory (Aljundi et al., 2019b). DataInf estimates data influence (Kwon et al., 2024); LESS uses optimizer-aware gradient similarity for targeted instruction-data selection (Xia et al., 2024). Online coreset selection considers joint representativeness and informativeness (Yoon et al., 2022). Our risk-screened coverage policy combines a local loss-change screen with provenance-stratified prompt-token coverage. This lexical coverage is neither behavioral diversity nor human importance. We compare implementations within our protocol rather than claim superiority over MIR, GSS, or broader retention systems.

On-policy and self-distillation. DAgger motivates supervision on learner-induced states (Ross et al., 2011); on-policy language-model distillation supervises model-generated trajectories (Agarwal et al., 2024; Zhao et al., 2026). For continual learning, SDFT uses an EMA teacher by default (Shenfeld et al., 2026). OPSD defines teacher and student policies through different contexts of the same model, with stop-gradient teacher targets, pointwise divergence clipping, and longer main-run continuations (Zhao et al., 2026). Here, *OP-SD (live teacher)* denotes our local forward-KL implementation with 8–64-token continuations, no persistent teacher, and no pointwise divergence clipping. It is held fixed across selector arms, but is not a reproduction of either complete upstream training configuration. Current-stage rollouts and stage-start cached responses have different roles:

the former belong to the base learner, while the latter are fixed retention targets. Caching avoids new annotations, but still requires inference and response storage. Appendix B gives broader context.

3 SETUP: CONTINUAL BEHAVIORAL ANCHORING

Let tasks arrive at stages $t = 1, \ldots, T$, with stage-start parameters θ_t^0. Write $\mathcal{L}_{\text{task}}^t(\theta)$ for the implemented OP-SD (live-teacher) current-stage objective, common to all selector arms. A reservoir $\mathcal{R}_{<t}$ stores prompts from earlier stages. For each $x_i \in \mathcal{R}_{<t}$, generate the deterministic response

$$\tilde{y}_i = f_{\theta_t^0}(x_i). \tag{1}$$

The cached target $\hat{y}_i$ is $\tilde{y}_i$ when nonempty; otherwise the implementation uses the reservoir row's original target. The stage index on $\hat{y}_i$ is implicit: targets are regenerated at the start of a stage and remain fixed for its 24 optimizer steps. At each refresh, a policy selects $S_t \subset \mathcal{R}_{<t}$ with $|S_t| = B$, and training uses

$$\mathcal{L}^t(\theta) = \mathcal{L}_{\text{task}}^t(\theta) + \beta \frac{\sum_{i \in S_t} \sum_{k \in \mathcal{T}_i} -\log p_\theta(\hat{y}_{i,k} \mid x_i, \hat{y}_{i,<k})}{\sum_{i \in S_t} |\mathcal{T}_i|}, \tag{2}$$

where $\mathcal{T}_i$ contains the unmasked cached-response positions that remain after left truncation to 1,024 tokens. Prompt and padding positions are masked. Thus the implementation averages over response tokens in the anchor batch, rather than first averaging each sequence. There is no anchoring term in stage 1, when the prior reservoir is empty. The anchoring term increases likelihood of selected cached responses; it does not certify that those responses are correct or desirable. Within a stage, selection changes the protected subset, not the cached target for a given prompt.

OP-SD samples a continuation from the current model under the original task prompt. The same model, evaluated without gradients under a privileged prompt containing the dataset reference answer, supplies teacher distributions on that continuation. The task loss is mean teacher-to-student token KL. Both distributions are recomputed at every microbatch, without a persistent teacher or pointwise divergence clipping. The comparisons assess anchoring and allocation under this implementation, not the complete upstream training configurations. Appendix D describes context construction, decoding, loss reductions, and gradient accumulation.

3.1 EVALUATION ENDPOINTS

Let $a_{u,j}$ denote the score on the fixed evaluation split of task j after stage u, and let $T = 3$. Final seen-task performance and Average Forgetting are

$$\text{SeenAvg}_T = \frac{1}{T} \sum_{j=1}^{T} a_{T,j}, \tag{3}$$

$$\text{AF}_T = \frac{1}{T-1} \sum_{j=1}^{T-1} \left(\max_{u \in \{j, \ldots, T-1\}} a_{u,j} - a_{T,j} \right). \tag{4}$$

Higher SEENAVG and lower Average Forgetting are better. SEENAVG includes the final task and therefore combines acquisition with retention. Forgetting compares earlier tasks with their best pre-final-stage scores; negative contributions indicate backward transfer and are not clipped to zero. Because each arm supplies its own historical maximum, lower forgetting can reflect either stronger preservation or a lower earlier peak; stagewise task scores are therefore reported separately in Appendix G.1. For ToolAlpaca, ChemData700K, and medical-o1-reasoning-SFT alike, $a_{u,j}$ is mean token F1 over the fixed 32-row task split. Evaluation uses greedy 96-token decoding; lowercase whitespace-delimited tokens are compared after ASCII-punctuation and article removal. No task-specific answer parser is applied; an empty prediction against a nonempty target, or a prediction with no token overlap, scores zero. These stream metrics measure reference-response token overlap rather than verified task success; held-out answer accuracy is a separate evaluation.

The held-out metric is a five-slice macro-accuracy with 500 examples per slice (2,500 predictions per checkpoint). The set $\mathcal{D}$ comprises: SciKnowEval Chemistry (Feng et al., 2024), GSM8K (Cobbe

et al., 2021), MMLU (Hendrycks et al., 2021), MedMCQA (Pal et al., 2022), and PubMedQA (Jin et al., 2019). For extracted predictions $\hat{z}_{d,i}$ and reference answers $z_{d,i}$,

$$\text{True500}_T = \frac{1}{5} \sum_{d \in \mathcal{D}} \frac{1}{500} \sum_{i=1}^{500} \mathbf{1}[\hat{z}_{d,i} = z_{d,i}] \,. \tag{5}$$

Thus TRUE500 equally weights five held-out accuracies; it is not the seen-task score or a direct forgetting measure. We report final stream performance, forgetting, and held-out accuracy as distinct endpoints. Multiple-choice evaluation uses the last standalone A–E token, GSM8K uses the last matched number, and PubMedQA uses normalized exact match; failed extraction scores zero. Appendix D specifies the fixed slices and decoding limits.

3.2 CONTROLLED STREAM AND BUDGETS

Within each seed block, branches resume from the same immutable ToolAlpaca stage-1 checkpoint (Tang et al., 2023), then train on ChemData700K (Zhang et al., 2024) and English medical-o1-reasoning-SFT (FreedomIntelligence, 2024; Chen et al., 2025). We use Qwen2.5-3B-Instruct (Qwen Team, 2024) with LoRA (Hu et al., 2022): rank 16, alpha 32, dropout 0.05, learning rate 10^{-4}, batch size 1, gradient accumulation 4, and 24 optimizer steps per stage. Each stage has 128 training, 32 evaluation, and 64 new reservoir examples. Prior reservoirs persist: chemistry uses 64 tool prompts, and medicine uses 128 tool/chemistry prompts. At each refresh, the candidate count is $M = 32$, the anchor budget is $B = 8$, and $\beta = 0.3$; selection refreshes every four steps. Row splits are index-disjoint, although one chemistry prompt has semantic overlap between training and reservoir slices. Appendix C records the configuration.

4 RISK-GUIDED SELECTION UNDER A FIXED ANCHOR BUDGET

4.1 GRADIENT CONFLICT AND CLONED-UPDATE RISK

At a refresh, let U be the current task microbatch and write $\ell_i(\theta) = \mathcal{L}_{\text{NLL}}(f_\theta(x_i), \hat{y}_i)$. The gradient-conflict score is $c_i = -\langle \nabla \ell_i, \nabla \mathcal{L}_{\text{task}}(\theta; U) \rangle$. For a small parameter displacement d, a first-order expansion gives

$$\ell_i(\theta + d) - \ell_i(\theta) \approx \nabla \ell_i(\theta)^\top d. \tag{6}$$

For plain SGD, $d = -\eta \nabla \mathcal{L}_{\text{task}}$, recovering ηc_i. An adaptive optimizer such as AdamW (Loshchilov & Hutter, 2019) also uses moment history and coordinate-wise scaling, motivating update-aware rather than raw-gradient comparisons (Xia et al., 2024). Consequently, agreement between gradient conflict and an AdamW loss change is an empirical question, not a consequence of the first-order expansion.

Let $\theta^+ = \text{AdamWClone}(\theta; \nabla \mathcal{L}_{\text{task}}(\theta; U))$ be a proposal from a cloned optimizer state. Define

$$r_i = \ell_i(\theta^+) - \ell_i(\theta). \tag{7}$$

Positive r_i indicates increased loss on the cached response under that proposal. The scorer evaluates candidate losses in FP32 using the captured AdamW state and BF16 microstep-0 task gradient. It models one microbatch, not the full four-microbatch accumulated update. Ranking by r_i is *cloned-update risk* (artifact: `exact-risk-only`); "cloned" distinguishes this proposal from the subsequent training step.

4.2 PROMPT-ONLY FACILITY-LOCATION COVERAGE

Let z_i be the set of tokenizer IDs in prompt x_i, and g_i its provenance stage. We use the fixed prompt-only similarity

$$s(i,j) = \begin{cases} |z_i \cap z_j| / |z_i \cup z_j|, & g_i = g_j, \\ 0, & g_i \neq g_j. \end{cases} \tag{8}$$

For the local candidate pool $\mathcal{C}$, use facility-location coverage (Wei et al., 2015):

$$F(S) = \frac{1}{|\mathcal{C}|} \sum_{j \in \mathcal{C}} \max_{i \in S} s(i,j), \qquad F(\varnothing) = 0. \tag{9}$$

The coverage objective uses prompt token IDs and provenance, never cached response text, task or held-out labels, or post-update evaluation outcomes. It measures lexical representation within the candidate pool. Similarity uses the untruncated rendered prompt, while training truncates to 1024 tokens; Section 9 discusses this mismatch.

4.3 RISK-SCREENED COVERAGE

Pointwise ranking does not explicitly reward coverage of the candidate set. We test a constrained allocation policy rather than assume that improved coverage improves downstream outcomes. For each nonempty provenance stratum $\mathcal{C}_g$, let $n_g = |\mathcal{C}_g|$. Hamilton apportionment assigns q_g proportionally to n_g, with at least one anchor per represented stratum when feasible and $\sum_g q_g = B$. The eligible set is

$$\mathcal{H}_g = \mathrm{TopRisk}_{\min(2q_g, n_g)}(\mathcal{C}_g). \tag{10}$$

Within stratum g, the policy greedily selects q_g elements of $\mathcal{H}_g$ by marginal gain $F(S \cup \{i\}) - F(S)$, with deterministic ties resolved by risk rank and prompt identity. Coverage is evaluated over *all* candidates in Eq. 9, not only the eligible set. The screen restricts which anchors can be chosen; it does not change the anchoring loss or its budget.

The evaluated screen uses multiplier 2. On 48 saved refresh traces, an equal-weight risk/coverage alternative shared 7.58 of eight anchors with global cloned-update top-8. The constrained policy shared 3.85, versus 1.75 for coverage-only, providing a more distinct allocation intervention. Appendix E distinguishes these same-state allocation diagnostics from policy-dependent training traces.

Scoring-matched controls. The four coverage-study arms are **risk-screened coverage**, defined above; **cloned-update risk**, the eight largest r_i; **coverage-only**, the same quotas and greedy coverage without the screen; and **random**, a seeded sample of eight from the scored pool. All four run the same cloned scorer on 32 candidates, including policies that ignore its scores. Candidate counts, anchor budgets, optimizer steps, scoring operations, and evaluation budgets are matched. These controls test allocation at a matched scoring workload, not the minimum deployment cost of each policy: an operational random selector could omit scoring. The pointwise-risk studies also include bottom-ranked controls.

5 EXPERIMENTAL DESIGN

Matched blocks and policy contrasts. The gradient-conflict and cloned-update comparisons use six seeds, 410–415; the coverage follow-up uses the first four blocks, 410–413. Within a block, branches restore the same base model, LoRA adapter, AdamW moments, scheduler position, training history, and Python/Torch/CUDA random states. They share task order, row partitions, target-generation rule, refresh schedule, and evaluation protocol. Table 11 gives the full matching specification. The four-seed follow-up overlaps the six-seed study; it is not an independent replication of the cloned-update results.

For one seed, let $Y(\pi)$ collect the three endpoints, oriented so that higher is better, under training policy π. The distinction between the two full-stream comparisons can be written as

$$\begin{aligned} Y(\pi_{\mathrm{risk}}) - Y(\pi_{\mathrm{base}}) = {} & [Y(\pi_{\mathrm{random}}) - Y(\pi_{\mathrm{base}})] \\ & + [Y(\pi_{\mathrm{risk}}) - Y(\pi_{\mathrm{random}})]. \end{aligned} \tag{11}$$

This accounting identity does not model training dynamics. The first contrast measures random-anchored training; the second measures a change in selection policy. Thus a positive risk-versus-base effect can coexist with a negative risk-versus-random effect. The base comparison includes anchoring computation.

Trajectories after branching. The endpoint comparison evaluates complete policies initialized from the same state. Subsequent parameter states, stage-start cached responses, and sampled pools may differ as the policies evolve. Within a fixed state and pool, only the selection rule changes; across runs, the comparison includes its downstream trajectory effects. Same-state diagnostics and full-stream outcomes thus answer different questions. Chemistry pool and score-map checks are reported in Appendix E. Risk-screened coverage always preceded coverage-only within a block; arm order was not counterbalanced, so recorded wall times do not support efficiency comparisons.

Table 1: Six-seed anchored-policy comparison (mean ± sample SD). Higher SEENAVG and TRUE500 are better; lower Average Forgetting (Eq. 4) is better.

Method	SEENAVG ↑	Forgetting ↓	TRUE500 ↑
OP-SD (live)	0.0618 ± 0.0367	0.0640 ± 0.0200	0.0347 ± 0.0580
Gradient-conflict top-8	0.1108 ± 0.0437	0.0350 ± 0.0198	0.3437 ± 0.0843
Grad.-scored random-8	0.1211 ± 0.0482	0.0252 ± 0.0309	0.3685 ± 0.0642
Grad.-scored bottom-8	0.1157 ± 0.0356	0.0337 ± 0.0197	0.4080 ± 0.0203

Table 2: Paired full-stream deltas for gradient-conflict top-8 minus comparator. Entries are mean ± sample SD; forgetting is an improvement, so positive favors top-8. Parentheses give positive seeds out of six.

Comparator	ΔSeen	ΔForget	ΔTrue500	Joint wins
OP-SD	$+0.0490 \pm 0.0233$ (6)	$+0.0291 \pm 0.0191$ (6)	$+0.3091 \pm 0.0769$ (6)	6/6
Grad.-random-8	-0.0103 ± 0.0128 (1)	-0.0098 ± 0.0153 (1)	-0.0247 ± 0.0254 (1)	0/6
Grad.-bottom-8	-0.0049 ± 0.0179 (3)	-0.0013 ± 0.0060 (2)	-0.0643 ± 0.0676 (0)	0/6

Endpoints and paired reporting. Define ΔSeen and $\Delta\text{True500}$ as treatment minus comparator, and $\Delta\text{Forget} = \text{AF}_T^{\text{comparator}} - \text{AF}_T^{\text{treatment}}$; positive values favor treatment on all three endpoints. A *joint-frontier win* is an observed Pareto improvement: all three paired deltas are nonnegative and at least one is positive. Counts summarize multi-endpoint consistency, not statistical significance; zero wins does not imply that every endpoint worsens in every run. Within each contrast, means and sample standard deviations use complete paired runs as independent units; refreshes, candidates, and benchmark rows are not additional runs.

We summarize the coverage comparisons against practical mean-effect targets of $\Delta\text{SEENAVG} \geq 0.01$, $\Delta\text{TRUE500} \geq 0.02$, non-worse mean forgetting, and at least three of four seed-level joint wins against both random and cloned-update risk. These practical-effect criteria and their evaluation appear in Appendix D. All four arms complete every block; no missing endpoint is imputed.

6 SEPARATING ANCHORING BENEFIT FROM POINTWISE SELECTION

6.1 ANCHORING IMPROVES THE BASE LEARNER WITHOUT REQUIRING CONFLICT RANKING

The first comparison branches into OP-SD, gradient-conflict top-8, gradient-scored random-8, and gradient-scored bottom-8. The three anchor arms execute the same candidate-gradient calls. Task batches and task losses match through chemistry step 4, before the first intervention, and first-refresh scores match exactly.

All three anchored variants have higher mean SEENAVG and TRUE500 and lower mean forgetting than OP-SD. Gradient-conflict top-8 improves all three endpoints in every paired run, with mean improvements $+0.0490$, $+0.0291$, and $+0.3091$, respectively (Table 2). Yet its mean paired changes against its gradient-scored random-8 control are negative on every endpoint. Bottom-ranked anchors also have stronger aggregate outcomes. Thus the anchored-policy advantage over OP-SD does not require gradient-conflict ordering.

Interpreting the baseline contrast. The held-out difference is large: OP-SD averages 0.0347 on TRUE500, versus 0.3437 for gradient-ranked and 0.3685 for random anchoring. These results apply to the implemented base learner and protocol. Final endpoint scores alone do not distinguish preservation of answer competence from prevention of output-format deterioration. The separate-model stress tests in Appendix L do not diagnose the Qwen2.5-3B baseline.

For this comparison, lower mean forgetting accompanies higher acquisition and final prior-task means, rather than a lower mean acquisition score. Appendix G.1 gives per-seed values. Stage-2 held-out generations and validity records were not retained, so we do not diagnose baseline collapse.

Table 3: Stagewise in-stream token F1 (mean ± sample SD; six blocks). S1 is shared; C2 and M3 follow chemistry and medicine.

Checkpoint	Arm	Tool	Chemistry	Medical
S1	Shared	$.1458 \pm .0331$	—	—
C2	OP-SD	$.0420 \pm .0233$	$.0366 \pm .0290$	—
C2	Grad.-random	$.0871 \pm .0199$	$.0618 \pm .0370$	—
M3	OP-SD	$.0290 \pm .0146$	$.0253 \pm .0165$	$.1312 \pm .0915$
M3	Grad.-random	$.0898 \pm .0430$	$.0674 \pm .0281$	$.2060 \pm .0936$

Table 4: Cloned-update risk minus its comparator over six seeds. Entries are mean ± sample SD; forgetting is an improvement, so positive favors cloned-update risk.

Comparator	ΔSeen	ΔForget	ΔTrue500	Joint wins
OP-SD	$+0.0712 \pm 0.0219$	$+0.0461 \pm 0.0190$	$+0.3644 \pm 0.0545$	6/6
Cloned-random-8	$+0.0120 \pm 0.0207$	$+0.0072 \pm 0.0231$	$+0.0306 \pm 0.0529$	3/6
Low-risk-8	$+0.0170 \pm 0.0219$	$+0.0142 \pm 0.0130$	$+0.0474 \pm 0.0817$	3/6

6.2 Local gradient conflict weakly matches the cloned AdamW proposal

We compare scores at six saved pre-refresh states: chemistry and medical steps 4, 12, and 20, with 32 candidates per state. All six states come from one training seed. Each comparison reuses the cached target and saved BF16 task gradient, evaluating the loss change under the FP32 cloned proposal. Mean/median/minimum Spearman agreement is $0.1009/0.0894/-0.3691$, and the top-8 sets share 14 selections out of 48 in total. For scale, two independent uniformly random size-eight subsets of a size-32 pool have expected overlap 2 per state, or 12 over six states; the observed overlap fraction is $14/48 = 29.17\%$, versus a 25% uniform reference. This is not a significance test and does not make either ranking random. Recomputing the gradient proxy in FP32 gives mean Spearman 0.0970 and the same overlap; cached-target identity is 100%.

In these states, the implemented gradient-conflict ranking has low agreement with the one-microbatch AdamW loss-change ranking. Higher precision alone does not remove that mismatch. Equation 6 explains why the two quantities need not coincide, but this diagnostic does not isolate the cause of disagreement or establish either score's relationship to final held-out accuracy.

6.3 Cloned-update ranking has favorable means and mixed joint outcomes

The cloned-update arm ranks candidates by Eq. 7 in the same six seed blocks. Its controls execute the same scoring operations but select a seeded random eight (cloned-scored random-8) or the eight lowest-scoring candidates (low-risk-8). Each refresh requires 64 candidate-loss evaluations: before and after the proposal for each of 32 candidates.

Mean paired changes favor cloned-update ranking over both controls (Table 4). Against cloned-scored random-8, they are $+0.0120$ in SeenAvg, $+0.0072$ in forgetting improvement, and $+0.0306$ in True500, with three of six joint wins. The low-risk comparison also has favorable means and three joint wins. This differs from the adverse mean effects of gradient-conflict ranking: the observed cloned-update advantage is promising within the tested stream, while its consistency and portability remain uncertain with six runs. It does not retroactively validate the gradient-conflict proxy. Appendix F gives paired uncertainty intervals, and Appendix G gives every seed-level endpoint.

7 Risk-Screened Coverage Introduces Endpoint Trade-offs

The coverage follow-up comprises sixteen arm–seed outcomes: four policies across four matched blocks. Each final checkpoint completes all five 500-example held-out evaluations, with no run excluded.

A held-out gain is not a joint improvement. Against random selection, risk-screened coverage increases mean True500 by 0.0311 ± 0.0385, but reduces SeenAvg by 0.0049 ± 0.0160 and

Table 5: Selector results over seeds 410–413 (mean ± sample SD). Higher SeenAvg and True500 are better; lower Average Forgetting (Eq. 4) is better.

Method	SeenAvg ↑	Forgetting ↓	True500 ↑
Risk-screened coverage	0.1084 ± 0.0487	0.0344 ± 0.0106	0.3816 ± 0.0362
Cloned-update	0.1269 ± 0.0355	0.0151 ± 0.0072	0.3957 ± 0.0168
Coverage-only	0.1200 ± 0.0432	0.0239 ± 0.0161	0.3787 ± 0.0566
Random	0.1132 ± 0.0602	0.0218 ± 0.0296	0.3505 ± 0.0745

Table 6: Paired selector effects over four complete seeds. Entries are mean ± sample SD; forgetting is an improvement, so positive favors risk-screened coverage.

Comparator	ΔSeen	ΔForget	ΔTrue500	Joint wins
Random	-0.0049 ± 0.0160	-0.0126 ± 0.0209	$+0.0311 \pm 0.0385$	0/4
Cloned-update	-0.0185 ± 0.0188	-0.0193 ± 0.0053	-0.0141 ± 0.0228	0/4
Coverage-only	-0.0116 ± 0.0165	-0.0105 ± 0.0091	$+0.0029 \pm 0.0311$	0/4

worsens forgetting by 0.0126 ± 0.0209 (Table 6). The held-out gain exceeds the stated mean-effect target and is positive in three of four seeds. The other criteria are not met. Against cloned-update risk, all three mean paired effects are negative.

Adding the screen to coverage-only lowers mean SeenAvg and increases mean forgetting, while producing a small positive True500 difference ($+0.0029 \pm 0.0311$). This is a trade-off, not an absence of change on every endpoint. None of the three comparator contrasts yields a joint-frontier win. Cloned-update risk has the strongest descriptive mean on all three endpoints in these four blocks; this result is not an independent replication or a claim that it is generally the best selector.

Seed-level trade-offs. Figure 2 shows each paired run and the separately marked mean. Against random, seeds 411 and 413 have the largest held-out gains ($+0.0784$ and $+0.0460$). Seed 410 instead improves the two stream endpoints but loses 0.0052 on True500. Against cloned-update risk, forgetting improvement is negative in every block. These profiles explain why a positive average on one endpoint can coexist with no joint wins. Complete paired values are in Appendix H, and the descriptive five-benchmark decomposition is in Appendix J.

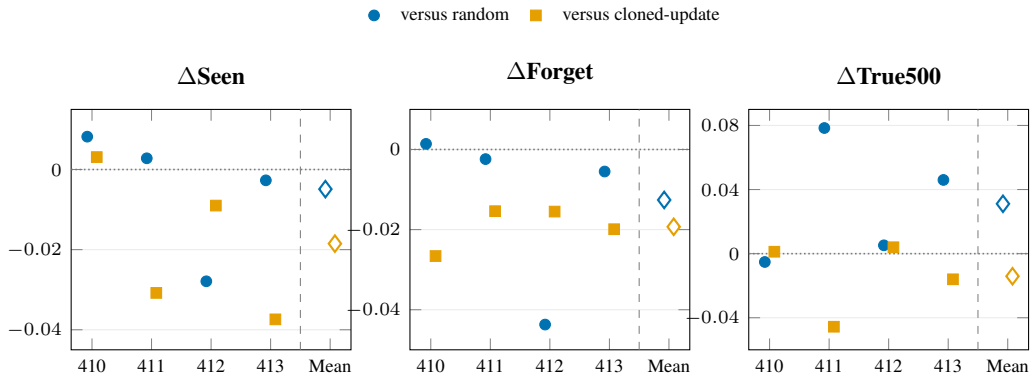


Figure 2: Risk-screened coverage minus comparator: filled markers show four seeds; open diamonds show means. Positive is favorable. Scales differ; practical-effect targets apply to means (Appendix D).

8 Replay, Computational Cost, and Interpretation

Anchoring is not uniformly additive to supervised replay. Across four shared-state seeds (304–307), adding gradient-selected anchoring to replay changes mean SeenAvg, forgetting improvement, and True500 by -0.0934, -0.0342, and -0.0969, respectively, with no joint win. Replay conditions

and evaluation slices are matched within this comparison. Thus anchoring improves OP-SD (live) but not this replay configuration. Its batching and accumulation differ from the selector study, so the results are not pooled and do not establish that other combinations would behave similarly.

What the workload matching controls. Each cloned-score selector performs 64 candidate-loss evaluations per refresh, with the same candidate and anchor counts and optimizer-step budget. This matches scoring operations, not runtime or token-level work: selected response lengths can differ, and operational random selection could omit scoring. Across the eight coverage-arm runs, recorded continued-training stages total 9.75 process-hours, excluding final evaluation and orchestration; peak per-process CUDA allocation is 34,406 MiB. Fixed arm order, co-tenancy, and asymmetric state serialization preclude interpreting these records as throughput or efficiency comparisons (Appendix K).

Predicted damage is not intervention value. The score r_i measures vulnerability to a task-only proposal, not the benefit of anchoring a set S. Even a faithful loss-change estimate does not determine protection of other examples, redundancy among anchor gradients, or the combined update's effect on acquisition. Equation 2 weights sequences by surviving response-token counts. Fixing eight anchors matches example count, not individual loss weights: selection changes both content and token-weighted composition. These policy properties are not separately identified causes of the observed outcomes.

Accordingly, a stronger local score or more representative prompt set must still demonstrate downstream value under the same objective and budget. The three tests distinguish those questions without assuming a causal link between local fidelity and final performance.

9 Limitations

The controlled evidence covers one backbone, one task order, short stages, and one live-teacher base learner. Six blocks support the pointwise comparisons; the four-block coverage follow-up overlaps them. The large gap above the weak baseline does not establish whether anchoring mainly stabilizes this particular learner or improves retention under better-stabilized training. Fixed or EMA teachers, divergence clipping, and longer streams remain untested here.

The stream scores are reference-response token F1, not verified task success; strict held-out accuracy also depends on extraction and output format. The stagewise results clarify acquisition and retention for the displayed control, but missing stage-2 held-out generations and validity records prevent a competence/format decomposition. Empty cache generations fall back to original dataset targets, and fallback frequency was not logged. The evidence therefore concerns the implemented anchoring policy, not a verified purely model-generated-target intervention.

The cloned scorer models one microbatch rather than a full accumulated update. Coverage is lexical, and 16/384 risk-screened and 24/384 coverage-only selected prompts exceed the training length limit, so selection can use tokens later removed by truncation. Subsequent diagnostics follow policy-dependent states. Cached responses may preserve errors or harmful behavior, and neither risk nor coverage measures human importance. The replay comparison is protocol-specific; there is no matched superiority test against MIR, GSS, or broader retention systems (Appendix M).

10 Conclusion

CANARY separates an anchored training intervention from the policy used to allocate its budget. In the controlled Qwen2.5-3B stream, gradient-ranked anchoring improves all three endpoints over OP-SD (live) in every paired block, while random and bottom-ranked anchors achieve stronger means. Cloned one-microbatch AdamW ranking yields favorable mean effects with mixed joint outcomes; risk-screened coverage trades held-out accuracy against stream performance and forgetting. The adverse replay comparison further limits claims of general complementarity. Together, these results distinguish anchored-policy benefit, local score fidelity, and full-stream allocation value. Evaluating these quantities separately explains what each comparison supports and provides a clearer basis for developing future retention policies.

AI USE STATEMENT

In this work, we used generative AI tools to provide feedback on research methodology and experiments, assist with research-code refactoring and experiment orchestration, and support analysis checks and interpretation of results. Additionally, we used these tools to identify and organize relevant literature, format references, and draft and edit manuscript sections, including their structure, readability, and visual presentation.

Within the experimental pipeline, language models generated synthetic training material: on-policy trajectories and stage-start responses cached as anchoring pseudo-targets (Section 3 and Appendix D). Empty cached generations fall back to existing dataset targets. This generation is part of the training method, distinct from research-assistant use; generated responses are not verified ground-truth labels.

AI-assisted manuscript checks covered citation and cross-reference consistency, arithmetic consistency of reported summaries, and typesetting. The AI-assisted text, code, analyses, and references were reviewed by an author. We take responsibility for the final content of this work, including AI-assisted text, claims, code, and artifacts.

REPRODUCIBILITY STATEMENT

Anonymous research artifact:
https://anonymous.4open.science/r/CANARY-ICLR27-0685/.

Sections 3–5 define the anchoring objective, selection policies, budgets, endpoint aggregates, and matched-block design. Appendices C and D record the data sources, configuration, matching specification, paired signs, and practical-effect comparison criteria. Appendix E summarizes selector-trace diagnostics; Appendix F specifies the statistical unit. Complete six-seed endpoints and the available stagewise task scores are in Appendix G; complete four-seed outcomes and benchmark slices are in Appendices H and J. Figures 1 and 2 distinguish the experimental questions and show paired seed effects. Resource accounting, additional diagnostics, and separate output-validity stress tests are in Appendices K and L. Known provenance and output-record limitations are identified in Section 9 and Appendices C and G.1.

ETHICS STATEMENT

Behavioral anchoring trains toward cached targets—normally stage-start responses, with original dataset targets used for empty-output fallback—rather than enforcing factual correctness or human preferences. It can stabilize incorrect, biased, unsafe, or otherwise undesirable behavior. Neither update risk nor lexical coverage identifies which behaviors people value or which forgetting is acceptable. The measured gains are not evidence of safety, fairness, or suitability for high-stakes use. Deployment would require content review and safety filtering, ways to override or remove harmful retained behaviors, privacy review of stored prompts and responses, and assessment of whose behaviors are preserved. The controlled experiments do not establish these safeguards. Section 9 and Appendix M discuss the scope of the findings and their implications.

REFERENCES

Rishabh Agarwal, Nino Vieillard, Yongchao Zhou, Piotr Stanczyk, Sabela Ramos, Matthieu Geist, and Olivier Bachem. On-policy distillation of language models: Learning from self-generated mistakes. In *International Conference on Learning Representations*, 2024. URL https://openreview.net/forum?id=3zKtaqxLhW.

Rahaf Aljundi, Eugene Belilovsky, Tinne Tuytelaars, Laurent Charlin, Massimo Caccia, Min Lin, and Lucas Page-Caccia. Online continual learning with maximal interfered retrieval. In *Advances in Neural Information Processing Systems*, volume 32, 2019a. URL https://proceedings.neurips.cc/paper/2019/hash/15825aee15eb335cc13f9b559f166ee8-Abstract.html.

Rahaf Aljundi, Min Lin, Baptiste Goujaud, and Yoshua Bengio. Gradient based sample selection for online continual learning. In *Advances in Neural Information Processing Systems*, vol-

ume 32, 2019b. URL https://proceedings.neurips.cc/paper_files/paper/2019/hash/e562cd9c0768d5464b64cf61da7fc6bb-Abstract.html.

Stella Biderman, Hailey Schoelkopf, Lintang Sutawika, Leo Gao, Jonathan Tow, Baber Abbasi, Alham Fikri Aji, Pawan Sasanka Ammanamanchi, Sidney Black, Jordan Clive, Anthony DiPofi, Julen Etxaniz, Benjamin Fattori, Jessica Zosa Forde, Charles Foster, Jeffrey Hsu, Mimansa Jaiswal, Wilson Y. Lee, Haonan Li, Charles Lovering, Niklas Muennighoff, Ellie Pavlick, Jason Phang, Aviya Skowron, Samson Tan, Xiangru Tang, Kevin A. Wang, Genta Indra Winata, François Yvon, and Andy Zou. Lessons from the trenches on reproducible evaluation of language models. arXiv preprint arXiv:2405.14782, 2024. URL https://arxiv.org/abs/2405.14782.

Tom B. Brown, Benjamin Mann, Nick Ryder, Melanie Subbiah, Jared Kaplan, Prafulla Dhariwal, Arvind Neelakantan, Pranav Shyam, Girish Sastry, Amanda Askell, Sandhini Agarwal, Ariel Herbert-Voss, Gretchen Krueger, Tom Henighan, Rewon Child, Aditya Ramesh, Daniel M. Ziegler, Jeffrey Wu, Clemens Winter, Christopher Hesse, Mark Chen, Eric Sigler, Mateusz Litwin, Scott Gray, Benjamin Chess, Jack Clark, Christopher Berner, Sam McCandlish, Alec Radford, Ilya Sutskever, and Dario Amodei. Language models are few-shot learners. In *Advances in Neural Information Processing Systems*, volume 33, pp. 1877–1901, 2020. URL https://arxiv.org/abs/2005.14165.

Pietro Buzzega, Matteo Boschini, Angelo Porrello, Davide Abati, and Simone Calderara. Dark experience for general continual learning: A strong, simple baseline. In *Advances in Neural Information Processing Systems*, volume 33, pp. 15920–15930, 2020. URL https://proceedings.neurips.cc/paper/2020/hash/b704ea2c39778f07c617f6b7ce480e9e-Abstract.html.

Arslan Chaudhry, Marc'Aurelio Ranzato, Marcus Rohrbach, and Mohamed Elhoseiny. Efficient lifelong learning with A-GEM. In *International Conference on Learning Representations*, 2019. URL https://openreview.net/forum?id=Hkf2_sC5FX.

Hongyang Chen, Zhongwu Sun, Hongfei Ye, Kunchi Li, and Xuemin Lin. Beyond static models: An evolving framework for continual learning in large language models across training stages. arXiv preprint arXiv:2603.12658, 2026. URL https://arxiv.org/abs/2603.12658.

Junying Chen, Zhenyang Cai, Ke Ji, Xidong Wang, Wanlong Liu, Rongsheng Wang, and Benyou Wang. Towards medical complex reasoning with LLMs through medical verifiable problems. In Wanxiang Che, Joyce Nabende, Ekaterina Shutova, and Mohammad Taher Pilehvar (eds.), *Findings of the Association for Computational Linguistics: ACL 2025*, pp. 14552–14573, Vienna, Austria, July 2025. Association for Computational Linguistics. doi: 10.18653/v1/2025.findings-acl.751. URL https://aclanthology.org/2025.findings-acl.751/.

Mark Chen, Jerry Tworek, Heewoo Jun, Qiming Yuan, Henrique Ponde de Oliveira Pinto, Jared Kaplan, Harri Edwards, Yuri Burda, Nicholas Joseph, Greg Brockman, Alex Ray, Raul Puri, Gretchen Krueger, Michael Petrov, Heidy Khlaaf, Girish Sastry, Pamela Mishkin, Brooke Chan, Scott Gray, Nick Ryder, Mikhail Pavlov, Alethea Power, Lukasz Kaiser, Mohammad Bavarian, Clemens Winter, Philippe Tillet, Felipe Petroski Such, Dave Cummings, Matthias Plappert, Fotios Chantzis, Elizabeth Barnes, Ariel Herbert-Voss, William Hebgen Guss, Alex Nichol, Alex Paino, Nikolas Tezak, Jie Tang, Igor Babuschkin, Suchir Balaji, Shantanu Jain, William Saunders, Christopher Hesse, Andrew N. Carr, Jan Leike, Josh Achiam, Vedant Misra, Evan Morikawa, Alec Radford, Matthew Knight, Miles Brundage, Mira Murati, Katie Mayer, Peter Welinder, Bob McGrew, Dario Amodei, Sam McCandlish, Ilya Sutskever, and Wojciech Zaremba. Evaluating large language models trained on code. arXiv preprint arXiv:2107.03374, 2021. URL https://arxiv.org/abs/2107.03374.

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

Kehua Feng, Keyan Ding, Weijie Wang, Xiang Zhuang, Zeyuan Wang, Ming Qin, Yu Zhao, Jianhua Yao, Qiang Zhang, and Huajun Chen. SciKnowEval: Evaluating multi-level scientific knowledge

of large language models. arXiv preprint arXiv:2406.09098, 2024. URL https://arxiv.org/abs/2406.09098v1. Version 1.

FreedomIntelligence. medical-o1-reasoning-SFT: Dataset card. Hugging Face dataset card, 2024. URL https://huggingface.co/datasets/FreedomIntelligence/medical-o1-reasoning-SFT. Accessed September 25, 2026.

Jinghan He, Haiyun Guo, Kuan Zhu, Zihan Zhao, Ming Tang, and Jinqiao Wang. SEEKR: Selective attention-guided knowledge retention for continual learning of large language models. In Yaser Al-Onaizan, Mohit Bansal, and Yun-Nung Chen (eds.), *Proceedings of the 2024 Conference on Empirical Methods in Natural Language Processing*, pp. 3254–3266, Miami, Florida, USA, November 2024. Association for Computational Linguistics. doi: 10.18653/v1/2024.emnlp-main.190. URL https://aclanthology.org/2024.emnlp-main.190/.

Yongquan He, Wenyuan Zhang, Xuancheng Huang, Peng Zhang, Lingxun Meng, Xiang Zhou, Ke Zeng, and Xunliang Cai. Don't half-listen: Capturing key-part information in continual instruction tuning. In Wanxiang Che, Joyce Nabende, Ekaterina Shutova, and Mohammad Taher Pilehvar (eds.), *Proceedings of the 63rd Annual Meeting of the Association for Computational Linguistics (Volume 1: Long Papers)*, pp. 23649–23668, Vienna, Austria, July 2025. Association for Computational Linguistics. doi: 10.18653/v1/2025.acl-long.1153. URL https://aclanthology.org/2025.acl-long.1153/.

Dan Hendrycks, Collin Burns, Steven Basart, Andy Zou, Mantas Mazeika, Dawn Song, and Jacob Steinhardt. Measuring massive multitask language understanding. In *International Conference on Learning Representations*, 2021. URL https://openreview.net/forum?id=d7KBjmI3GmQ.

William Hoy and Nurcin Celik. STABLE: Gated continual learning for large language models. arXiv preprint arXiv:2510.16089, 2025. URL https://arxiv.org/abs/2510.16089.

Edward J. Hu, Yelong Shen, Phillip Wallis, Zeyuan Allen-Zhu, Yuanzhi Li, Shean Wang, Lu Wang, and Weizhu Chen. LoRA: Low-rank adaptation of large language models. In *International Conference on Learning Representations*, 2022. URL https://openreview.net/forum?id=nZeVKeeFYf9.

Jianheng Huang, Leyang Cui, Ante Wang, Chengyi Yang, Xinting Liao, Linfeng Song, Junfeng Yao, and Jinsong Su. Mitigating catastrophic forgetting in large language models with self-synthesized rehearsal. In Lun-Wei Ku, Andre Martins, and Vivek Srikumar (eds.), *Proceedings of the 62nd Annual Meeting of the Association for Computational Linguistics (Volume 1: Long Papers)*, pp. 1416–1428, Bangkok, Thailand, August 2024. Association for Computational Linguistics. doi: 10.18653/v1/2024.acl-long.77. URL https://aclanthology.org/2024.acl-long.77/.

Jonas Hübotter, Frederike Lübeck, Lejs Behric, Anton Baumann, Marco Bagatella, Daniel Marta, Ido Hakimi, Idan Shenfeld, Thomas Kleine Buening, Carlos Guestrin, and Andreas Krause. Reinforcement learning via Self-Distillation. In *Proceedings of the 43rd International Conference on Machine Learning*, 2026. URL https://openreview.net/forum?id=QkfkxyRizZ.

Qiao Jin, Bhuwan Dhingra, Zhengping Liu, William Cohen, and Xinghua Lu. PubMedQA: A dataset for biomedical research question answering. In Kentaro Inui, Jing Jiang, Vincent Ng, and Xiaojun Wan (eds.), *Proceedings of the 2019 Conference on Empirical Methods in Natural Language Processing and the 9th International Joint Conference on Natural Language Processing (EMNLP-IJCNLP)*, pp. 2567–2577, Hong Kong, China, November 2019. Association for Computational Linguistics. doi: 10.18653/v1/D19-1259. URL https://aclanthology.org/D19-1259/.

Yoon Kim and Alexander M. Rush. Sequence-level knowledge distillation. In Jian Su, Kevin Duh, and Xavier Carreras (eds.), *Proceedings of the 2016 Conference on Empirical Methods in Natural Language Processing*, pp. 1317–1327, Austin, Texas, November 2016. Association for Computational Linguistics. doi: 10.18653/v1/D16-1139. URL https://aclanthology.org/D16-1139/.

James Kirkpatrick, Razvan Pascanu, Neil Rabinowitz, Joel Veness, Guillaume Desjardins, Andrei A. Rusu, Kieran Milan, John Quan, Tiago Ramalho, Agnieszka Grabska-Barwinska, Demis Hassabis,

Claudia Clopath, Dharshan Kumaran, and Raia Hadsell. Overcoming catastrophic forgetting in neural networks. *Proceedings of the National Academy of Sciences*, 114(13):3521–3526, 2017. doi: 10.1073/pnas.1611835114. URL https://doi.org/10.1073/pnas.1611835114.

Yongchan Kwon, Eric Wu, Kevin Wu, and James Zou. DataInf: Efficiently estimating data influence in LoRA-tuned LLMs and diffusion models. In *International Conference on Learning Representations*, 2024. URL https://openreview.net/forum?id=9m02ib92Wz.

Zhizhong Li and Derek Hoiem. Learning without forgetting. *IEEE Transactions on Pattern Analysis and Machine Intelligence*, 40(12):2935–2947, 2018. doi: 10.1109/TPAMI.2017.2773081. URL https://doi.org/10.1109/TPAMI.2017.2773081.

Hunter Lightman, Vineet Kosaraju, Yuri Burda, Harrison Edwards, Bowen Baker, Teddy Lee, Jan Leike, John Schulman, Ilya Sutskever, and Karl Cobbe. Let's verify step by step. In *International Conference on Learning Representations*, 2024. URL https://openreview.net/forum?id=v8L0pN6EOi.

Stephanie Lin, Jacob Hilton, and Owain Evans. TruthfulQA: Measuring how models mimic human falsehoods. In Smaranda Muresan, Preslav Nakov, and Aline Villavicencio (eds.), *Proceedings of the 60th Annual Meeting of the Association for Computational Linguistics (Volume 1: Long Papers)*, pp. 3214–3252, Dublin, Ireland, May 2022. Association for Computational Linguistics. doi: 10.18653/v1/2022.acl-long.229. URL https://aclanthology.org/2022.acl-long.229/.

David Lopez-Paz and Marc'Aurelio Ranzato. Gradient episodic memory for continual learning. In *Advances in Neural Information Processing Systems*, volume 30, 2017. URL https://arxiv.org/abs/1706.08840.

Ilya Loshchilov and Frank Hutter. Decoupled weight decay regularization. In *International Conference on Learning Representations*, 2019. URL https://openreview.net/forum?id=Bkg6RiCqY7.

Gustav Olaf Yunus Laitinen-Fredriksson Lundstrom-Imanov. Mechanistic analysis of catastrophic forgetting in large language models during continual fine-tuning. arXiv preprint arXiv:2601.18699, 2026. URL https://arxiv.org/abs/2601.18699.

Yun Luo, Zhen Yang, Fandong Meng, Yafu Li, Jie Zhou, and Yue Zhang. An empirical study of catastrophic forgetting in large language models during continual fine-tuning. *IEEE Transactions on Audio, Speech and Language Processing*, 2025. URL https://ieeexplore.ieee.org/document/11151751/.

Kevin Meng, Arnab Sen Sharma, Alex J. Andonian, Yonatan Belinkov, and David Bau. Mass-editing memory in a transformer. In *International Conference on Learning Representations*, 2023. URL https://openreview.net/forum?id=MkbcAHIYgyS.

Eric Mitchell, Charles Lin, Antoine Bosselut, Chelsea Finn, and Christopher D Manning. Fast model editing at scale. In *International Conference on Learning Representations*, 2022a. URL https://openreview.net/forum?id=0DcZxeWfOPt.

Eric Mitchell, Charles Lin, Antoine Bosselut, Christopher D Manning, and Chelsea Finn. Memory-based model editing at scale. In *Proceedings of the 39th International Conference on Machine Learning*, volume 162 of *Proceedings of Machine Learning Research*, pp. 15817–15831. PMLR, 2022b. URL https://proceedings.mlr.press/v162/mitchell22a.html.

Moni Naor. Evaluation may be easier than generation. In *Proceedings of the Twenty-Eighth Annual ACM Symposium on Theory of Computing*, pp. 74–83. Association for Computing Machinery, 1996. doi: 10.1145/237814.237833. URL https://doi.org/10.1145/237814.237833.

Long Ouyang, Jeff Wu, Xu Jiang, Diogo Almeida, Carroll L. Wainwright, Pamela Mishkin, Chong Zhang, Sandhini Agarwal, Katarina Slama, Alex Ray, John Schulman, Jacob Hilton, Fraser Kelton, Luke Miller, Maddie Simens, Amanda Askell, Peter Welinder, Paul Christiano, Jan Leike, and Ryan Lowe. Training language models to follow instructions with human feedback. In *Advances in Neural Information Processing Systems*, volume 35, pp. 27730–27744, 2022. URL https://arxiv.org/abs/2203.02155.

Ankit Pal, Logesh Kumar Umapathi, and Malaikannan Sankarasubbu. Medmcqa: A large-scale multi-subject multi-choice dataset for medical domain question answering. In Gerardo Flores, George H Chen, Tom Pollard, Joyce C Ho, and Tristan Naumann (eds.), *Proceedings of the Conference on Health, Inference, and Learning*, volume 174 of *Proceedings of Machine Learning Research*, pp. 248–260. PMLR, 07–08 Apr 2022. URL https://proceedings.mlr.press/v174/pal22a.html.

Qwen Team. Qwen2.5-3B-Instruct: Model card. Hugging Face model card, 2024. URL https://huggingface.co/Qwen/Qwen2.5-3B-Instruct. Accessed September 25, 2026.

Rafael Rafailov, Archit Sharma, Eric Mitchell, Christopher D Manning, Stefano Ermon, and Chelsea Finn. Direct preference optimization: Your language model is secretly a reward model. In *Advances in Neural Information Processing Systems*, volume 36, pp. 53728–53741, 2023. URL https://arxiv.org/abs/2305.18290.

Anastasia Razdaibiedina, Yuning Mao, Rui Hou, Madian Khabsa, Mike Lewis, and Amjad Almahairi. Progressive prompts: Continual learning for language models. In *International Conference on Learning Representations*, 2023. URL https://openreview.net/forum?id=UJTgQBc91_.

Stéphane Ross, Geoffrey Gordon, and Drew Bagnell. A reduction of imitation learning and structured prediction to no-regret online learning. In *Proceedings of the Fourteenth International Conference on Artificial Intelligence and Statistics*, volume 15 of *Proceedings of Machine Learning Research*, pp. 627–635. PMLR, 2011. URL https://proceedings.mlr.press/v15/ross11a.html.

Keisuke Sakaguchi, Ronan Le Bras, Chandra Bhagavatula, and Yejin Choi. WinoGrande: An adversarial Winograd schema challenge at scale. *Communications of the ACM*, 64(9):99–106, 2021. doi: 10.1145/3474381. URL https://doi.org/10.1145/3474381.

John Schulman, Filip Wolski, Prafulla Dhariwal, Alec Radford, and Oleg Klimov. Proximal policy optimization algorithms. arXiv preprint arXiv:1707.06347, 2017. URL https://arxiv.org/abs/1707.06347.

Thomas Scialom, Tuhin Chakrabarty, and Smaranda Muresan. Fine-tuned language models are continual learners. In Yoav Goldberg, Zornitsa Kozareva, and Yue Zhang (eds.), *Proceedings of the 2022 Conference on Empirical Methods in Natural Language Processing*, pp. 6107–6122, Abu Dhabi, United Arab Emirates, December 2022. Association for Computational Linguistics. doi: 10.18653/v1/2022.emnlp-main.410. URL https://aclanthology.org/2022.emnlp-main.410/.

Idan Shenfeld, Mehul Damani, Jonas Hübotter, and Pulkit Agrawal. Self-Distillation enables continual learning. In *Proceedings of the 43rd International Conference on Machine Learning*, 2026. URL https://openreview.net/forum?id=qA6FgH0nnZ.

Qiaoyu Tang, Ziliang Deng, Hongyu Lin, Xianpei Han, Qiao Liang, Boxi Cao, and Le Sun. ToolAlpaca: Generalized tool learning for language models with 3000 simulated cases. arXiv preprint arXiv:2306.05301, 2023. URL https://arxiv.org/abs/2306.05301.

Xiao Wang, Yuansen Zhang, Tianze Chen, Songyang Gao, Senjie Jin, Xianjun Yang, Zhiheng Xi, Rui Zheng, Yicheng Zou, Tao Gui, Qi Zhang, and Xuanjing Huang. TRACE: A comprehensive benchmark for continual learning in large language models. arXiv preprint arXiv:2310.06762, 2023a. URL https://arxiv.org/abs/2310.06762.

Xuezhi Wang, Jason Wei, Dale Schuurmans, Quoc V. Le, Ed H. Chi, Sharan Narang, Aakanksha Chowdhery, and Denny Zhou. Self-Consistency improves chain of thought reasoning in language models. In *International Conference on Learning Representations*, 2023b. URL https://openreview.net/forum?id=1PL1NIMMrw.

Yifan Wang, Yafei Liu, Chufan Shi, Haoling Li, Chen Chen, Haonan Lu, and Yujiu Yang. InsCL: A data-efficient continual learning paradigm for fine-tuning large language models with instructions. In Kevin Duh, Helena Gomez, and Steven Bethard (eds.), *Proceedings of the 2024*

Conference of the North American Chapter of the Association for Computational Linguistics: Human Language Technologies (Volume 1: Long Papers), pp. 663–677, Mexico City, Mexico, June 2024. Association for Computational Linguistics. doi: 10.18653/v1/2024.naacl-long.37. URL https://aclanthology.org/2024.naacl-long.37/.

Kai Wei, Rishabh Iyer, and Jeff Bilmes. Submodularity in data subset selection and active learning. In Francis Bach and David Blei (eds.), *Proceedings of the 32nd International Conference on Machine Learning*, volume 37 of *Proceedings of Machine Learning Research*, pp. 1954–1963, Lille, France, 07–09 Jul 2015. PMLR. URL https://proceedings.mlr.press/v37/wei15.html.

Mengzhou Xia, Sadhika Malladi, Suchin Gururangan, Sanjeev Arora, and Danqi Chen. LESS: Selecting influential data for targeted instruction tuning. In *Proceedings of the 41st International Conference on Machine Learning*, 2024. URL https://openreview.net/forum?id=PG5fV50maR.

Jaehong Yoon, Divyam Madaan, Eunho Yang, and Sung Ju Hwang. Online coreset selection for rehearsal-based continual learning. In *International Conference on Learning Representations*, 2022. URL https://openreview.net/forum?id=f9D-5WNG4Nv.

Rowan Zellers, Ari Holtzman, Yonatan Bisk, Ali Farhadi, and Yejin Choi. HellaSwag: Can a machine really finish your sentence? In Anna Korhonen, David Traum, and Lluís Màrquez (eds.), *Proceedings of the 57th Annual Meeting of the Association for Computational Linguistics*, pp. 4791–4800, Florence, Italy, July 2019. Association for Computational Linguistics. doi: 10.18653/v1/P19-1472. URL https://aclanthology.org/P19-1472/.

Di Zhang, Wei Liu, Qian Tan, Jingdan Chen, Hang Yan, Yuliang Yan, Jiatong Li, Weiran Huang, Xiangyu Yue, Wanli Ouyang, Dongzhan Zhou, Shufei Zhang, Mao Su, Han-Sen Zhong, and Yuqiang Li. ChemLLM: A chemical large language model. arXiv preprint arXiv:2402.06852, 2024. URL https://arxiv.org/abs/2402.06852v2.

Siyan Zhao, Zhihui Xie, Mengchen Liu, Jing Huang, Guan Pang, Feiyu Chen, and Aditya Grover. Self-Distilled Reasoner: On-policy self-distillation for large language models. In *Proceedings of the 43rd International Conference on Machine Learning*, 2026. URL https://openreview.net/forum?id=Jpxfof0EaS.

Jeffrey Zhou, Tianjian Lu, Swaroop Mishra, Siddhartha Brahma, Sujoy Basu, Yi Luan, Denny Zhou, and Le Hou. Instruction-following evaluation for large language models. arXiv preprint arXiv:2311.07911, 2023. URL https://arxiv.org/abs/2311.07911.

APPENDIX

A EXPERIMENTAL SCOPE

Table 7 separates the conclusions of the completed comparisons. The evidence for an anchored policy is not interchangeable with evidence for its score or selection rule. All full-stream findings are conditional on the reported training and evaluation protocol.

Table 7: Empirical findings and the scope of their interpretation.

Component	Empirical finding	Interpretation and scope
Anchored training	Three anchored arms improve all three mean endpoints over OP-SD in the six-seed Qwen2.5-3B stream.	The advantage does not require gradient-conflict ranking; it concerns the full anchored intervention.
Gradient-conflict ranking	Mean paired effects are negative against both scored random and bottom-ranked controls.	This implemented ordering does not explain the anchored-policy advantage in the evaluated stream.
Local score fidelity	Mean Spearman agreement with cloned AdamW loss change is 0.1009 in six saved states from one seed.	The comparison is local to the specified one-microbatch proposal, not a calibration to final retention.
Cloned-update ranking	Mean effects favor ranking over random and low-risk controls, with three of six joint wins against each.	Favorable means and mixed joint outcomes leave consistency across runs and settings unresolved.
Risk-screened coverage	Mean held-out accuracy improves over random, while mean stream performance and forgetting worsen.	The four-seed follow-up shows an endpoint trade-off and overlaps the six-seed study.
Replay complementarity	Adding the tested gradient-selected anchoring branch worsens all three mean endpoints relative to supervised replay.	The result is specific to the separate replay configuration, not a universal claim about combining retention mechanisms.

B BROADER REPLAY AND SELECTION CONTEXT

Isolation, self-distillation, and locality. Progressive Prompts isolates task-specific soft prompts while freezing the backbone (Razdaibiedina et al., 2023); model-editing methods instead optimize locality of a requested change (Mitchell et al., 2022a;b; Meng et al., 2023). These approaches change where or whether an update is applied. CANARY separates the cached-response anchoring objective from the policies that allocate its fixed budget. DAgger provides an imitation-learning precedent for supervision on learner-induced states (Ross et al., 2011); it is not itself a language-model self-distillation method. Generalized on-policy distillation (Agarwal et al., 2024), privileged-context OPSD (Zhao et al., 2026), and continual-learning SDFT (Shenfeld et al., 2026) instantiate different training systems. Within our anchored-arm comparisons, the OP-SD base learner and anchoring objective are fixed; only the selector policy varies.

Update gating and mechanistic scope. STABLE gates low-rank updates under a stability budget (Hoy & Celik, 2025), whereas our comparisons hold the anchoring objective fixed and vary allocation. A recent survey organizes continual learning across language-model training stages (Chen et al., 2026); a separate preprint studies forgetting mechanistically (Lundstrom-Imanov, 2026). These are broader perspectives, not evidence that the present behavioral comparisons identify an internal mechanism or generalize across training stages.

Other post-training objectives. Few-shot in-context learning changes the information in a prompt without requiring a parameter update (Brown et al., 2020); our privileged teacher context instead supplies targets for parameter training. Reinforcement learning via self-distillation uses rich textual feedback for self-teacher supervision (Hübotter et al., 2026), a different information source from fixed stage-start responses. PPO is a policy-optimization method (Schulman et al., 2017); InstructGPT combines demonstrations, preference modeling, and reinforcement learning (Ouyang et al., 2022);

Table 8: Closest replay and selection families. The comparison is conceptual; it does not imply matched empirical coverage.

Family	Stored or generated object	Selection / retention signal	Relation to CANARY
MIR (Aljundi et al., 2019a)	Stored memories or generated replay samples	Prospective loss increase under a virtual update	Establishes the interference-aware retrieval principle; the present study tests attribution under cached LLM targets.
CT0 / InsCL (Scialom et al., 2022; Wang et al., 2024)	Prior instruction examples	Fixed replay or instruction-level similarity/information	Continual instruction tuning, but not a prospective update-risk estimation.
SSR (Huang et al., 2024)	Synthesized examples	Quality and diversity filtering	Avoids retaining old real examples; uses synthesized rehearsal rather than our fixed stage-start targets.
SEEKR / KPIG (He et al., 2024; 2025)	Replay examples; attention heads or instruction parts	Head forgettability/sensitivity or masked instruction-part information gain	Selects attention targets or replay instructions, not prompt-token coverage.
DER (Buzzega et al., 2020)	Examples and historical logits	Reservoir sampling and output consistency	Conceptual ancestor of cached behavior; its objects are classifier logits, not autoregressive responses.
LESS / DataInf (Xia et al., 2024; Kwon et al., 2024)	Candidate tuning examples	Gradient similarity or influence	Targets acquisition utility or attribution, not damage to retained behavior.
Online coreset (Yoon et al., 2022)	Rehearsal examples	Joint representativeness and informativeness	General-CL evidence for set-level selection, not LLM-specific evidence.
CANARY (this work)	Prompts and stage-start responses	Cached-response anchoring with gradient-conflict, cloned-update risk, and risk-screened coverage policies	Separates the anchoring advantage from the performance of the tested risk-guided selectors.

DPO directly optimizes a preference objective (Rafailov et al., 2023). We do not evaluate these as alternative base learners or claim that our KL-plus-anchor objective reproduces them.

Generation, selection, and verification. In a formal distribution-learning setting, Naor (1996) separates the difficulty of learning probability evaluators from learning generators; this is not a theorem that checking arbitrary LLM answers is easier than generating them. Self-consistency aggregates answers from sampled reasoning paths (Wang et al., 2023b), while process supervision provides feedback on intermediate reasoning steps (Lightman et al., 2024). Neither procedure is implemented by increasing the likelihood of a single cached response. An anchoring target therefore carries no independent correctness guarantee.

Coverage and filtering precedents. Facility-location objectives and filtered active submodular selection combine representativeness with an informativeness screen (Wei et al., 2015). Our screen instead uses cached-response loss change under a cloned optimizer proposal, with provenance quotas and coverage evaluated over the full local pool. This is a specific retention-policy test, not a claim to originate filter-then-cover selection or to inherit a downstream performance guarantee.

Benchmark scope. TRACE evaluates broader heterogeneous continual-LLM adaptation (Wang et al., 2023a). Our three-stage stream and five retained true-500 slices cover a narrower setting. Complementary evaluation targets include commonsense completion in HellaSwag (Zellers et al., 2019), pronoun-resolution reasoning in WinoGrande (Sakaguchi et al., 2021), truthfulness in TruthfulQA (Lin et al., 2022), functional code correctness in HumanEval (Chen et al., 2021), and verifiable instruction following in IFEval (Zhou et al., 2023). These five additional benchmarks are context for possible extensions, not evaluations performed in this paper. The selector results do not establish benchmark-level superiority or broad continual-learning generalization.

C Data and Experimental Protocol

Table 9: Controlled task stream. Dataset names identify the configured sources; all arms use identical row slices.

Stage	Domain	Source	Role in the controlled stream
1	Tool use	ToolAlpaca (Tang et al., 2023)	Immutable shared stage-1 adapter and optimizer state.
2	Chemistry	ChemData700K (Zhang et al., 2024)	First continued-training stage; reservoir contains 64 tool prompts.
3	Medical	medical-o1-reasoning-SFT (Freedom-Intelligence, 2024; Chen et al., 2025)	Second continuation; reservoir contains 64 tool and 64 chemistry prompts.

Table 10: Configuration used for the selector comparison.

Component	Value
Model / adaptation	`Qwen/Qwen2.5-3B-Instruct`; upstream revision not recorded; LoRA rank 16, alpha 32, dropout 0.05
Optimization	learning rate 10^{-4}; batch size 1; accumulation 4; 24 optimizer steps per stage
Per-stage slices	128 train; 32 evaluation; 64 new reservoir rows
Selection	32 scored candidates; 8 anchors; refresh every 4 steps; anchor weight $\beta = 0.3$
Risk-screened coverage	Hamilton provenance quotas; top-$2q_g$ risk screen; within-stratum token-Jaccard facility location
Seed blocks	410–415 for pointwise-risk studies; 410–413 for coverage; shared stage-1 state within each block
Primary evaluation	SeenAvg, Average Forgetting (Eq. 4), and five 500-example True500 slices

The reservoir has no exact within-reservoir or cross-stage prompt duplicates. One upstream chemistry prompt is semantically duplicated across the index-disjoint train and reservoir slices. Cached targets

use greedy stage-start responses, with original dataset targets as the empty-output fallback, and remain fixed within each 24-step stage. Because evaluation can change with harness, decoding, and sample-selection details, we hold the checkpoint-evaluation schedule, slices, decoding, and per-benchmark counts fixed across arms (Biderman et al., 2024).

D Implementation and Evaluation Details

D.1 Endpoint definitions and paired signs

The endpoint aggregates are defined in Section 3.1, Eqs. (3)–(5). Throughout, $\Delta\text{Forget} = \text{AF}_T^{\text{comparator}} - \text{AF}_T^{\text{treatment}}$, so a positive paired change favors the named treatment on each endpoint. A joint-frontier win requires no worsening of any endpoint and at least one improvement. It is a descriptive multi-endpoint criterion, not a significance test. Outcomes and paired effects are displayed separately at four-decimal precision. Aggregation scripts read the unrounded run values, compute paired effects, means, and sample SDs, and round only for display; displayed deltas therefore need not equal subtraction of already rounded endpoint cells.

D.2 Implemented losses

For an example with reference-answer token count n, OP-SD samples $L = \min(64, \max(8, n))$ continuation tokens from the current model under the ordinary prompt, using temperature 0.8, top-p 0.95, and a minimum of eight new tokens. The prompt prefix is left-truncated as needed so that prompt and continuation fit 1,024 tokens. The teacher context contains the original task, the dataset reference answer, and an instruction to answer the original task. The same sampled continuation is appended to the ordinary student context and this privileged teacher context. The current model in evaluation mode and under `no_grad` supplies teacher logits; the train-mode forward pass supplies student logits. The implementation computes $D_{\text{KL}}(p_{\text{teacher}} \| p_{\text{student}})$ at each continuation position, averages over continuation positions and then over examples, and repeats this procedure at every microbatch.

Stage-start cache generation is greedy with at most 96 new tokens. If decoding is empty, the implementation falls back to the reservoir row's original target. Anchor NLL uses the model's causal-language-model loss after masking the rendered prompt and right padding; overlength prompt–response sequences are truncated from the left. The loss is a mean over all unmasked response tokens in the selected batch. At each of four accumulated microbatches, the training code adds 0.3 times this anchor loss to OP-SD and divides the total by four before backpropagation.

D.3 Task and held-out evaluators

All three stream tasks use the same normalized token-F1 scorer. Text is stripped and lowercased, ASCII punctuation and the articles `a`/`an`/`the` are removed, whitespace is collapsed, and multiset token overlap determines precision and recall. The reported task score is the arithmetic mean across 32 rows. ToolAlpaca uses its `prompt`/`target` fields; chemistry concatenates `instruction` and nonempty `input` and scores against `output`; medical uses `Question`/`Response`. Generation is greedy with at most 96 new tokens.

The held-out evaluator also uses greedy decoding. SciKnowEval Chemistry, MMLU, and MedMCQA receive at most eight new tokens and are scored from the last standalone A–E token; no match scores zero. GSM8K receives at most 96 tokens; the last signed decimal-like substring is compared after removing commas and dollar signs, with failed extraction scored zero. PubMedQA receives at most eight tokens and uses the same lowercase, punctuation/article-stripped normalization for exact match. The frozen local slices contain the first 500 eligible SciKnowEval test rows with domain `Chemistry` and type `mcq-4-choices`, the first 500 GSM8K test rows, 500 MMLU test rows selected round-robin over lexicographically sorted subjects, the first 500 MedMCQA validation rows, and the first 500 labeled PubMedQA training rows. These rows are serialized under `data/offline/benchmarks_500` and reused across arms.

D.4 MATCHED CONDITIONS

The matching specification in Table 11 applies within each seed block. Branches share their initial state and protocol; their subsequent states, cached target realizations, sampled pools, and score maps may diverge under their respective policies. This evaluates complete training policies, whereas same-state diagnostics hold a local comparison fixed. All four coverage arms complete every selected block and all final held-out slices. No missing evaluation is imputed.

Table 11: Matched initialization and protocol for the four-seed selector study. Realized post-intervention trajectories are not held fixed.

Component	Matched quantities
Initial state	Base-model identity, LoRA adapter, AdamW moments, scheduler position, training history, and random-number generator states.
Training protocol	Task order, row partitions, accumulation schedule, optimizer-step budget, and stopping point.
Anchor protocol	Reservoir sources, candidate count, provenance labels, stage-start target-generation rule, active budget, and refresh schedule; later candidate pools and score maps may differ.
Evaluation	Each arm's final checkpoint evaluated on the same benchmark rows, decoding parameters, parsers, and 500 examples in each of the five TRUE500 slices.
Reporting	Complete training seed as the independent unit, inclusion of every valid run, and no imputation of missing endpoints.

D.5 PRACTICAL-EFFECT CRITERIA

Table 12 summarizes the coverage comparisons against the practical-effect criteria stated in Section 5. The numerical endpoint targets concern *mean paired effects*; the joint-frontier target concerns the number of complete seeds satisfying the three-endpoint condition. The same criteria are applied relative to both random and cloned-update risk. They provide a descriptive assessment of effect size and multi-endpoint consistency, not a significance test.

Table 12: Practical-effect criteria for the coverage comparisons. Endpoint thresholds apply to mean paired effects; the joint-win threshold applies to seeds. Forgetting is comparator minus risk-screened coverage, so nonnegative values favor the risk-screened policy.

Comparator	Criterion	Target	Observed mean $\pm$ SD	Meets target?
Random	ΔSeen	≥ 0.0100	-0.0049 ± 0.0160	No
Random	ΔForget	≥ 0	-0.0126 ± 0.0209	No
Random	ΔTrue500	≥ 0.0200	$+0.0311 \pm 0.0385$	Yes
Random	Joint frontier	$\geq 3/4$	$0/4$	No
Cloned-update	ΔSeen	≥ 0.0100	-0.0185 ± 0.0188	No
Cloned-update	ΔForget	≥ 0	-0.0193 ± 0.0053	No
Cloned-update	ΔTrue500	≥ 0.0200	-0.0141 ± 0.0228	No
Cloned-update	Joint frontier	$\geq 3/4$	$0/4$	No

Risk-screened coverage meets the mean TRUE500 target relative to random, but not the full criterion set relative to either primary comparator. This follow-up was not extended to additional task orders or model families. Its scope therefore remains the reported four-seed stream; the observed outcomes do not establish what would happen in untested settings.

E BEHAVIOR OF THE RISK-SCREENED COVERAGE SELECTOR

E.1 SAME-STATE ALLOCATION DIAGNOSTICS

On 48 saved refresh traces, allocation rules are compared at shared states using selection overlap, risk percentiles, and facility coverage. These are local allocation diagnostics, not downstream performance

measurements. A fixed equal-weight risk/coverage alternative shared 7.58 of eight anchors with global cloned-update top-8. The constrained top-$2q_g$ policy shared 3.85, with mean selected risk percentile 0.744 versus 0.887 for global top-8. Coverage-only shared 1.75 anchors with global top-8. Mean normalized facility coverage was 0.417 for global cloned-update top-8 and 0.439 for the constrained screen. The evaluated screen multiplier is 2. These observations quantify the distinctness of the allocation rules on shared traces; they do not measure retention gains.

E.2 Realized selection during training

Table 13 summarizes the 48 refreshes per coverage-based arm. Candidate and selection counts, quota reconstruction, and eligibility checks identify the intervention that actually ran. The minimum score gap at a top-$2q_g$ screen boundary is 8.94×10^{-7}. All 384 risk-screened selections satisfy the eligibility rule. Pooled-anchor and per-refresh risk-percentile SDs describe different units and are reported separately.

Table 13: Selector characteristics for the two coverage-based arms. SDs are sample SDs at the unit named in the last column.

Diagnostic	Risk-screened	Coverage-only	Unit
Valid refreshes / candidate–selection sizes	48/48; 32→8	48/48; 32→8	refresh
Quota histogram reconstructs exactly	48/48	48/48	refresh
All selections inside top-$2q_g$ screen	48/48	not applicable	refresh
Selected risk percentile, pooled	0.7586 ± 0.1537	0.4670 ± 0.2978	384 anchors
Mean risk percentile per refresh	0.7586 ± 0.0390	0.4670 ± 0.0834	48 refreshes
Overlap with global cloned-update top 8	3.833 ± 0.808	1.646 ± 1.021	48 refreshes
Normalized facility coverage	0.5142 ± 0.0208	0.5250 ± 0.0209	48 refreshes
Selected prompt length	557.5 ± 169.4	595.2 ± 179.4	384 anchors
Selected prompts over 1024 tokens	16/384	24/384	anchor event

E.3 Initialization, identities, and trajectory dependence

Corresponding ordered prompt pools match at all 48 paired refreshes when identified by a SHA-256 digest of prompt plus system text, excluding the cached response. Full cached-example hashes, which additionally include the arm-dependent target and metadata, match throughout the 24 chemistry refreshes. Their medical-stage set intersection is 0.875 ± 1.154 of 32 (range 0–4), reflecting different cached-example records rather than different sampled prompts. Because the full hash includes targets and metadata, this comparison alone does not identify which record field differs. Score maps match at the four first refreshes, and no medical score map matches across the two coverage arms. Later diagnostics are therefore observations along different policy trajectories, not independent replications or fixed-state estimates of an isolated risk-screen effect.

Similarity uses the untruncated rendered prompt, whereas training truncates at 1024 tokens. The prompt-length diagnostics therefore also test alignment between what selection represents and what the training loss consumes. Refreshes and anchor events remain nested observations, not extra independent training samples.

F Statistical Unit and Reporting

Within each contrast, a complete paired training seed is the independent unit: six seeds (410–415) for the pointwise-risk comparisons and four overlapping seeds (410–413) for the coverage follow-up. Means and sample SDs are computed at that level. Stages, saved states, refreshes, candidates, selected anchors, and individual evaluation rows are repeated or nested observations. The six-state score-fidelity analysis comes from one seed and is not six independent training replications.

We report endpoint magnitudes, sample SDs, seed-level effects, and joint-win counts. A sample SD describes run-to-run heterogeneity; it is not a confidence interval for the mean. A joint-win count describes the observed Pareto criterion and does not replace the endpoint effects or a significance test. Table 14 gives unadjusted two-sided 95% paired-t intervals from the six unrounded seed effects and five degrees of freedom. They assume independent seed blocks and approximate normality of paired

effects. An interval containing zero is not evidence of equivalence. No refresh-level significance claim is made. The five held-out slices provide a descriptive decomposition, not confirmatory tests of domain-level hypotheses; inspecting them does not support claims of domain-specific superiority.

Table 14: Mean paired effect [unadjusted 95% paired-t interval]. Forgetting is comparator minus top-8, so positive favors top-8.

Contrast	ΔSeen	ΔForget	ΔTrue500
Gradient top − OP-SD	.0490 [.0245, .0734]	.0291 [.0090, .0491]	.3091 [.2284, .3897]
Gradient top − gradient-random	−.0103 [−.0237, .0032]	−.0098 [−.0259, .0063]	−.0247 [−.0514, .0019]
Cloned top − OP-SD	.0712 [.0483, .0942]	.0461 [.0261, .0660]	.3644 [.3072, .4216]
Cloned top − cloned-random	.0120 [−.0097, .0337]	.0072 [−.0170, .0315]	.0306 [−.0249, .0861]

G Principal Six-Seed Outcomes

Tables 16 and 17 report every endpoint used in the two principal six-seed comparisons. The protocol-qualified random labels identify the scoring setup described for each comparison. Their reported endpoint vectors match in all six blocks. Each contrast contains six paired seed blocks; listing the same endpoint values under two protocol labels does not supply additional independent replications.

G.1 Stagewise acquisition and retention

Table 15 exposes the task-score matrix for the shared stage-1 checkpoint, OP-SD, and gradient-scored random anchoring. It reports the normalized token-F1 measurements that underlie SeenAvg and forgetting, not verified task success. Stage-2 held-out generations and parser-validity records were not retained, so the available artifacts cannot decompose the held-out endpoint into valid-output rate and accuracy among valid outputs. When correctness implies validity, strict accuracy equals their product; this is descriptive accounting, not a causal competence/format decomposition, because valid-output subsets can differ by policy.

Table 15: Per-seed task scores at acquisition and after subsequent stages. S1 Tool is shared within each seed block; C2 denotes the post-chemistry checkpoint and M3 the post-medical checkpoint.

Seed	Arm	S1 Tool	C2 Tool	C2 Chem	M3 Tool	M3 Chem	M3 Med
410	OP-SD	.1899	.0214	.0027	.0288	.0243	.1886
410	Grad.-random	.1899	.1114	.0609	.0807	.0704	.2124
411	OP-SD	.1438	.0467	.0467	.0298	.0109	.0365
411	Grad.-random	.1438	.0885	.0349	.0809	.0305	.0636
412	OP-SD	.1794	.0740	.0506	.0218	.0413	.2095
412	Grad.-random	.1794	.1102	.0681	.1765	.1100	.2971
413	OP-SD	.1164	.0155	.0020	.0151	.0013	.0008
413	Grad.-random	.1164	.0739	.0440	.0716	.0423	.1230
414	OP-SD	.1376	.0316	.0757	.0217	.0413	.2109
414	Grad.-random	.1376	.0644	.1314	.0655	.0767	.2626
415	OP-SD	.1075	.0628	.0417	.0567	.0330	.1408
415	Grad.-random	.1075	.0741	.0317	.0639	.0747	.2773

Table 16: All gradient-scored arm–seed endpoints.

Seed	Arm	Seen	Forget	True500	Seed	Arm	Seen	Forget	True500
410	OP-SD	.0806	.0697	.1480	413	OP-SD	.0057	.0510	.0000
410	Grad.-top	.1046	.0571	.4048	413	Grad.-top	.0762	.0243	.2496
410	Grad.-random	.1212	.0499	.4208	413	Grad.-random	.0789	.0233	.3152
410	Grad.-bottom	.0971	.0526	.4216	413	Grad.-bottom	.1023	.0139	.3776
411	OP-SD	.0257	.0749	.0036	414	OP-SD	.0913	.0752	.0072
411	Grad.-top	.0465	.0453	.2224	414	Grad.-top	.1331	.0544	.3852
411	Grad.-random	.0583	.0337	.2628	414	Grad.-random	.1349	.0634	.4112
411	Grad.-bottom	.0698	.0452	.3912	414	Grad.-bottom	.1223	.0561	.4312
412	OP-SD	.0909	.0835	.0040	415	OP-SD	.0768	.0298	.0452
412	Grad.-top	.1627	.0174	.4072	415	Grad.-top	.1419	.0112	.3932
412	Grad.-random	.1945	−.0194	.4032	415	Grad.-random	.1387	.0003	.3976
412	Grad.-bottom	.1749	.0156	.4192	415	Grad.-bottom	.1280	.0186	.4072

Table 17: All cloned-update-scored arm–seed endpoints.

Seed	Arm	Seen	Forget	True500	Seed	Arm	Seen	Forget	True500
410	OP-SD	.0806	.0697	.1480	413	OP-SD	.0057	.0510	.0000
410	Cloned-top	.1264	.0219	.4144	413	Cloned-top	.1137	.0089	.3772
410	Cloned-random	.1212	.0499	.4208	413	Cloned-random	.0789	.0233	.3152
410	Low-risk	.1402	.0330	.4192	413	Low-risk	.0633	.0303	.2908
411	OP-SD	.0257	.0749	.0036	414	OP-SD	.0913	.0752	.0072
411	Cloned-top	.0919	.0207	.3868	414	Cloned-top	.1520	.0342	.4252
411	Cloned-random	.0583	.0337	.2628	414	Cloned-random	.1349	.0634	.4112
411	Low-risk	.0738	.0404	.1912	414	Low-risk	.1225	.0604	.3976
412	OP-SD	.0909	.0835	.0040	415	OP-SD	.0768	.0298	.0452
412	Cloned-top	.1757	.0088	.4044	415	Cloned-top	.1389	.0133	.3864
412	Cloned-random	.1945	−.0194	.4032	415	Cloned-random	.1387	.0003	.3976
412	Low-risk	.1718	.0257	.4056	415	Low-risk	.1248	.0029	.4056

Table 18: Complete seed-level top-8 paired effects. G is gradient-conflict and C is cloned-update; their random controls are protocol-specific. Each triplet is ΔSeen, ΔForget, ΔTrue500, where positive favors top-8. The final comparator is gradient-bottom for G and low-risk for C.

		vs. OP-SD			vs. own random			vs. bottom/low		
Protocol	Seed	S	F	T	S	F	T	S	F	T
G	410	.0240	.0126	.2568	−.0166	−.0072	−.0160	.0074	−.0045	−.0168
G	411	.0207	.0296	.2188	−.0119	−.0116	−.0404	−.0233	−.0001	−.1688
G	412	.0718	.0660	.4032	−.0318	−.0368	.0040	−.0122	−.0018	−.0120
G	413	.0705	.0267	.2496	−.0028	−.0010	−.0656	−.0261	−.0104	−.1280
G	414	.0419	.0208	.3780	−.0018	.0090	−.0260	.0108	.0017	−.0460
G	415	.0651	.0186	.3480	.0033	−.0109	−.0044	.0139	.0074	−.0140
C	410	.0458	.0479	.2664	.0052	.0280	−.0064	−.0139	.0111	−.0048
C	411	.0662	.0542	.3832	.0336	.0130	.1240	.0181	.0197	.1956
C	412	.0848	.0747	.4004	−.0189	−.0282	.0012	.0039	.0169	−.0012
C	413	.1080	.0421	.3772	.0347	.0144	.0620	.0504	.0214	.0864
C	414	.0607	.0410	.4180	.0171	.0292	.0140	.0295	.0262	.0276
C	415	.0621	.0165	.3412	.0003	−.0130	−.0112	.0141	−.0103	−.0192

H Run-Level Selector Results

Table 19: All selector arm–seed outcomes. Risk-screened denotes risk-screened coverage. All runs restore matched initial states and complete final evaluation.

Seed	Arm	Seen	Forgetting	True500
410	Risk-screened	0.1294	0.0485	0.4156
410	Coverage-only	0.1266	0.0431	0.4168
410	Cloned-update	0.1264	0.0219	0.4144
410	Random	0.1212	0.0499	0.4208
411	Risk-screened	0.0611	0.0361	0.3412
411	Coverage-only	0.0686	0.0288	0.2956
411	Cloned-update	0.0919	0.0207	0.3868
411	Random	0.0583	0.0337	0.2628
412	Risk-screened	0.1667	0.0242	0.4084
412	Coverage-only	0.1732	0.0191	0.4124
412	Cloned-update	0.1757	0.0088	0.4044
412	Random	0.1945	−0.0194	0.4032
413	Risk-screened	0.0763	0.0288	0.3612
413	Coverage-only	0.1116	0.0048	0.3900
413	Cloned-update	0.1137	0.0089	0.3772
413	Random	0.0789	0.0233	0.3152

Table 20: Seed-level effects of risk-screened coverage. Forgetting is comparator minus treatment, so positive favors risk-screened coverage.

Seed	Comparator	ΔSeen	ΔForget	ΔTrue500	Joint win
410	Random	+0.0082	+0.0014	−0.0052	0
411	Random	+0.0028	−0.0024	+0.0784	0
412	Random	−0.0279	−0.0437	+0.0052	0
413	Random	−0.0027	−0.0055	+0.0460	0
410	Cloned-update	+0.0031	−0.0266	+0.0012	0
411	Cloned-update	−0.0308	−0.0154	−0.0456	0
412	Cloned-update	−0.0090	−0.0155	+0.0040	0
413	Cloned-update	−0.0374	−0.0199	−0.0160	0
410	Coverage-only	+0.0028	−0.0054	−0.0012	0
411	Coverage-only	−0.0075	−0.0073	+0.0456	0
412	Coverage-only	−0.0066	−0.0052	−0.0040	0
413	Coverage-only	−0.0353	−0.0240	−0.0288	0

I Seed-Level Effect Geometry

Figure 2 plots the paired values in Appendix H. Seed identifiers are categorical, so observations are not connected by lines. The mean is separated from the four seed effects and is not counted as a fifth run. The largest held-out gains relative to random occur in seeds 411 and 413. Seed 410 improves both stream endpoints but loses held-out accuracy. Against cloned-update risk, forgetting improvement is negative in all four blocks. This profile explains the joint win counts without implying that every endpoint worsens in every run.

J True-500 Slice Decomposition

Risk-screened coverage minus random is $+0.0020 \pm 0.0107$ on GSM8K, $+0.0445 \pm 0.0446$ on MedMCQA, $+0.0525 \pm 0.0841$ on MMLU, -0.0030 ± 0.0363 on PubMedQA, and $+0.0595 \pm 0.0838$ on SciKnow-Chem. With four seeds, high variance, and five inspected slices, this only localizes the aggregate true-500 difference; it is not evidence for domain-level superiority.

Table 21: Per-slice accuracy (mean ± sample SD over four seeds). Every arm–seed–slice cell contains 500 examples. Risk-screened denotes risk-screened coverage.

Slice	Risk-screened	Coverage-only	Cloned-update	Random
GSM8K	0.0265 ± 0.0179	0.0205 ± 0.0161	0.0285 ± 0.0183	0.0245 ± 0.0219
MedMCQA	0.4050 ± 0.0816	0.4030 ± 0.1118	0.4175 ± 0.0629	0.3605 ± 0.1260
MMLU	0.5735 ± 0.1049	0.5330 ± 0.1964	0.5730 ± 0.1137	0.5210 ± 0.1887
PubMedQA	0.2160 ± 0.1427	0.2590 ± 0.1787	0.2450 ± 0.1616	0.2190 ± 0.1741
SciKnow-Chem	0.6870 ± 0.1088	0.6780 ± 0.1459	0.7145 ± 0.0593	0.6275 ± 0.1923

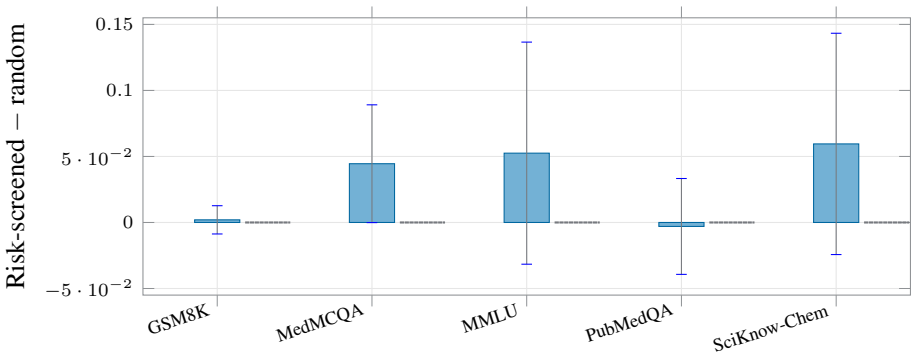


Figure 3: Descriptive decomposition of the TRUE500 contrast against random (mean ± sample SD over four paired seeds). Bars show the mean slice-level differences and error bars summarize seed heterogeneity, not uncertainty from 2,500 independent examples. Slice-level contrasts are descriptive, not confirmatory.

Table 22: Raw true-500 slice accuracies for all selector arm–seed cells.

Seed	Arm	Chem	GSM8K	MMLU	MedMCQA	PubMedQA
410	Risk-screened	.764	.042	.642	.486	.144
411	Risk-screened	.528	.010	.420	.318	.430
412	Risk-screened	.750	.042	.640	.462	.148
413	Risk-screened	.706	.012	.592	.354	.142
410	Coverage-only	.768	.030	.644	.488	.154
411	Coverage-only	.460	.010	.240	.242	.526
412	Coverage-only	.746	.038	.646	.468	.164
413	Coverage-only	.738	.004	.602	.414	.192
410	Cloned-update	.770	.034	.644	.484	.140
411	Cloned-update	.632	.044	.406	.370	.482
412	Cloned-update	.740	.034	.646	.458	.144
413	Cloned-update	.716	.002	.596	.358	.214
410	Random	.772	.054	.654	.484	.140
411	Random	.350	.006	.248	.232	.478
412	Random	.740	.028	.640	.452	.156
413	Random	.648	.010	.542	.274	.102

K Robustness and Diagnostic Analyses

Attribution controls. The first six-seed comparison matches candidate-gradient calls across the anchored branches. The cloned-update comparison instead matches before/after candidate-loss evaluations across its selector controls. These controls identify the effect of changing allocation within each scoring protocol. Contrasts are evaluated within their stated scoring protocol. The matching random-control endpoint values reported in Appendix G and the four coverage seeds overlapping the cloned-update study do not add independent seed blocks.

Resource accounting. Every cloned-score selector executes the same 64 candidate-loss evaluations per refresh. Across the eight risk-screened coverage and coverage-only runs, recorded continued-training stage times sum to 9.75 process-hours, excluding final evaluation and orchestration; the maximum per-process CUDA peak allocation is 34,406 MiB. Concurrent jobs, fixed within-block arm order, and asymmetric state serialization confound wall time. These measurements are not total allocated GPU-hours, throughput, latency, energy efficiency, or the minimum cost of random selection without scoring.

Rollout checks. The generation procedure passes the minimum-new-token constraint directly. All 96 inspected rollouts satisfy that constraint and are nonempty. This checks that continuations were produced, but does not establish answer validity or retained task competence. Separate parser-invalid stress-test outputs are examined in Appendix L.

Replay and runtime scope. The replay comparison holds its replay source, target information, active budget, refresh cadence, and evaluation slices fixed across its two arms. Adding the tested anchoring branch worsens all three mean endpoints (Section 8). The reported 17.78% increase in mean step time is descriptive: fixed arm order and co-tenancy prevent attributing it to the added mechanism as an efficiency estimate. Its batching and gradient accumulation differ from those of the selector study, so the studies are not pooled.

Reservoir and target checks. Prior-stage reservoirs persist cumulatively, with 64 prompts after stage 1 and 128 after stage 2. Candidate pools are sampled uniformly without replacement. Targets use greedy stage-start responses or the original-target empty-output fallback and stay fixed for 24 optimizer steps. No exact within-reservoir or cross-stage prompt duplicates are reported; one chemistry prompt is semantically duplicated across index-disjoint training and reservoir slices. Reservoir membership limits which behaviors can be protected, and no human-importance labels or acceptable-forgetting judgments enter selection. Code, truthfulness, and instruction-following benchmarks beyond the named evaluations are not part of the selector comparison.

L Qualitative Output Analysis

We inspected two all-zero stress-test rows to distinguish empty generations or evaluation failures from model-format collapse. These rows are separate from the controlled selector comparison and are excluded from efficacy and cross-backbone claims.

Table 23: Separate format-collapse stress tests. Prefixes are exact decoded text shortened by ellipses. Each setting covers 20 examples from each of five benchmarks, for 100 inspected outputs.

Model setting	Decoded prefix	Output-validity observations
Qwen3-4B OP-SD	`Reference reference reference reference...`	100/100 nonempty outputs; parser-valid rate 0.00; strict accuracy 0.00. Repetition occurs across all five benchmarks.
Qwen2.5-7B anchoring stress test	`ThoughtThought ThoughtThought...`	100/100 nonempty outputs; parser-valid rate 0.00; strict accuracy 0.00. GSM8K generations reach the 96-token cap; the other outputs contain eight repeated tokens.

These cases show why nonempty generation is insufficient as an output-validity check: a decoded string can exist while carrying no parseable answer. They do not show that the present selector causes, prevents, or repairs collapse.

Selector/training-length mismatch. Across the eight coverage-arm runs, 16 of 384 risk-screened coverage selections and 24 of 384 coverage-only selections exceed 1,024 prompt tokens; each arm's candidate pools contain 37 such occurrences among 1,536 candidates. Facility coverage is computed from the untruncated rendered-prompt token set, whereas training uses `max_length=1024`. Thus part of the geometry signal can come from prompt tokens removed by left truncation before the training loss is evaluated. This is a construct-validity limitation; it does not change the observed reported endpoint trade-offs.

The Qwen3-4B and Qwen2.5-7B cases are not the main Qwen2.5-3B OP-SD baseline. They therefore do not explain that baseline's low final held-out accuracy or establish that anchoring prevents collapse in the controlled stream.

M Implications and Broader Impact

An anchored method's gain over its base learner is not automatically evidence for the rule that selected its anchors. In the evaluated stream, gradient-conflict ranking is unnecessary for the anchored-policy advantage; cloned-update ranking has favorable mean effects with mixed joint outcomes; and risk-screened coverage introduces an endpoint trade-off. Evaluating intervention benefit, local score fidelity, and full-stream policy value separately makes these findings distinguishable. The supervised-replay comparison further shows why an additional retention objective should be tested rather than assumed beneficial.

Behavior anchors preserve cached targets—normally the model's stage-start responses, with original dataset targets used for empty-output fallback—rather than enforcing factual correctness or human preferences. Preserving a target can preserve its errors or harms. Deployment would require content review, safety filtering, override and removal mechanisms, privacy review of retained prompts and responses, and assessment of whose behaviors receive protection. The current study does not evaluate those safeguards or define acceptable forgetting. Its empirical scope is the reported model, stream, training configurations, and endpoints.