# REFRESHING LESS, SELECTING BETTER: REUSING STALE GRADIENT FEATURES FOR EFFICIENT INFLUENCE-BASED DATA SELECTION

**Anonymous authors**
Paper under double-blind review

## ABSTRACT

Gradient-based data selection methods such as LESS score each candidate example by the alignment between its gradient and the gradient of a target validation set, and the scoring stage dominates their cost because per-example gradient features must be recomputed whenever the model checkpoint changes. Across three selection seeds, two model families, two candidate pools, and two target tasks, gradient features cached at a post-warmup checkpoint and paired with fresh validation gradients preserve the ranking 40 optimizer steps later with Spearman correlation between 0.952 and 0.991, while the top-10% subset they induce misses 10 to 22% of the examples that full recomputation selects. We therefore propose Cached Diverse Influence Selection (CDIS), which recomputes gradient features for the top-ranked fraction $p$ of candidates under the stale scores, fits an affine calibration on the recomputed examples, and selects the final subset under source and length quotas. A refresh fraction at or above the selection fraction recovers the exact top-$k$ subset whenever the calibrated stale scores have bounded error, and the measured budget curves follow this rule: at $p = 0.3$ the recovered top-$k$ subset coincides with full recomputation in every setting, allocating the same budget per stratum recovers the stratified subset at 0.92 to 1.00, and the wall-clock of the gradient stage drops by a factor of 3.5 to 3.6. Iterating the cache over four checkpoints keeps top-$k$ overlap at 0.98 or higher at 1.9 gradient features per example against 4 for full recomputation, and the agreement between stale and recomputed scores on the refreshed examples provides a free check for unsafe reuse. Downstream, unconstrained top-$k$ selection on influence scores collapses to a single data source and scores 13 points below random selection on GSM8K. CDIS scores 12 points above random selection with paired confidence intervals that exclude zero and trails full recomputation by 4.3 points, one training-run standard deviation, at 3.4 times lower selection cost.

## 1 INTRODUCTION

Post-training of large language models increasingly relies on data selection to reach a target capability within a fixed compute budget (Albalak et al., 2024; Zhou et al., 2023; Liu et al., 2024). Gradient-based selection methods such as LESS (Xia et al., 2024) represent each candidate example by a projected gradient feature at a model checkpoint, represent the target task by validation gradients at the same checkpoint, and select the candidates whose features align with the target (Koh & Liang, 2017; Pruthi et al., 2020; Park et al., 2023), which ties the selected data to the model and to the target distribution.

The cost of these methods is concentrated in the scoring stage. Over a post-training run, the gradient cost of a selection pipeline factorizes into three quantities: the cost of one per-example gradient feature, the number of candidates that receive a feature, and the number of times the features are recomputed. Prior work on efficiency has concentrated on the first factor: random projection reduces the dimension of each feature (Park et al., 2023; Xia et al., 2024), proxy models reduce the cost of computing it (Chen et al., 2026), and learned influence models replace the gradient computation (Yu et al., 2024). The third factor recurs in practice, because LESS computes gradient features at four checkpoints of a warmup run and averages the scores, and iterative pipelines re-select as the model



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.

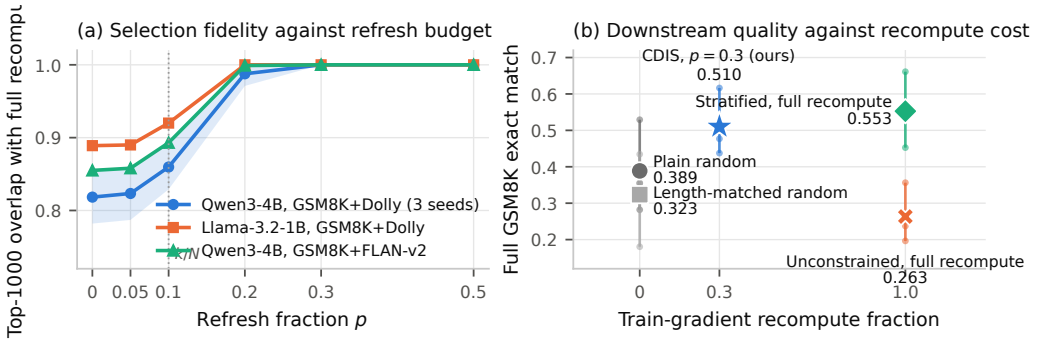



Figure 1: Headline results. (a) Overlap between the top-1,000 subset selected from partially refreshed scores and the subset from full recomputation against the refresh fraction $p$, for Qwen3-4B-Base on three GSM8K +Dolly pools (mean and min–max band), Llama-3.2-1B with its own gradient features, and Qwen3-4B-Base on a GSM8K +FLAN-v2 pool; the dotted line marks $k/N = 0.1$. (b) Full GSM8K exact match against the fraction of train-gradient features recomputed per selection cycle, three-seed means with min–max whiskers.

changes (Mirzasoleiman et al., 2020; Killamsetty et al., 2021). This paper asks how much of that recomputation is necessary.

Our starting point is a measurement of how influence scores change between post-training checkpoints. When train-gradient features cached at one checkpoint are paired with validation gradients from a later checkpoint, the resulting scores preserve the later checkpoint's ranking with Spearman correlation between 0.952 and 0.991 across three selection seeds, two model families, two candidate pools, and two target tasks, while the top-10% subset they induce misses 10 to 22% of the examples that full recomputation selects. This gap is where partial recomputation pays off: the stale ranking identifies where the subset can change, and the remaining candidates keep their cached features.

We turn this observation into a selection procedure that we call Cached Diverse Influence Selection (CDIS). Given cached train-gradient features, fresh validation gradients, and a refresh fraction $p$, CDIS scores every candidate with the cached features, recomputes gradient features for the top $pN$ candidates under the stale scores, fits an affine map from stale to recomputed scores on those candidates, and assigns every remaining candidate its calibrated stale score. The budget has a lower bound with a reason: a refresh fraction at or above the selection fraction $k/N$ recovers the exact top-$k$ subset whenever the calibrated stale scores have bounded error (Proposition 1), and the measured budget curves rise steeply once $p$ passes $k/N$ and reach full fidelity at $p = 0.3$ with 3.5 to 3.6 times lower wall-clock for the gradient stage (Figure 1a). The same argument applied per stratum says how to allocate the budget for a stratified selection rule, and per-stratum allocation recovers the stratified subset at 0.92 to 1.00. The saving compounds when the cache is iterated, and over four checkpoints the iterated cache keeps top-$k$ overlap at 0.98 or higher at 1.9 gradient features per example against 4 for full recomputation. Finally, because the refreshed candidates carry both a stale and a recomputed score, their agreement comes free with the refresh, tracks the fidelity of the cached ranking, and separates unstable early-training intervals from stable ones.

A faithful cache is necessary for good selection, and the selection rule that consumes the scores matters as much. On a candidate pool that mixes 5,000 GSM8K and 5,000 Dolly examples under a GSM8K target, unconstrained top-$k$ selection on influence scores picks 1,000 GSM8K examples and zero Dolly examples in every seed and scores 13 points below random selection on the full GSM8K test set, because the selected subset collapses in source and length coverage. CDIS therefore applies the calibrated scores through a stratified selection rule with quotas over data sources and prompt-length buckets, and with this rule it scores 12.2 points above plain random selection and 24.7 points above unconstrained selection on full GSM8K with paired confidence intervals that exclude zero, and 4.3 points, one training-run standard deviation, below stratified selection with full recomputation (Figure 1b).

Our contributions are the following.

- We measure the temporal stability of influence-based train-gradient features across post-training checkpoints on three seeds, two model families, two candidate pools, two target tasks, and checkpoint distances from 40 to 240 optimizer steps.
- We propose targeted partial refresh with affine calibration, give a sufficient condition on the refresh fraction for exact recovery of the top-$k$ subset, derive from it a rule for $p$ and a per-stratum allocation of the budget, and provide an online check for unsafe reuse that comes free with the refresh. At $p = 0.3$ the method recovers the full-recomputation subset in every setting and outperforms random refresh and checkpoint skipping.
- We show that cached influence scores require a diversity-preserving selection rule to be useful downstream, quantify the cost of caching against training-run variance, and show that the refresh composes with quota-based, normalization-based, and label-free selection rules.

## 2 Related work

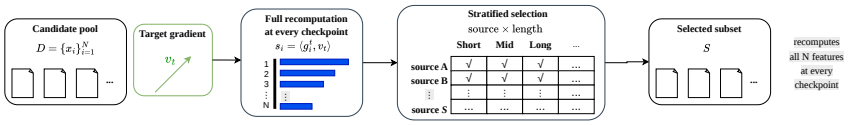
**Influence estimation and gradient-based selection.** Influence functions (Koh & Liang, 2017) and their scalable approximations (Grosse et al., 2023; Kwon et al., 2024; Choe et al., 2025) estimate the effect of a training example on a model prediction; TracIn (Pruthi et al., 2020) accumulates gradient inner products over checkpoints, TRAK (Park et al., 2023) and datamodels (Ilyas et al., 2022) attribute predictions at scale through random projection, and DsDm (Engstrom et al., 2024) applies datamodels to pretraining data selection. LESS (Xia et al., 2024) adapts this line to instruction tuning with Adam-aware projected gradients stored in a datastore and computed at several warmup checkpoints; CDIS operates on that datastore and changes when its features are recomputed.

**Reducing the cost of gradient-based selection.** Existing work lowers the cost of a single scoring pass: random projection (Park et al., 2023; Xia et al., 2024) reduces the dimension of each gradient feature, IProX (Chen et al., 2026) constructs proxy models whose gradients preserve influence at lower cost, MATES (Yu et al., 2024) trains a small data influence model that is retrained periodically as the pretraining model changes, and GREATS (Wang et al., 2024) moves selection into the training loop with ghost inner products. CDIS reduces the number of features that a checkpoint-aware pipeline recomputes, a separate factor of the total cost, and the two factors multiply: proxy or projected gradients can be cached and refreshed with the same procedure. The setting in which this matters most is repeated selection, as in gradient-matching coresets that re-solve selection every epoch (Mirzasoleiman et al., 2020; Killamsetty et al., 2021). Wang et al. (2025) show that influence depends on the training stage; we measure how much post-training rankings drift between nearby checkpoints and exploit the concentration of that drift at the top of the ranking.

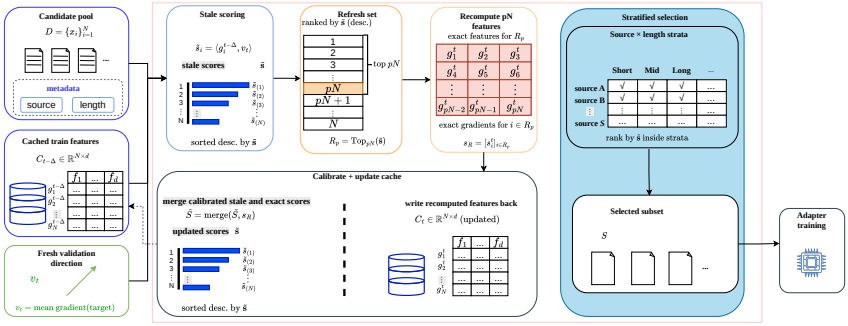
**Diversity in instruction-tuning data selection.** Quality-ranked or influence-ranked selection tends to produce unbalanced subsets, and several methods enforce balance or diversity: BIDS normalizes influence within task groups (Dai et al., 2025), G-DIG and TAGCOS cluster gradient features (Pan et al., 2024; Zhang et al., 2025b), QDIT and D3 combine quality with diversity objectives (Bukharin et al., 2024; Zhang et al., 2025a), and DEITA and InsTag use complexity and tag diversity (Liu et al., 2024; Lu et al., 2024); Xia et al. (2025) report that many selection methods score below random selection at scale, which our unconstrained baseline reproduces; our contribution on this side is the interaction with caching, since a faithful cache inherits the collapse of full recomputation.

## 3 Problem setup

**Gradient-based selection.** Let $\mathcal{D} = \{x_i\}_{i=1}^{N}$ be a candidate pool, $\mathcal{V}$ a validation set for the target task, and $\theta_t$ a model checkpoint. A LESS-style selector represents each candidate by a projected, optimizer-aware gradient feature $g_i^t \in \mathbb{R}^d$ computed at $\theta_t$, reduces the validation gradients at $\theta_t$ to a target direction $v^t$, and scores each candidate by $s_i^t = \langle g_i^t, v^t \rangle$. LESS computes these scores at several checkpoints $t_1, \ldots, t_T$ of a warmup run and averages them with learning-rate weights, $S_i = \sum_j w_j \, s_i^{t_j}$, before selecting the top-$k$ candidates (Xia et al., 2024). Computing $g_i^t$ for all $N$ candidates is the expensive step; $v^t$ comes from a small validation set and is cheap to refresh at every checkpoint.




(a) Prior balanced influence selection scores every candidate at full cost at each checkpoint and then applies a stratified selection rule.




(b) CDIS scores the pool with cached features $C_{t-\Delta}$ and the fresh target gradient $v_t$, recomputes exact scores for the top-$pN$ refresh set $\mathcal{R}_p$, calibrates the remaining stale scores, writes the recomputed features back to the cache, and applies the same stratified selection rule to the updated scores $\hat{s}$.

Figure 2: From prior balanced influence selection to CDIS. The selection rule on the right of each panel is shared; the refresh layer in (b) reduces the number of train-gradient features recomputed per checkpoint, and the stale and recomputed scores on $\mathcal{R}_p$ provide the online check of Section 4.3.

**Cost model.** A pipeline that scores at $T$ checkpoints computes $TN$ train-gradient features at a cost of $c$ each; if the features from the first checkpoint are cached and a fraction $p$ of them is recomputed at each later checkpoint, the cost becomes $Nc + (T-1)pNc$ and the saving approaches a factor of $1/p$ as $T$ grows. Projection and proxy models reduce $c$, and our work reduces $p$.

**Stale reuse and fidelity.** Let $\ell < r$ be two checkpoints, with features $g_i^\ell$ cached at $\ell$ and validation gradients $v^r$ available at $r$. Full recomputation scores candidates by $s_i^r = \langle g_i^r, v^r \rangle$, and stale reuse scores them by

$$\tilde{s}_i = \langle g_i^\ell, v^r \rangle, \tag{1}$$

and we measure how well $\tilde{s}$ preserves $s^r$ with the Spearman rank correlation over the pool and with the overlap $|\mathcal{S} \cap \mathcal{S}_{\text{full}}|/|\mathcal{S}_{\text{full}}|$ between the subset $\mathcal{S}$ that a selection rule produces from $\tilde{s}$ and the subset $\mathcal{S}_{\text{full}}$ that the same rule produces from $s^r$.

## 4 Method

CDIS has two parts: a targeted partial refresh (Sections 4.1 to 4.3) turns cached features and a refresh budget into a calibrated score vector, and a stratified selection rule (Section 4.4) turns the score vector into a subset. Figure 2 contrasts the pipeline with prior balanced influence selection, and Algorithm 1 in Appendix B gives the full procedure.

### 4.1 Targeted partial refresh

Given cached features $G^\ell = \{g_i^\ell\}_{i=1}^N$, the current validation direction $v^r$, and a refresh fraction $p \in (0, 1]$, the refresh proceeds in four steps. First, it computes the stale score $\tilde{s}_i = \langle g_i^\ell, v^r \rangle$ for every candidate. Second, it forms the refresh set $\mathcal{R}_p = \text{TopK}_{\lceil pN \rceil}([N]; \tilde{s})$ from the $pN$ candidates with the highest stale scores. Third, it recomputes $g_i^r$ and the exact score $s_i^r = \langle g_i^r, v^r \rangle$ for every $i \in \mathcal{R}_p$ and writes $g_i^r$ back to the cache. Fourth, it fits an affine calibration

$$(a, b) = \arg\min_{a,b} \sum_{i \in \mathcal{R}_p} \left( a\tilde{s}_i + b - s_i^r \right)^2 \tag{2}$$

on the refreshed candidates and returns the hybrid scores $\hat{s}_i = s_i^r$ for $i \in \mathcal{R}_p$ and $\hat{s}_i = a\tilde{s}_i + b$ otherwise. The refresh set is chosen by stale rank because the candidates that can enter or leave the selected subset after recomputation are those whose stale scores lie near the selection threshold, which for a top-$k$ rule with $k \leq pN$ sit at the top of the stale ranking. The calibration places recomputed and stale scores on a common scale, which matters whenever a selection rule compares candidates inside and outside $\mathcal{R}_p$, and the next checkpoint repeats the procedure on the updated cache.

### 4.2 A SUFFICIENT REFRESH BUDGET

**Proposition 1** (Sufficient refresh set)**.** *Let $s^r$ be the recomputed scores, assumed distinct, let $\mathcal{T}_k$ be the top-$k$ subset under $s^r$, and let $\tau = \min_{i \in \mathcal{T}_k} s_i^r$. Let $c(\tilde{s}_i) = a\tilde{s}_i + b$ with $a > 0$ be the calibrated stale scores and let $\varepsilon = \max_i |s_i^r - c(\tilde{s}_i)|$ be their maximum error. If the refresh set satisfies $\mathcal{R} \supseteq \{i : c(\tilde{s}_i) \geq \tau - \varepsilon\}$, then the top-$k$ subset under the hybrid scores $\hat{s}$ equals $\mathcal{T}_k$.*

The proof is in Appendix A. Because $\mathcal{T}_k$ is contained in the set $\{i : c(\tilde{s}_i) \geq \tau - \varepsilon\}$, that set has at least $k$ elements, so $p \geq k/N$ is necessary for the condition, and the number of additional candidates it requires grows with the error $\varepsilon$ and with the density of stale scores just below the threshold. This gives a rule for $p$ before any downstream training: the selection fraction plus a margin for the drift, widened for long or early intervals. The error $\varepsilon$ depends on the whole pool, and the refresh set exposes the residuals on its members, which the check of Section 4.3 reads. For a stratified rule the same argument applies within each stratum, with the stratum's own threshold and quota, so exact recovery requires the refresh set to contain the union of the per-stratum margin sets. A single refresh set chosen by global stale rank covers the strata whose scores are high under the target and can leave strata with low scores, such as general instruction data under a mathematical target, to their calibrated stale scores. We therefore consider two allocations of the same budget $pN$: global allocation refreshes the top-$pN$ candidates of the pool, and per-stratum allocation refreshes the top-$p$ fraction of every stratum, which covers each stratum's margin whenever $p$ exceeds the stratum's own selection fraction. Section 6.1 confirms the prediction: global allocation recovers the top-$k$ subset exactly and the stratified subset at 77 to 94%, with the missing examples in the lower-scoring source, and per-stratum allocation recovers the stratified subset at 92 to 100%.

### 4.3 AN ONLINE CHECK FOR UNSAFE REUSE

The candidates in $\mathcal{R}_p$ carry both a stale and a recomputed score, so their agreement is observable at refresh time as a by-product of the refresh. We use the Spearman correlation between $\tilde{s}$ and $s^r$ on $\mathcal{R}_p$ as a check: low agreement indicates that the cached checkpoint lies in an unstable region of training or too far from the current checkpoint, and a practitioner then increases $p$ or recomputes all features. Section 6.2 validates this statistic on every measured interval.

### 4.4 STRATIFIED SELECTION

The final subset is selected from $\hat{s}$ under quotas. Source quotas split $k$ across data sources. Within each source, candidates are split into four equal-count buckets by prompt length, and the source quota is divided evenly across buckets with largest-remainder rounding. Within each stratum the highest-scoring candidates fill the quota, and underfilled strata are topped up from the same source and then from the whole pool by score. The main experiments use two sources with equal quotas, the uniform allocation over sources (Section 6.4). When source labels are unavailable, we cluster the cached gradient features with $k$-means and allocate quotas in proportion to cluster size.

## 5 EXPERIMENTAL SETUP

**Models, warmup, and pools.** The main model is Qwen3-4B-Base (Yang et al., 2025); Llama-3.2-1B (Grattafiori et al., 2024; Meta, 2024) serves as a second model family with its own gradient features, warmup, and selections. Following LESS, each model is warmed up with LoRA (Hu et al., 2022) on a random 7% of the Dolly instruction data (Conover et al., 2023) for one epoch of 132 optimizer steps, with checkpoints every 40 steps. Cached features come from checkpoint 80 and the selection checkpoint is 120; the early intervals 40→80 and 40→120 serve as unstable comparisons, and a three-epoch warmup with the same data and spacing provides checkpoints 160, 240, and 320 for

| | | Top-1,000 overlap | | | Stratified, global allocation | | | Stratified, per-stratum | |
|---|---|---|---|---|---|---|---|---|---|
| Setting | Spearman | stale | $p$=0.2 | $p$=0.3 | stale | $p$=0.2 | $p$=0.3 | $p$=0.2 | $p$=0.3 |
| Qwen3-4B, seed 0 | 0.987 | 0.856 | 0.999 | 1.000 | 0.824 | 0.920 | 0.927 | 0.986 | 0.999 |
| Qwen3-4B, seed 1 | 0.976 | 0.782 | 0.971 | 1.000 | 0.720 | 0.837 | 0.857 | 0.955 | 0.996 |
| Qwen3-4B, seed 2 | 0.988 | 0.817 | 0.993 | 1.000 | 0.830 | 0.935 | 0.940 | 0.986 | 0.998 |
| Llama-3.2-1B, seed 0 | 0.991 | 0.889 | 1.000 | 1.000 | 0.829 | 0.885 | 0.904 | 0.978 | 0.992 |
| Qwen3-4B, GSM8K +FLAN-v2 pool | 0.989 | 0.855 | 0.999 | 1.000 | 0.833 | 0.929 | 0.936 | 0.989 | 1.000 |
| Qwen3-4B, seed 0, MMLU target | 0.969 | 0.872 | 0.996 | 1.000 | 0.733 | 0.819 | 0.819 | 0.934 | 0.977 |
| Qwen3-4B, seed 1, MMLU target | 0.952 | 0.896 | 0.996 | 1.000 | 0.698 | 0.765 | 0.765 | 0.864 | 0.919 |

Table 1: Selection fidelity of stale reuse and targeted refresh (checkpoint 80 to 120; GSM8K target on GSM8K +Dolly pools unless noted). Overlap is on the top-10% subset under unconstrained top-$k$ and under the stratified rule; global allocation refreshes the top-$pN$ candidates of the pool and per-stratum allocation the top-$p$ fraction of every stratum, at the same budget.

the distance and repeated-selection studies. The main pool contains 5,000 GSM8K training examples (Cobbe et al., 2021) and 5,000 Dolly examples, sampled per selection seed (seeds 0, 1, and 2); a second pool replaces Dolly by 5,000 FLAN-v2 examples (Longpre et al., 2023). The main target is GSM8K, with the validation direction from the 100 GSM8K validation examples of the LESS pipeline; the MMLU target (Hendrycks et al., 2021) uses the 285 MMLU validation examples of LESS with its per-subtask reduction. Selection takes the top 10%, that is, $k = 1{,}000$ examples.

**Features, training, and evaluation.** Train and validation features are Adam-aware projected gradients from the LESS implementation with $d = 8{,}192$, collected at checkpoints 40, 80, and 120. Each selected subset trains a LoRA adapter for five epochs with the warmup hyperparameters (Appendix C), and the same adapters are evaluated with the LESS 8-shot evaluator on the full GSM8K test set (1,319 items) and on the 2,232 numeric-answer items of ASDiv (Miao et al., 2020), with exact match after normalization as the metric. Absolute accuracies under this protocol sit below few-shot results reported for the base model, since adapters train on chat-formatted examples and are scored by exact match within 256 generated tokens, and every method shares it. Selection-rule ablations use a 200-item GSM8K subset (GSM200). All methods answer the same test items, so we report paired, seed-stratified bootstrap confidence intervals over per-item correctness differences (10,000 resamples), and we retrain four fixed selections with three training seeds each to measure the within-selection standard deviation.

**Baselines and comparisons.** Downstream baselines are unconstrained top-$k$ with full recomputation, plain random selection, length-matched random selection (matching the prompt-length profile of the unconstrained subset), and stratified selection with full recomputation. Fidelity comparisons at equal or larger savings are random refresh (a random 30% of the pool) and checkpoint skipping (features from a checkpoint twice as far back, with the refresh skipped). BIDS-style normalization (Dai et al., 2025) and a subset-transfer diagnostic on Mistral-7B-v0.3 (Jiang et al., 2023; Mistral AI, 2024) complete the comparisons (Appendix O), and wall-clock is measured on one NVIDIA A6000 (48 GB).

## 6 Results

### 6.1 Stale reuse and targeted refresh

Table 1 reports how much of the checkpoint-120 selection survives with checkpoint-80 features and fresh validation gradients. The global ranking survives almost intact, with Spearman correlations between 0.952 and 0.991 across all seven settings, and the selected subset shifts: stale scores recover 0.78 to 0.90 of the top-1,000 subset and 0.70 to 0.83 of the stratified subset. This is the situation that Proposition 1 anticipates, because the drift is small relative to the score range and still large enough near the selection threshold to move up to 30% of the selected examples.

**Targeted refresh.** Targeted refresh closes this gap. Recomputing features for the top 20% of candidates under the stale ranking raises top-1,000 overlap to at least 0.971 in every setting, and $p = 0.3$ recovers the subset exactly in all seven. Figure 1(a) shows why: the curve rises steeply once

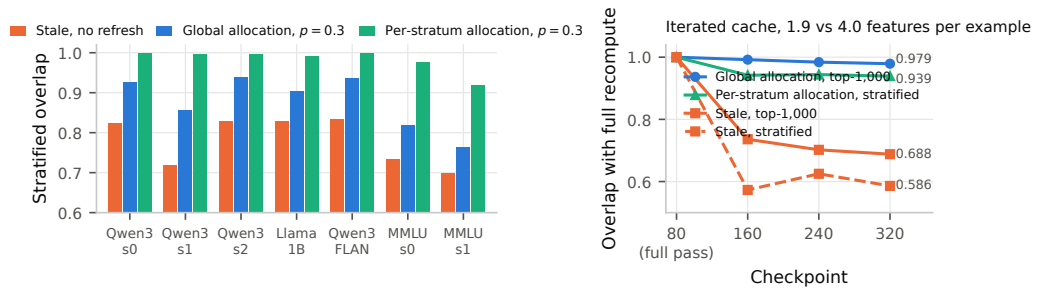



Figure 3: (a) Stratified overlap with full recomputation for stale reuse, global allocation, and per-stratum allocation at $p = 0.3$ on the seven settings of Table 1. (b) Repeated selection with an iterated cache on the three-epoch run (Qwen3-4B seed 0): a full pass at checkpoint 80, then $0.3N$ features refreshed at each later checkpoint, at 1.9 features per example against 4.0 (Appendix L).

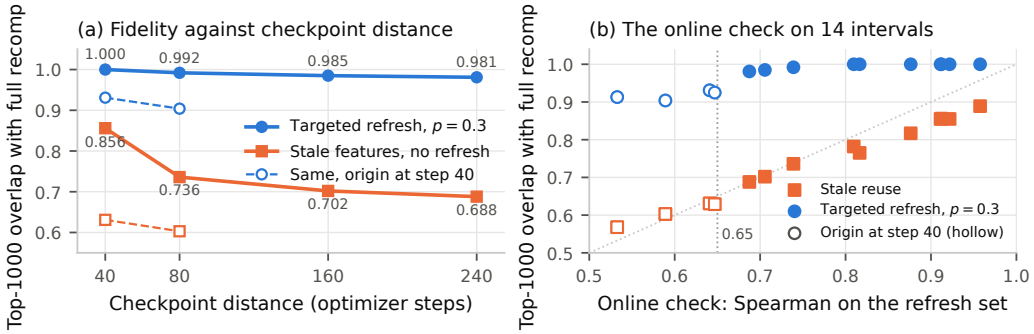



Figure 4: (a) Fidelity against checkpoint distance on Qwen3-4B seed 0, single refresh from the origin; filled markers start at checkpoint 80 and hollow markers at checkpoint 40. (b) The online check (Spearman correlation between stale and recomputed scores on the $p = 0.3$ refresh set) against the top-1,000 overlap of stale reuse and of targeted refresh on all fourteen single-refresh intervals (Appendix K).

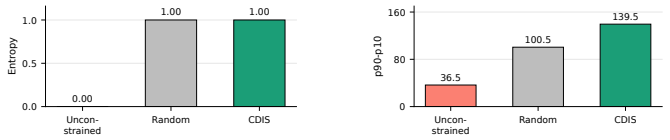
$p$ passes the selection fraction $k/N = 0.1$, as Proposition 1 predicts, and saturates by $p = 0.3$. We therefore set $p = 0.3$ for a top-10% selection, the selection fraction plus a margin of 0.2 for the drift.

**Refresh allocation.** The stratified subset recovers less than the top-$k$ subset, as Section 4.2 predicts: a global refresh set concentrates on the source that scores highest under the target, so the other source keeps its calibrated stale scores. Figure 3(a) confirms this in both directions: global allocation misses Dolly examples under the GSM8K target and GSM8K examples under MMLU. Allocating the same budget per stratum removes the shortfall and recovers the stratified subset at 0.992 to 1.000 under GSM8K and at 0.977 and 0.919 under MMLU.

**Alternatives and calibration.** Targeted refresh outperforms two simpler ways to spend less on gradients: reusing checkpoint-40 features at checkpoint 120, which halves the number of scoring passes, recovers 0.57 to 0.77 of the top-1,000 subset, and refreshing a random 30% of the pool recovers 0.89 to 0.92 at the same budget, with lower stratified overlap and downstream accuracy (Appendix H). The affine calibration changes post-warmup overlap at $p \geq 0.2$ by at most 0.001 and adds 1.9 to 7.4 points at $p = 0.1$ and on early or long intervals (Appendix J).

### 6.2 Checkpoint distance, repeated selection, and the online check

**Distance, origin, and repeated selection.** Figure 4(a) extends the single-refresh measurement to longer intervals and to an earlier origin. From checkpoint 80, stale fidelity decays from 0.856 at distance 40 to 0.688 at distance 240 while targeted refresh at $p = 0.3$ stays between 1.000 and 0.981, and an origin at checkpoint 40 lowers both to 0.603 and 0.904 at distance 80, against 0.736 and 0.992




(a) Source entropy of the selected subset.

(b) Prompt-length coverage (p10 to p90).

Figure 5: Unconstrained top-$k$ selection on influence scores collapses to a single source and a narrow length window; CDIS keeps source balance and a wide length range (per-seed profiles in Appendix F).

| | | Full GSM8K | | Full ASDiv (transfer) | |
|---|---|---|---|---|---|
| Method | Recompute | EM | CDIS − method [95% CI] | EM | CDIS − method [95% CI] |
| Unconstrained top-$k$, full recompute | $N$ | 0.263 | +24.7 [+23.1, +26.3] | 0.163 | +15.6 [+14.4, +16.7] |
| Plain random | 0 | 0.389 | +12.2 [+10.6, +13.7] | 0.248 | +7.1 [+6.0, +8.1] |
| Length-matched random | 0 | 0.323 | +18.7 [+17.0, +20.3] | 0.170 | +14.9 [+13.7, +16.0] |
| Stratified, full recompute | $N$ | 0.553 | −4.3 [−5.5, −3.0] | 0.339 | −2.0 [−2.9, −1.1] |
| CDIS, $p = 0.3$, global allocation | $0.3N$ | 0.510 | — | 0.319 | — |

Table 2: Downstream quality on Qwen3-4B-Base (top-10% of the 10k GSM8K +Dolly pool, five-epoch LoRA, same adapters on full GSM8K and full ASDiv). EM is the three-seed mean (per-seed values in Appendix E); differences are percentage points with paired, seed-stratified bootstrap 95% CIs over per-item correctness (10,000 resamples).

from checkpoint 80. The saving in the cost model of Section 3 accrues when the cache is iterated over the checkpoints that LESS scores, so Figure 3(b) runs one full pass at checkpoint 80 followed by three consecutive refreshes at checkpoints 160, 240, and 320 of the warmup run, at a cumulative cost of 1.9 features per example against 4.0. Fidelity holds across the chain: global allocation recovers the top-1,000 subset at 0.979 or higher at every checkpoint and the checkpoint-averaged LESS score at 0.999, per-stratum allocation recovers the stratified subset at 0.939 or higher and at 0.990 for the aggregate, and stale features fall to 0.69 to 0.74. Chains that start at checkpoint 40, inside the unstable early interval, reach 0.90 to 1.00 top-1,000 and 0.62 to 0.84 stratified overlap at checkpoint 120 (Appendix L, with a per-cohort calibration for the mixed-age cache).

**The online check.** Because the refresh computes exact scores for the refresh set anyway, the agreement between stale and recomputed scores on that set is available at every refresh, and Figure 4(b) shows that it predicts fidelity. Across all fourteen single-refresh intervals it tracks the top-1,000 overlap of stale reuse to within 0.07, and the global rank correlation stays near 0.91 across the same intervals (Appendix K). The four intervals on which $p = 0.3$ recovers less than 0.95 of the subset are exactly those with agreement below 0.65, every interval above 0.68 recovers at least 0.981, and in the chains the check flags the first step of the early-origin Qwen3 chains at 0.64. At the long single-refresh distances, where the agreement drops to 0.69 to 0.74 while targeted refresh still recovers 0.98, a threshold of 0.75 triggers extra gradient computation and keeps the selection intact.

### 6.3 Downstream quality and training-run variance

**Unconstrained influence selection collapses.** A faithful cache reproduces full recomputation, and full recomputation itself selects poorly when its scores are consumed without constraints. Top-$k$ selection on raw influence scores picks 1,000 GSM8K examples and zero Dolly examples in all three seeds, concentrated in a 30 to 66 token prompt-length window (Figure 5), with the lowest training loss and the lowest test accuracy of every method we test: 0.263 on full GSM8K against 0.389 for plain random (Table 2). Because the collapse appears at full recomputation and the refresh preserves it exactly, the score signal itself produces it, and since length-matched random selection also falls below plain random in the three-seed mean, length concentration accounts for part of the loss and source concentration for the rest.

**Stratified selection recovers a large margin, and the cache keeps most of it.** Once the same scores pass through the stratified rule, the picture reverses. Stratified selection with full recomputation

reaches 0.553 on full GSM8K and 0.339 on ASDiv, 16.4 [14.8, 18.0] and 9.0 [8.0, 10.1] points above plain random, and CDIS with $p = 0.3$ and global allocation follows at 0.510 and 0.319, which is 12.2 and 7.1 points above plain random and 24.7 and 15.6 above unconstrained selection. Both stratified selectors exceed both random controls on every seed and both test sets, and unconstrained selection stays below plain random on every seed on ASDiv and on two of three seeds on GSM8K, within 0.001 on the third (Appendix E). The difference to full recomputation is 4.3 [3.0, 5.5] points on GSM8K and 2.0 [1.1, 2.9] on ASDiv, from selections that agree with the reference subset in 0.86 to 0.94 of their examples.

**Training-run variance.** The paired intervals cover item sampling, so we put that difference on the scale of training noise by freezing four selections and retraining each with three training seeds (Appendix G). Retraining an identical selection moves full GSM8K exact match by up to 8.9 points, with a pooled within-selection standard deviation of 4.25 points; CDIS stays below full recomputation in every retraining pair, by 4.9 and 7.5 points in the retrain means, so its cost amounts to one to two standard deviations while its margin over random selection amounts to three. CDIS therefore lowers selection-cycle cost by 3.37× at a cost of one to two training-run standard deviations on the target task and two points on transfer, and keeps a margin over random selection several times the noise floor.

### 6.4 Selection rules

The stratified rule needs both of its constraints. On the seed-0 GSM200 probe (Appendix M; sampling standard error 3.5 points), length quotas alone still select 1,000 GSM8K examples and reach 0.100, source quotas alone balance the sources and reach 0.410, and the combination reaches 0.655, with CDIS at $p = 0.3$ tracking each variant. The equal source split is also the best one, because raising the GSM8K share to 70% and 80% lowers GSM200 accuracy to 0.400 and 0.350, differences far beyond the probe's sampling error, and BIDS-style normalization, the closest balanced-influence rule, is preserved as well (0.615 with full recomputation, 0.625 at $p = 0.3$).

Source labels can come from the cache itself. Clustering the cached checkpoint-80 features with $k$-means ($k = 8$) recovers the GSM8K/Dolly split with 99.8% purity, quotas proportional to cluster size select 517 GSM8K and 483 Dolly examples, and the refresh preserves this rule at 0.924 stratified overlap, with every differing example a Dolly-side swap as Section 4.2 predicts. Downstream on seed 0, the label-free rule reaches 0.579 on full GSM8K with full recomputation, above plain random at 0.529, and 0.468 with $p = 0.3$, a difference inside the training-run range of Appendix G.

### 6.5 Cost accounting

The train-gradient stage drops from 3,935.75 to 1,087.33 seconds on Qwen3-4B-Base (3.62×) and from 1,579.72 to 448.60 seconds on Llama-3.2-1B (3.52×), and since validation-gradient extraction (106.70 s) and CPU scoring, calibration, and selection (6.83 s) are shared, the selection cycle drops from 4,049.28 to 1,200.86 seconds (3.37×) while downstream LoRA training is unchanged (Appendix N). The factors exceed 3.33× because the refreshed candidates under a GSM8K target are shorter than the pool average (Appendix F). Because the cache costs one full pass, the saving begins with the second selection cycle, at 1.54× for two cycles, 2.29× for five, and 3.37× in the limit, and the four-checkpoint chain of Figure 3(b) realizes 2.1× on the train-gradient stage.

## 7 Conclusion

Influence rankings in LESS-style selection drift slowly between post-training checkpoints, and the drift concentrates at the top of the ranking, where a targeted refresh absorbs it: a refresh fraction above the selection fraction recovers the recomputed subset exactly on three seeds, two model families, two pools, and two targets at 3.5 to 3.6 times lower gradient wall-clock, and sustains 98% fidelity over three consecutive refreshes at half the gradient cost. Per-stratum allocation carries Proposition 1 to stratified selection, and with that rule CDIS keeps the downstream quality of full recomputation within one to two training-run standard deviations at a fraction of its cost; downstream training uses the global allocation, and the per-stratum selections and the MMLU target are measured at the selection level.

## REFERENCES


Alon Albalak, Yanai Elazar, Sang Michael Xie, Shayne Longpre, Nathan Lambert, Xinyi Wang, Niklas Muennighoff, Bairu Hou, Liangming Pan, Haewon Jeong, Colin Raffel, Shiyu Chang, Tatsunori Hashimoto, and William Yang Wang. A survey on data selection for language models. *Transactions on Machine Learning Research*, 2024. URL https://dblp.org/rec/journals/tmlr/AlbalakEXLL0MHP24.

Alexander Bukharin, Shiyang Li, Zhengyang Wang, Jingfeng Yang, Bing Yin, Xian Li, Chao Zhang, Tuo Zhao, and Haoming Jiang. Data diversity matters for robust instruction tuning. In *Findings of the Association for Computational Linguistics: EMNLP 2024*, pp. 3411–3425, Miami, Florida, USA, November 2024. Association for Computational Linguistics. doi: 10.18653/v1/2024.findings-emnlp.195. URL https://aclanthology.org/2024.findings-emnlp.195/.

Sirui Chen, Yunzhe Qi, Mengting Ai, Yifan Sun, Ruizhong Qiu, Jiaru Zou, and Jingrui He. Influence-preserving proxies for gradient-based data selection in LLM fine-tuning. In *The Fourteenth International Conference on Learning Representations*, 2026. URL https://openreview.net/forum?id=PDNpRLxDlI.

Sang Keun Choe, Hwijeen Ahn, Juhan Bae, Kewen Zhao, Youngseog Chung, Adithya Pratapa, Willie Neiswanger, Emma Strubell, Teruko Mitamura, Jeff G. Schneider, Eduard H. Hovy, Roger Baker Grosse, and Eric P. Xing. What is your data worth to GPT? LLM-scale data valuation with influence functions. In *Advances in Neural Information Processing Systems*, volume 38, pp. 145944–145985. Curran Associates, Inc., 2025. doi: 10.52202/085713-4883. URL https://proceedings.neurips.cc/paper_files/paper/2025/hash/d6d26053b977f8c589669fd201615119-Abstract-Conference.html.

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. doi: 10.48550/arXiv.2110.14168. URL https://arxiv.org/abs/2110.14168.

Mike Conover, Matt Hayes, Ankit Mathur, Jianwei Xie, Jun Wan, Sam Shah, Ali Ghodsi, Patrick Wendell, Matei Zaharia, and Reynold Xin. Free dolly: Introducing the world's first truly open instruction-tuned LLM, April 2023. URL https://www.databricks.com/blog/2023/04/12/dolly-first-open-commercially-viable-instruction-tuned-llm. Databricks blog post.

Qirun Dai, Dylan Zhang, Jiaqi W. Ma, and Hao Peng. Improving influence-based instruction tuning data selection for balanced learning of diverse capabilities. In *Findings of the Association for Computational Linguistics: EMNLP 2025*, pp. 7079–7102, Suzhou, China, November 2025. Association for Computational Linguistics. doi: 10.18653/v1/2025.findings-emnlp.373. URL https://aclanthology.org/2025.findings-emnlp.373/.

Logan Engstrom, Axel Feldmann, and Aleksander Madry. DsDm: Model-aware dataset selection with datamodels. In *Proceedings of the 41st International Conference on Machine Learning*, Proceedings of Machine Learning Research. PMLR, 2024. URL https://proceedings.mlr.press/v235/engstrom24a.html.

Aaron Grattafiori et al. The Llama 3 herd of models. *arXiv preprint arXiv:2407.21783*, 2024. URL https://arxiv.org/abs/2407.21783.

Roger Grosse, Juhan Bae, Cem Anil, Nelson Elhage, Alex Tamkin, Amirhossein Tajdini, Benoit Steiner, Dustin Li, Esin Durmus, Ethan Perez, Evan Hubinger, Kamilė Lukošiūtė, Karina Nguyen, Nicholas Joseph, Sam McCandlish, Jared Kaplan, and Samuel R. Bowman. Studying large language model generalization with influence functions. *arXiv preprint arXiv:2308.03296*, 2023. URL https://arxiv.org/abs/2308.03296.

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://dblp.org/rec/conf/iclr/HendrycksBBZMSS21.

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.

Andrew Ilyas, Sung Min Park, Logan Engstrom, Guillaume Leclerc, and Aleksander Madry. Datamodels: Understanding predictions with data and data with predictions. In *Proceedings of the 39th International Conference on Machine Learning*, volume 162 of *Proceedings of Machine Learning Research*, pp. 9525–9587. PMLR, 2022. URL https://proceedings.mlr.press/v162/ilyas22a.html.

Albert Q. Jiang, Alexandre Sablayrolles, Arthur Mensch, Chris Bamford, Devendra Singh Chaplot, Diego de las Casas, Florian Bressand, Gianna Lengyel, Guillaume Lample, Lucile Saulnier, Lélio Renard Lavaud, Marie-Anne Lachaux, Pierre Stock, Teven Le Scao, Thibaut Lavril, Thomas Wang, Timothée Lacroix, and William El Sayed. Mistral 7b. *arXiv preprint arXiv:2310.06825*, 2023. URL https://arxiv.org/abs/2310.06825.

Krishnateja Killamsetty, Durga Sivasubramanian, Ganesh Ramakrishnan, Abir De, and Rishabh Iyer. GRAD-MATCH: Gradient matching based data subset selection for efficient deep model training. In *Proceedings of the 38th International Conference on Machine Learning*, Proceedings of Machine Learning Research. PMLR, 2021. URL https://proceedings.mlr.press/v139/killamsetty21a.html.

Pang Wei Koh and Percy Liang. Understanding black-box predictions via influence functions. In *Proceedings of the 34th International Conference on Machine Learning*, volume 70 of *Proceedings of Machine Learning Research*, pp. 1885–1894. PMLR, 06–11 Aug 2017. URL https://proceedings.mlr.press/v70/koh17a.html.

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

Wei Liu, Weihao Zeng, Keqing He, Yong Jiang, and Junxian He. What makes good data for alignment? a comprehensive study of automatic data selection in instruction tuning. In *The Twelfth International Conference on Learning Representations*, 2024. URL https://openreview.net/forum?id=BTKAeLqLMw.

Shayne Longpre, Le Hou, Tu Vu, Albert Webson, Hyung Won Chung, Yi Tay, Denny Zhou, Quoc V. Le, Barret Zoph, Jason Wei, and Adam Roberts. The Flan collection: Designing data and methods for effective instruction tuning. In *Proceedings of the 40th International Conference on Machine Learning*, Proceedings of Machine Learning Research. PMLR, 2023. URL https://dblp.org/rec/conf/icml/LongpreHVWCTZLZ23.

Keming Lu, Hongyi Yuan, Zheng Yuan, Runji Lin, Junyang Lin, Chuanqi Tan, Chang Zhou, and Jingren Zhou. #InsTag: Instruction tagging for analyzing supervised fine-tuning of large language models. In *The Twelfth International Conference on Learning Representations*, 2024. URL https://openreview.net/forum?id=pszewhybU9.

Meta. Llama-3.2-1B model card, 2024. URL https://huggingface.co/meta-llama/Llama-3.2-1B. Official model card; accessed September 25, 2026.

Shen-yun Miao, Chao-Chun Liang, and Keh-Yih Su. A diverse corpus for evaluating and developing english math word problem solvers. In *Proceedings of the 58th Annual Meeting of the Association for Computational Linguistics*, pp. 975–984, Online, 2020. Association for Computational Linguistics. doi: 10.18653/v1/2020.acl-main.92. URL https://aclanthology.org/2020.acl-main.92/.

Baharan Mirzasoleiman, Jeff Bilmes, and Jure Leskovec. Coresets for data-efficient training of machine learning models. In *Proceedings of the 37th International Conference on Machine Learning*, Proceedings of Machine Learning Research. PMLR, 2020. URL https://dblp.org/rec/conf/icml/MirzasoleimanBL20.

Mistral AI. Mistral-7B-v0.3 model card, 2024. URL https://huggingface.co/mistralai/Mistral-7B-v0.3. Official model card; accessed September 25, 2026.

Xingyuan Pan, Luyang Huang, Liyan Kang, Zhicheng Liu, Yu Lu, and Shanbo Cheng. G-DIG: Towards gradient-based DIverse and hiGh-quality instruction data selection for machine translation. In *Proceedings of the 62nd Annual Meeting of the Association for Computational Linguistics (Volume 1: Long Papers)*, pp. 15395–15406, Bangkok, Thailand, August 2024. Association for Computational Linguistics. doi: 10.18653/v1/2024.acl-long.821. URL https://aclanthology.org/2024.acl-long.821/.

Sung Min Park, Kristian Georgiev, Andrew Ilyas, Guillaume Leclerc, and Aleksander Madry. TRAK: Attributing model behavior at scale. In *Proceedings of the 40th International Conference on Machine Learning*, volume 202 of *Proceedings of Machine Learning Research*, pp. 27074–27113. PMLR, 23–29 Jul 2023. URL https://proceedings.mlr.press/v202/park23c.html.

Garima Pruthi, Frederick Liu, Satyen Kale, and Mukund Sundararajan. Estimating training data influence by tracing gradient descent. In *Advances in Neural Information Processing Systems*, volume 33, pp. 19920–19930, 2020. URL https://proceedings.neurips.cc/paper/2020/hash/e6385d39ec9394f2f3a354d9d2b88eec-Abstract.html.

Jiachen T. Wang, Tong Wu, Dawn Song, Prateek Mittal, and Ruoxi Jia. GREATS: Online selection of high-quality data for LLM training in every iteration. In *Advances in Neural Information Processing Systems 37 (NeurIPS 2024)*, 2024. URL https://openreview.net/forum?id=232VcN8tSx.

Jiachen T. Wang, Dawn Song, James Zou, Prateek Mittal, and Ruoxi Jia. Capturing the temporal dependence of training data influence. In *The Thirteenth International Conference on Learning Representations*, 2025. URL https://openreview.net/forum?id=uHLgDEgiS5.

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*, volume 235 of *Proceedings of Machine Learning Research*, pp. 54104–54132. PMLR, 21–27 Jul 2024. URL https://proceedings.mlr.press/v235/xia24c.html.

Tingyu Xia, Bowen Yu, Kai Dang, An Yang, Yuan Wu, Yuan Tian, Yi Chang, and Junyang Lin. Rethinking data selection at scale: Random selection is almost all you need. In *Findings of the Association for Computational Linguistics: EMNLP 2025*, pp. 2698–2711, Suzhou, China, November 2025. Association for Computational Linguistics. doi: 10.18653/v1/2025.findings-emnlp.146. URL https://aclanthology.org/2025.findings-emnlp.146/.

An Yang, Anfeng Li, Baosong Yang, Beichen Zhang, Binyuan Hui, Bo Zheng, Bowen Yu, Chang Gao, Chengen Huang, Chenxu Lv, Chujie Zheng, Dayiheng Liu, Fan Zhou, Fei Huang, Feng Hu, Hao Ge, Haoran Wei, Huan Lin, Jialong Tang, Jian Yang, Jianhong Tu, Jianwei Zhang, Jianxin Yang, Jiaxi Yang, Jing Zhou, Jingren Zhou, Junyang Lin, Kai Dang, Keqin Bao, Kexin Yang, Le Yu, Lianghao Deng, Mei Li, Mingfeng Xue, Mingze Li, Pei Zhang, Peng Wang, Qin Zhu, Rui Men, Ruize Gao, Shixuan Liu, Shuang Luo, Tianhao Li, Tianyi Tang, Wenbiao Yin, Xingzhang Ren, Xinyu Wang, Xinyu Zhang, Xuancheng Ren, Yang Fan, Yang Su, Yichang Zhang, Yinger Zhang, Yu Wan, Yuqiong Liu, Zekun Wang, Zeyu Cui, Zhenru Zhang, Zhipeng Zhou, and Zihan Qiu. Qwen3 technical report. *arXiv preprint arXiv:2505.09388*, 2025. doi: 10.48550/arXiv.2505.09388. URL https://arxiv.org/abs/2505.09388.

Zichun Yu, Spandan Das, and Chenyan Xiong. MATES: Model-aware data selection for efficient pretraining with data influence models. In *Advances in Neural Information Processing Systems*, volume 37, 2024. URL https://dblp.org/rec/conf/nips/YuDX24.

Jia Zhang, Chen-Xi Zhang, Yao Liu, Yi-Xuan Jin, Xiao-Wen Yang, Bo Zheng, Yi Liu, and Lan-Zhe Guo. D3: Diversity, difficulty, and dependability-aware data selection for sample-efficient LLM instruction tuning. In *Proceedings of the Thirty-Fourth International Joint Conference on Artificial Intelligence, IJCAI-25*, pp. 8348–8356. International Joint Conferences on Artificial Intelligence Organization, 2025a. doi: 10.24963/ijcai.2025/928. URL https://www.ijcai.org/proceedings/2025/928.

Jipeng Zhang, Yaxuan Qin, Renjie Pi, Weizhong Zhang, Rui Pan, and Tong Zhang. TAGCOS: Task-agnostic gradient clustered COreset selection for instruction tuning data. In *Findings of the Association for Computational Linguistics: NAACL 2025*, pp. 4686–4701, Albuquerque, New Mexico, April 2025b. Association for Computational Linguistics. doi: 10.18653/v1/2025.findings-naacl.264. URL https://aclanthology.org/2025.findings-naacl.264/.

Chunting Zhou, Pengfei Liu, Puxin Xu, Srinivasan Iyer, Jiao Sun, Yuning Mao, Xuezhe Ma, Avia Efrat, Ping Yu, Lili Yu, Susan Zhang, Gargi Ghosh, Mike Lewis, Luke Zettlemoyer, and Omer Levy. LIMA: Less is more for alignment. In *Advances in Neural Information Processing Systems*, volume 36, 2023. URL https://openreview.net/forum?id=KBMOKmX2he.

## A Proof of Proposition 1

Let $\mathcal{M} = \{i : c(\tilde{s}_i) \geq \tau - \varepsilon\}$ and assume $\mathcal{R} \supseteq \mathcal{M}$. Consider first any $i \in \mathcal{T}_k$. By definition $s_i^r \geq \tau$, and by the error bound $c(\tilde{s}_i) \geq s_i^r - \varepsilon \geq \tau - \varepsilon$, so $i \in \mathcal{M} \subseteq \mathcal{R}$ and the hybrid score is exact, $\hat{s}_i = s_i^r \geq \tau$. Consider next any $j \notin \mathcal{R}$. Then $j \notin \mathcal{M}$, so $c(\tilde{s}_j) < \tau - \varepsilon$, and the error bound gives $s_j^r \leq c(\tilde{s}_j) + \varepsilon < \tau$, hence $j \notin \mathcal{T}_k$; its hybrid score is $\hat{s}_j = c(\tilde{s}_j) < \tau - \varepsilon < \tau$. Finally, any $j \in \mathcal{R} \setminus \mathcal{T}_k$ has exact hybrid score $\hat{s}_j = s_j^r < \tau$ because the scores are distinct and $j$ lies outside the top $k$. Hence every member of $\mathcal{T}_k$ has hybrid score at least $\tau$ and every non-member has hybrid score below $\tau$, so the top-$k$ subset under $\hat{s}$ is $\mathcal{T}_k$. □

When $\mathcal{R}$ is the top-$pN$ set under $\tilde{s}$ and the calibration slope $a$ is positive, $\mathcal{R}$ is also the top-$pN$ set under $c(\tilde{s})$, so the condition $\mathcal{R} \supseteq \mathcal{M}$ holds exactly when $pN \geq |\mathcal{M}|$. Since $\mathcal{T}_k \subseteq \mathcal{M}$, this requires $p \geq k/N$; the excess $|\mathcal{M}| - k$ counts the candidates outside $\mathcal{T}_k$ whose calibrated stale score lies within $\varepsilon$ below the threshold, together with the candidates whose stale score exceeds the threshold although their exact score does not. For a stratified rule with strata $(c, b)$, the argument applies within each stratum with its own threshold $\tau_{c,b}$ and quota $k_{c,b}$, and exact recovery of the whole subset requires $\mathcal{R}$ to contain the union of the stratum-level sets $\mathcal{M}_{c,b}$.

## B Algorithm

**Algorithm 1** CDIS: targeted partial refresh with stratified selection

**Require:** Pool $\mathcal{D} = \{x_i\}_{i=1}^N$; cached features $G^\ell = \{g_i^\ell\}_{i=1}^N$; validation direction $v^r$; refresh fraction $p$; subset size $k$; stratum assignment $(c_i, b_i)$; stratum quotas $\{k_{c,b}\}$.
**Ensure:** Selected subset $\mathcal{S}$, updated cache, and check statistic $\rho$.

$\tilde{s}_i \leftarrow \langle g_i^\ell, v^r \rangle$ for all $i \in [N]$ ▷ stale scores
$\mathcal{R}_p \leftarrow \mathrm{TopK}_{\lceil pN \rceil}([N]; \tilde{s})$ ▷ refresh set
**for** $i \in \mathcal{R}_p$ **do**
    recompute $g_i^r$ at $\theta_r$; $s_i^r \leftarrow \langle g_i^r, v^r \rangle$; $g_i^\ell \leftarrow g_i^r$ ▷ update cache
**end for**
$(a, b) \leftarrow \arg\min_{a,b} \sum_{i \in \mathcal{R}_p} (a\tilde{s}_i + b - s_i^r)^2$ ▷ affine calibration
$\rho \leftarrow \mathrm{Spearman}(\{\tilde{s}_i\}_{i \in \mathcal{R}_p}, \{s_i^r\}_{i \in \mathcal{R}_p})$ ▷ online check, Section 4.3
$\hat{s}_i \leftarrow s_i^r$ if $i \in \mathcal{R}_p$ else $a\tilde{s}_i + b$, for all $i \in [N]$
$\mathcal{S} \leftarrow \emptyset$
**for all** strata $(c, b)$ **do**
    $\mathcal{S} \leftarrow \mathcal{S} \cup \mathrm{TopK}_{k_{c,b}}(\{i : c_i = c,\ b_i = b\}; \hat{s})$
**end for**
**if** $|\mathcal{S}| < k$ **then**
    $\mathcal{S} \leftarrow \mathcal{S} \cup \mathrm{TopK}_{k-|\mathcal{S}|}([N] \setminus \mathcal{S}; \hat{s})$ ▷ top up underfilled strata
**end if**
**return** $\mathcal{S}$, $G^\ell$, $\rho$

## C Implementation details

**Candidate pool construction.** For each selection seed we sample 5,000 GSM8K training examples and 5,000 Dolly examples without replacement, convert the GSM8K examples into a chat format with a user turn that asks the model to solve the problem and an assistant turn with the reference solution, keep the Dolly examples in their processed instruction format, and permute the concatenation with the same seed. Each example carries an index, a source label, and a prompt-length proxy equal to the whitespace-token count of all message contents. The GSM8K +FLAN-v2 pool replaces the Dolly half by 5,000 FLAN-v2 examples with the same procedure.

**Warmup and gradient stores.** The warmup follows LESS: LoRA with rank 8, $\alpha = 32$, dropout 0.1 on the query, key, value, and output projections; a random 7% of Dolly; one epoch; per-device batch size 1 with gradient accumulation 8; learning rate $2 \cdot 10^{-5}$ with a linear schedule, warmup ratio 0.03, and weight decay 0; checkpoints every 40 steps. Train features are Adam-aware projected gradients

with $d = 8{,}192$ from the LESS implementation, computed for the whole pool at checkpoints 40, 80, and 120. Validation gradients are computed at each checkpoint from the 100 GSM8K validation examples of the LESS pipeline, for both model families, and averaged into $v^t$. The three-epoch warmup run uses the same data, schedule type, and checkpoint spacing and stops after 396 steps.

**Refresh and selection.** For a refresh fraction $p$, the refresh set is the top $\lceil pN \rceil$ candidates by stale score, the affine calibration is a global least-squares fit on the refresh set, and the hybrid scores follow Eq. 2. The stratified rule allocates 500 examples to each source, splits each source into four equal-count length buckets by the prompt-length proxy, divides each source quota evenly across buckets with largest-remainder rounding, fills each stratum by descending hybrid score, and tops up underfilled strata from the same source and then from the whole pool. The BIDS-style rule standardizes scores within each source, or within each source and length bucket, and takes a global top-$k$. The label-free rule runs $k$-means with $k = 8$ on the checkpoint-80 features and allocates quotas in proportion to cluster size, or uniformly across clusters for the ablation.

**Downstream training and evaluation.** Each selected subset trains a LoRA adapter for five epochs with the warmup hyperparameters and maximum sequence length 512. Evaluation uses the LESS 8-shot GSM8K evaluator with at most 256 new tokens and batch size 4 on the full GSM8K test set and on the prepared ASDiv split. Exact match lowercases prediction and answer, removes punctuation, and collapses whitespace before comparison. Paired bootstrap intervals resample items within each selection seed 10,000 times and report the 2.5th and 97.5th percentiles of the mean per-item difference. Table 3 lists the settings.

| Component | Setting |
|---|---|
| Main model; second family | Qwen3-4B-Base; Llama-3.2-1B |
| Candidate pool | 5k GSM8K train + 5k Dolly (or 5k FLAN-v2), sampled per seed |
| Selection size | top 10% = 1,000 examples |
| Projection dimension; gradient type | 8,192; Adam-aware projected gradients (LESS) |
| Cached and target checkpoints | 80 and 120 (main); 40→80, 40→120 (early); 80→160/240/320 (three-epoch run) |
| Refresh fractions | 0, 0.05, 0.1, 0.2, 0.3, 0.5, 1 |
| Stratified rule | equal source quotas, four length buckets per source |
| LoRA | $r = 8$, $\alpha = 32$, dropout 0.1, modules q/k/v/o |
| Training | 5 epochs, max length 512, lr $2 \cdot 10^{-5}$, batch 1 $\times$ accumulation 8, AdamW, linear schedule |
| Evaluation | 8-shot GSM-style evaluator, 256 new tokens, exact match |
| Uncertainty | paired seed-stratified bootstrap, 10,000 resamples; 12-run training-seed decomposition |

Table 3: Settings for the headline experiments.

## D PROMPT AND ANSWER FORMATS

GSM8K candidate examples use the user message `Solve the following math problem. Question: [problem] Answer:` and the reference solution as the assistant message. Dolly and FLAN-v2 examples keep their processed instruction-tuning messages with a normalized source label. The full GSM8K test split is converted into the evaluator format with `question` and `answer` fields. For ASDiv we concatenate the problem body and question of the `EleutherAI/asdiv` validation split, extract a single numeric answer with a regular expression, skip the 73 items with fractional or non-numeric targets, and write answers with the GSM-style `####` suffix so that the same evaluator, prompts, and decoding settings apply.

## E PER-SEED DOWNSTREAM RESULTS

| | Full GSM8K | | | | Full ASDiv | | | |
|---|---|---|---|---|---|---|---|---|
| Method | Seed 0 | Seed 1 | Seed 2 | Mean | Seed 0 | Seed 1 | Seed 2 | Mean |
| Unconstrained top-$k$, full recompute | 0.196 | 0.237 | 0.356 | 0.263 | 0.126 | 0.164 | 0.199 | 0.163 |
| Plain random | 0.529 | 0.282 | 0.355 | 0.389 | 0.365 | 0.174 | 0.206 | 0.248 |
| Length-matched random | 0.356 | 0.434 | 0.180 | 0.323 | 0.182 | 0.247 | 0.081 | 0.170 |
| Stratified, full recompute | 0.661 | 0.453 | 0.544 | 0.553 | 0.467 | 0.250 | 0.299 | 0.339 |
| CDIS, $p = 0.3$ | 0.616 | 0.477 | 0.438 | 0.510 | 0.436 | 0.289 | 0.232 | 0.319 |

Table 4: Per-seed exact match behind Table 2. Both stratified selectors exceed both random controls on every seed and both test sets; CDIS exceeds full recomputation on seed 1 and trails it on seeds 0 and 2, and the two random controls swap places across seeds.

## F SELECTED-SUBSET PROFILES

| Method | Seed | GSM8K | Dolly | Source entropy | Length mean | Length p10–p90 |
|---|---|---|---|---|---|---|
| Unconstrained top-$k$ | 0 | 1000 | 0 | 0.000 | 44.8 | 30–64 |
| Unconstrained top-$k$ | 1 | 1000 | 0 | 0.000 | 46.9 | 30–69 |
| Plain random | 0 | 525 | 475 | 0.998 | 57.9 | 9–99 |
| Plain random | 1 | 490 | 510 | 1.000 | 61.0 | 10–121 |
| Length-matched random | 0 | 772 | 228 | 0.775 | 43.4 | 26–63 |
| Length-matched random | 1 | 844 | 156 | 0.625 | 46.0 | 29–69 |
| Stratified, full recompute | 0 | 500 | 500 | 1.000 | 69.2 | 10–155 |
| Stratified, full recompute | 1 | 500 | 500 | 1.000 | 54.2 | 9–105 |
| CDIS, $p = 0.3$ | 0 | 500 | 500 | 1.000 | 67.0 | 10–154 |
| CDIS, $p = 0.3$ | 1 | 500 | 500 | 1.000 | 66.3 | 10–145 |

Table 5: Source counts, normalized source entropy, and prompt-length statistics of the selected subsets on seeds 0 and 1. Unconstrained selection collapses to one source and a narrow length window; the stratified rules keep source balance and a wide length range.

## G TRAINING-SEED DECOMPOSITION

| Fixed selection | Seed a | Seed b | Seed c | Mean | SD | Range |
|---|---|---|---|---|---|---|
| Seed-0 pool, stratified full recompute | 0.661 | 0.666 | 0.577 | 0.635 | 0.050 | 0.089 |
| Seed-0 pool, CDIS $p = 0.3$ | 0.616 | 0.583 | 0.557 | 0.586 | 0.030 | 0.059 |
| Seed-2 pool, stratified full recompute | 0.544 | 0.567 | 0.479 | 0.530 | 0.046 | 0.088 |
| Seed-2 pool, CDIS $p = 0.3$ | 0.438 | 0.503 | 0.425 | 0.455 | 0.042 | 0.077 |

Table 6: Training-seed decomposition on full GSM8K. Each row retrains the same selected subset with three training seeds (the original seed and two others) under identical hyperparameters. The pooled within-selection standard deviation is 0.0425.

## H ALTERNATIVES AT EQUAL OR LARGER SAVINGS

| Strategy | Gradient cost | Top-1,000 overlap at step 120 | | |
|---|---|---|---|---|
| | | Qwen3 s0 | Qwen3 s1 | Llama |
| Checkpoint skipping (reuse step-40 features) | half the scoring passes | 0.603 | 0.568 | 0.765 |
| Random refresh, 30% | $0.3N$ per checkpoint | 0.890 | — | 0.920 |
| Targeted refresh, 30% | $0.3N$ per checkpoint | 1.000 | 1.000 | 1.000 |

Table 7: Alternatives at equal or larger savings. Checkpoint skipping reuses features from twice the distance without refresh. Random and targeted refresh recompute the same number of features and differ in which candidates receive them. Random refresh on seed 1 was measured under the stratified rule (Table 8) and is omitted at the score level.

## I RANDOM REFRESH ON DOWNSTREAM QUALITY

| Seed | Refresh policy at 30% | Stratified overlap | Top-500 prefix overlap | GSM200 | Full GSM8K |
|---|---|---|---|---|---|
| 0 | Targeted | 0.927 | 1.000 | 0.600 | 0.616 |
| 0 | Random | 0.839 | 0.826 | 0.550 | 0.545 |
| 1 | Targeted | 0.857 | 1.000 | 0.535 | 0.477 |
| 1 | Random | 0.739 | 0.734 | 0.310 | 0.322 |

Table 8: Targeted against random refresh at the same 30% budget under the stratified rule, with downstream exact match of the resulting adapters.

## J REFRESH-BUDGET CURVE AND CALIBRATION ABLATION

Table 9 lists top-1,000 overlap with full recomputation for every setting, interval, and refresh fraction, with and without the affine calibration. Calibration changes post-warmup intervals at $p \geq 0.2$ by at most 0.001, raises overlap by 1.9 to 7.4 points at $p = 0.1$ and on early or long intervals, and the calibrated overlap is at least the uncalibrated one in every case.

| Setting | Interval | stale | $p = 0.05$ | | $p = 0.1$ | | $p = 0.2$ | $p = 0.3$ | $p = 0.5$ |
|---|---|---|---|---|---|---|---|---|---|
| | | | cal. | raw | cal. | raw | | | |
| Qwen3-4B s0 | 40→80 | 0.631 | 0.648 | 0.631 | 0.689 | 0.631 | 0.826 | 0.931 | 0.999 |
| Qwen3-4B s0 | 40→120 | 0.603 | 0.625 | 0.603 | 0.661 | 0.603 | 0.795 | 0.904 | 0.995 |
| Qwen3-4B s0 | 80→120 | 0.856 | 0.859 | 0.859 | 0.894 | 0.894 | 0.999 | 1.000 | 1.000 |
| Qwen3-4B s1 | 40→80 | 0.629 | 0.654 | 0.629 | 0.693 | 0.629 | 0.819 | 0.925 | 0.999 |
| Qwen3-4B s1 | 40→120 | 0.568 | 0.599 | 0.568 | 0.643 | 0.569 | 0.785 | 0.913 | 0.998 |
| Qwen3-4B s1 | 80→120 | 0.782 | 0.787 | 0.782 | 0.829 | 0.782 | 0.971 | 1.000 | 1.000 |
| Qwen3-4B s2 | 80→120 | 0.817 | 0.824 | 0.818 | 0.856 | 0.837 | 0.993 | 1.000 | 1.000 |
| Llama-3.2-1B | 40→80 | 0.855 | 0.856 | 0.855 | 0.899 | 0.855 | 0.999 | 1.000 | 1.000 |
| Llama-3.2-1B | 40→120 | 0.765 | 0.777 | 0.765 | 0.833 | 0.765 | 0.980 | 1.000 | 1.000 |
| Llama-3.2-1B | 80→120 | 0.889 | 0.890 | 0.889 | 0.920 | 0.889 | 1.000 | 1.000 | 1.000 |
| Qwen3-4B FLAN | 80→120 | 0.855 | 0.858 | 0.858 | 0.893 | 0.893 | 0.999 | 1.000 | 1.000 |
| Qwen3-4B 3 ep. | 80→160 | 0.736 | — | — | — | — | — | 0.992 | — |
| Qwen3-4B 3 ep. | 80→240 | 0.702 | — | — | — | — | — | 0.985 | — |
| Qwen3-4B 3 ep. | 80→320 | 0.688 | — | — | — | — | — | 0.981 | — |

Table 9: Top-1,000 overlap with full recomputation for targeted refresh. At $p = 0.05$ and $p = 0.1$ the columns give the value with (cal.) and without (raw) affine calibration; at $p \geq 0.2$ the two agree to within 0.001 on every interval and one value is shown.

## K THE ONLINE CHECK ON EVERY SINGLE-REFRESH INTERVAL

| Setting | Interval | Check | Stale | Targeted |
|---|---|---|---|---|
| Qwen3-4B s0 | 40→80 | 0.641 | 0.631 | 0.931 |
| Qwen3-4B s1 | 40→80 | 0.647 | 0.629 | 0.925 |
| Qwen3-4B s0 | 40→120 | 0.589 | 0.603 | 0.904 |
| Qwen3-4B s1 | 40→120 | 0.533 | 0.568 | 0.913 |
| Qwen3-4B s0 | 80→120 | 0.912 | 0.856 | 1.000 |
| Qwen3-4B s1 | 80→120 | 0.810 | 0.782 | 1.000 |
| Qwen3-4B s2 | 80→120 | 0.876 | 0.817 | 1.000 |
| Qwen3-4B FLAN | 80→120 | 0.911 | 0.855 | 1.000 |
| Llama-3.2-1B | 40→80 | 0.922 | 0.855 | 1.000 |
| Llama-3.2-1B | 40→120 | 0.817 | 0.765 | 1.000 |
| Llama-3.2-1B | 80→120 | 0.958 | 0.889 | 1.000 |
| Qwen3-4B 3 ep. | 80→160 | 0.739 | 0.736 | 0.992 |
| Qwen3-4B 3 ep. | 80→240 | 0.706 | 0.702 | 0.985 |
| Qwen3-4B 3 ep. | 80→320 | 0.687 | 0.688 | 0.981 |

Table 10: The online check on all fourteen single-refresh intervals. Check is the Spearman correlation between stale and recomputed scores on the $p = 0.3$ refresh set, observable at refresh time; Stale and Targeted are the top-1,000 overlaps of stale reuse and of targeted refresh at $p = 0.3$ with full recomputation.

| Setting | Interval | Global $\rho$ | Stale top-10% | Check $p$=0.1 | Check $p$=0.2 | Check $p$=0.3 | Band $R^2$ |
|---|---|---|---|---|---|---|---|
| Qwen3-4B s0 | 40→80 | 0.911 | 0.631 | 0.526 | 0.621 | 0.641 | 0.433 |
| Qwen3-4B s0 | 80→120 | 0.987 | 0.856 | 0.811 | 0.864 | 0.912 | 0.860 |
| Qwen3-4B s1 | 40→80 | 0.916 | 0.629 | 0.509 | 0.612 | 0.647 | 0.439 |
| Qwen3-4B s1 | 80→120 | 0.976 | 0.782 | 0.745 | 0.781 | 0.810 | 0.717 |

Table 11: The global rank correlation stays above 0.91 on the early interval although the top-10% overlap drops to 0.63, whereas the check on the refresh set drops to 0.64.

## L REFRESH ALLOCATION AND ITERATED CHAINS

**Allocation.** Table 12 lists, for every setting of Table 1, the top-1,000 and stratified overlaps of global and per-stratum allocation at $p \in \{0.1, 0.2, 0.3\}$ together with the source of the examples that the policy misses. Per-stratum allocation spends part of the budget on strata whose scores are low under the target, so its top-1,000 overlap sits below that of global allocation; its target is the stratified subset.

| | | Global allocation, top-1,000 / stratified | | | Per-stratum allocation, top-1,000 / stratified | | |
|---|---|---|---|---|---|---|---|
| Setting | stale | $p$=0.1 | $p$=0.2 | $p$=0.3 | $p$=0.1 | $p$=0.2 | $p$=0.3 |
| Qwen3-4B s0, GSM8K | 0.856 / 0.824 | 0.894 / 0.879 | 0.999 / 0.920 | 1.000 / 0.927 | 0.871 / 0.841 | 0.910 / 0.986 | 0.935 / 0.999 |
| Qwen3-4B s1, GSM8K | 0.782 / 0.720 | 0.829 / 0.792 | 0.971 / 0.837 | 1.000 / 0.857 | 0.817 / 0.772 | 0.884 / 0.955 | 0.920 / 0.996 |
| Qwen3-4B s2, GSM8K | 0.817 / 0.830 | 0.856 / 0.889 | 0.993 / 0.935 | 1.000 / 0.940 | 0.847 / 0.852 | 0.894 / 0.986 | 0.919 / 0.998 |
| Llama-3.2-1B, GSM8K | 0.889 / 0.829 | 0.920 / 0.871 | 1.000 / 0.885 | 1.000 / 0.904 | 0.906 / 0.866 | 0.915 / 0.978 | 0.937 / 0.992 |
| Qwen3-4B, FLAN pool, GSM8K | 0.855 / 0.833 | 0.893 / 0.888 | 0.999 / 0.929 | 1.000 / 0.936 | 0.871 / 0.874 | 0.908 / 0.989 | 0.935 / 1.000 |
| Qwen3-4B s0, MMLU | 0.872 / 0.733 | 0.907 / 0.816 | 0.996 / 0.819 | 1.000 / 0.819 | 0.878 / 0.777 | 0.918 / 0.934 | 0.984 / 0.977 |
| Qwen3-4B s1, MMLU | 0.896 / 0.698 | 0.921 / 0.756 | 0.996 / 0.765 | 1.000 / 0.765 | 0.896 / 0.739 | 0.936 / 0.864 | 0.975 / 0.919 |

Table 12: Global against per-stratum refresh allocation at equal budget, checkpoint 80 to 120. Under the GSM8K target every example that global allocation misses at $p = 0.3$ is a Dolly example (73, 143, 60, 96, and 64 of 1,000 in the five rows); under the MMLU target every missed example is a GSM8K example (181 and 235).

**Iterated chains.** Table 13 gives the post-warmup chain of Figure 3(b) in full, and Table 14 lists the chains that start at checkpoint 40, the early part of warmup, for both allocations and both calibrations, and the post-warmup chain with the cohort calibration.

| Policy ($p = 0.3$) | Cost | Step 160 | Step 240 | Step 320 | Aggregate |
|---|---|---|---|---|---|
| Full recomputation | 4.0 | 1.000 / 1.000 | 1.000 / 1.000 | 1.000 / 1.000 | 1.000 / 1.000 |
| Stale features | 1.0 | 0.736 / 0.573 | 0.702 / 0.625 | 0.688 / 0.586 | 0.820 / 0.713 |
| Iterated, global allocation | 1.9 | 0.992 / 0.749 | 0.984 / 0.816 | 0.979 / 0.786 | 0.999 / 0.844 |
| Iterated, per-stratum allocation | 1.9 | 0.846 / 0.942 | 0.790 / 0.944 | 0.777 / 0.939 | 0.846 / 0.990 |

Table 13: Repeated selection with an iterated cache on the three-epoch run (Qwen3-4B seed 0): a full pass at checkpoint 80, then $0.3N$ features refreshed and written back at each later checkpoint. Cost is the cumulative number of train-gradient features per example. Cells are top-1,000 / stratified overlap with full recomputation at that checkpoint; Aggregate compares the learning-rate-weighted sum of the four checkpoint scores, the LESS selection rule.

The naive calibration ranks by raw stale score and fits one affine map on the refresh set. The cohort calibration refreshes a uniform random pilot of 5% of every age cohort of the cache (candidates whose features were last recomputed at the same checkpoint), fits one affine map per cohort on the pilot, ranks the remaining candidates by cohort-calibrated stale score, spends the rest of the budget on that ranking, and refits the cohort maps on the whole refresh set. Check is the Spearman correlation between stale and recomputed scores on the refresh set at that step.

| Setting | Policy ($p = 0.3$) | Step 1 | Check | Step 2 | Check | Step 3 | Aggregate |
|---|---|---|---|---|---|---|---|
| Qwen3-4B s0, 40→80→120 | stale features | 0.631 / 0.456 | — | 0.603 / 0.477 | — | — | 0.782 / 0.664 |
| | global, naive | 0.931 / 0.688 | 0.641 | 0.898 / 0.702 | 0.950 | — | 0.992 / 0.792 |
| | global, cohort | 0.908 / 0.676 | 0.722 | 0.927 / 0.727 | 0.875 | — | 0.988 / 0.785 |
| | per-stratum, naive | 0.675 / 0.818 | 0.894 | 0.613 / 0.838 | 0.973 | — | 0.811 / 0.935 |
| | per-stratum, cohort | 0.676 / 0.791 | 0.904 | 0.751 / 0.875 | 0.944 | — | 0.844 / 0.912 |
| Qwen3-4B s1, 40→80→120 | stale features | 0.629 / 0.473 | — | 0.568 / 0.394 | — | — | 0.775 / 0.649 |
| | global, naive | 0.925 / 0.693 | 0.647 | 0.900 / 0.622 | 0.874 | — | 0.999 / 0.788 |
| | global, cohort | 0.896 / 0.692 | 0.732 | 0.939 / 0.670 | 0.805 | — | 0.992 / 0.790 |
| | per-stratum, naive | 0.681 / 0.816 | 0.892 | 0.625 / 0.766 | 0.920 | — | 0.828 / 0.939 |
| | per-stratum, cohort | 0.679 / 0.795 | 0.902 | 0.768 / 0.856 | 0.900 | — | 0.861 / 0.930 |
| Llama-3.2-1B, 40→80→120 | stale features | 0.855 / 0.668 | — | 0.765 / 0.566 | — | — | 0.911 / 0.765 |
| | global, naive | 1.000 / 0.782 | 0.922 | 1.000 / 0.723 | 0.964 | — | 1.000 / 0.848 |
| | global, cohort | 1.000 / 0.781 | 0.932 | 1.000 / 0.735 | 0.963 | — | 1.000 / 0.853 |
| | per-stratum, naive | 0.906 / 0.919 | 0.927 | 0.695 / 0.551 | 0.865 | — | 0.894 / 0.901 |
| | per-stratum, cohort | 0.898 / 0.910 | 0.943 | 0.861 / 0.881 | 0.812 | — | 0.946 / 0.958 |
| Qwen3-4B s0, 80→160→240→320 | global, cohort | 0.981 / 0.745 | 0.805 | 0.979 / 0.817 | 0.949 | 0.981 / 0.790 | 0.997 / 0.841 |
| | per-stratum, cohort | 0.840 / 0.923 | 0.877 | 0.840 / 0.918 | 0.921 | 0.869 / 0.964 | 0.906 / 0.970 |

Table 14: Iterated chains under the GSM8K target: early-origin chains with both calibrations, and the post-warmup chain with the cohort calibration (its naive rows are in Table 13 above). Cells are top-1,000 / stratified overlap with full recomputation at that checkpoint; Aggregate compares the learning-rate-weighted checkpoint sums. Under the MMLU target the early-origin global chains reach 0.891 and 0.929 top-1,000 overlap at checkpoint 120 and 0.960 and 0.973 for the aggregate on seeds 0 and 1.

## M SELECTION RULES

| Rule (GSM200, seed 0) | Full | $p = 0.3$ |
|---|---|---|
| Unconstrained top-$k$ | 0.220 | — |
| Length quotas | 0.100 | 0.160 |
| Source quotas | 0.410 | 0.440 |
| Source × length quotas (50/50) | 0.655 | 0.600 |
| Source × length quotas (70/30) | 0.400 | 0.440 |
| Source × length quotas (80/20) | 0.350 | 0.345 |
| BIDS-style, source | 0.535 | 0.540 |
| BIDS-style, source × length | 0.615 | 0.625 |

| Rule (full GSM8K, seed 0) | Full | $p = 0.3$ |
|---|---|---|
| Source × length quotas (labels) | 0.661 | 0.616 |
| Cluster × length, proportional | 0.579 | 0.468 |
| Cluster × length, uniform | 0.350 | 0.406 |
| Plain random | 0.529 | — |
| Unconstrained top-$k$ | 0.196 | — |

Table 15: Selection rules on seed 0. Left: single-run exact match on the 200-item GSM200 probe for quota, ratio, and normalization variants. Right: single-run full GSM8K exact match for the label-free cluster rule ($k$-means with $k = 8$ on cached features) against the labeled rule and controls; Table 6 gives the training-run range for single runs.

## N COST ACCOUNTING

| Stage | Full recompute | CDIS $p = 0.3$ | Changed |
|---|---|---|---|
| Train-gradient features at checkpoint 120 | 3,935.75 s | 1,087.33 s | yes |
| Validation-gradient extraction | 106.70 s | 106.70 s | no |
| Scoring, calibration, selection (CPU) | 6.83 s | 6.83 s | no |
| Selection cycle | 4,049.28 s | 1,200.86 s | 3.37× |
| LoRA training on 1,000 selected examples | 1,433 s | 1,442 s | no |

Table 16: Wall-clock accounting of one selection cycle on Qwen3-4B-Base with the 10k seed-0 pool on one NVIDIA A6000. The train-gradient stage covers 10,000 examples for full recomputation and 3,000 for CDIS; LoRA times are two-seed means. The cache footprint is $10{,}000 \times 8{,}192$ float32 values, about 313 MB.

All inputs of the amortized scenario are the measured values of Table 16. Over $R$ selection cycles the saving is $R \cdot 4049/(4049 + (R-1) \cdot 1201)$, since CDIS pays one full pass to build the cache. The pool-size extrapolation scales the train-gradient stage with the pool and keeps validation gradients, materialization, and training fixed.

| Cycles $R$ | Selection stage | With training |
|---|---|---|
| 1 | 1.00× | 1.00× |
| 2 | 1.54× | 1.35× |
| 3 | 1.88× | 1.53× |
| 5 | 2.29× | 1.71× |
| 10 | 2.73× | 1.87× |
| $\infty$ | 3.37× | 2.07× |

| Pool | Selection cycle | With training | $R = 5$, cache paid |
|---|---|---|---|
| 10k | 3.37× | 2.07× | 2.29× |
| 50k | 3.57× | 3.04× | 2.36× |
| 100k | 3.59× | 3.29× | 2.37× |

Table 17: Left: speedup over $R$ selection cycles when CDIS pays one full pass to build the cache. Right: steady-state per-cycle speedup as the pool grows.

## O SUBSET TRANSFER TO A THIRD MODEL FAMILY

To test whether the cached subset behaves like the recomputed subset when a different model consumes it, we train Mistral-7B-v0.3 on the seed-0 subsets selected by Qwen3-4B with the stratified rule under full recomputation and under $p = 0.3$, plus a plain-random control, without any Mistral-side gradient computation. The two stratified subsets reach 0.397 and 0.396 full GSM8K exact match

on Mistral, so the difference between cached and recomputed selection stays negligible under transfer. Plain random reaches 0.427, above both influence-selected subsets, because those subsets were chosen with Qwen3-4B gradients and carry the length profile of Qwen3-4B's preferences; for Mistral they are a fixed, source-balanced sample selected by another model, and the diagnostic isolates the cache from the selector. Table 1 reports the within-model result for the second family.

| Transferred subset (selected by Qwen3-4B, seed 0) | Mistral train loss | Mistral full GSM8K EM |
|---|---|---|
| Plain random | 0.740 | 0.427 |
| Stratified, full recompute | 0.619 | 0.397 |
| CDIS, $p = 0.3$ | 0.649 | 0.396 |

Table 18: Subset transfer to Mistral-7B-v0.3.

## AI USE STATEMENT

In this work, we used generative AI tools (a large language model assistant) for writing support: editing the manuscript text for grammar and clarity, proofreading, checking that terminology and notation stay consistent across sections, and formatting references. We have reviewed all AI-assisted work: the authors read every edited passage against the original wording, checked that each sentence states the result recorded in the experiment outputs, and verified the formatted references against their sources. We take responsibility for the final content of this work, including text, claims, and artifacts produced with the aid of generative AI.

## REPRODUCIBILITY STATEMENT

Appendix C lists every hyperparameter of warmup, gradient collection, selection, downstream training, and evaluation, and Appendix D gives the prompt and answer formats. The code release contains the refresh and calibration implementation, the stratified, normalized, and cluster-based selection rules, the global and per-stratum allocation and iterated-cache simulations, the paired bootstrap and training-seed decomposition tools, the refresh-budget and band-agreement analysis scripts, the patch that adapts the public LESS pipeline, and the run scripts for every table in this paper. All datasets are public.

## ETHICS STATEMENT

This work studies data selection for post-training language models on public datasets and introduces no human-subject data. Cheaper selection lowers the cost of targeted post-training, and our results show that influence scores applied without diversity constraints select a narrow subset that generalizes poorly, which argues for evaluating selected data on coverage and transfer as well as target accuracy.