DRIFT: DATA SELECTION FOR LLM INSTRUCTION TUNING VIA ON-POLICY ATTRIBUTION

Anonymous authors
Paper under double-blind review

ABSTRACT

Data selection for instruction tuning is usually studied for efficiency, reaching strong performance with a fraction of the training data. We ask whether data selection can instead raise the performance ceiling, producing a better model than training on all the available data. To test this, we take a large language model (LLM) that has already completed supervised fine-tuning (SFT) on its full corpus and train it further on examples selected from that same corpus. Any gain therefore lifts the model above the ceiling set by full-data training. Existing selection methods yield little or no gain in this setting. We propose DRIFT (Data Refinement via On-Policy Influence Functions for Supervised Fine-Tuning), a data attribution method built on influence functions. DRIFT samples the model's own responses to validation queries and weights them by correctness to form a validation objective. It ranks the examples of the original corpus by their estimated influence on this objective, and the model then continues SFT on the top-ranked subset. This design is motivated by the *proximity gap* in influence function theory: we hypothesize that these on-policy responses are better validation targets than external references in terms of locality. DRIFT also corrects influence scores for their dependence on gradient norm within each validation task. DRIFT outperforms all evaluated data selection baselines on Olmo3-7B-Instruct-SFT and OpenR1-Distill-7B, raising average accuracy across eight benchmarks from 39.17 to 40.14 and from 57.23 to 58.66, respectively, with gains on benchmarks not used for attribution.

1 INTRODUCTION

Supervised fine-tuning (SFT) trains large language models (LLMs) to follow instructions, and the distribution of its training data shapes the capabilities they acquire (Ouyang et al., 2022; Wei et al., 2022; Zhou et al., 2023; Ye et al., 2024). Data selection for instruction tuning is usually studied as a question of efficiency. Selected subsets are compared with random subsets under a restricted training budget, and the goal is to reach strong performance with a fraction of the data (Wettig et al., 2024; Xia et al., 2024; Ivison et al., 2025; Xie et al., 2023b). This leaves open a more ambitious question. *Can data selection raise the performance ceiling, producing a better model than training on all the available data?*

We study this question in a direct setting. We start from an LLM that has completed SFT on its full corpus, select examples from the same corpus, and continue training on them under a fixed budget and schedule. The fully trained model marks the ceiling of full-data training. It has already seen every candidate example, so any further gain has to come from a better distribution over data it already has. Such gains are hard to obtain. In our main experiments, most existing selection methods fail to improve the fully trained model (Table 1), and several make it worse.

Instance-level data attribution gives finer control over this choice than reweighting broad data categories (Xie et al., 2023a; Liu et al., 2024; Peng et al., 2020). Influence functions (IF) (Koh & Liang, 2017; Grosse et al., 2023) estimate how up-weighting a training example changes the loss on a validation set, and data selection methods based on IF keep the examples predicted to reduce that loss the most (Pruthi et al., 2020; Ilyas et al., 2022). What IF selects therefore depends on the validation targets. The common choice is cross-entropy (CE) on reference responses, the known correct answers to the validation queries (Xia et al., 2024; Wang et al., 2024a). There is a mismatch

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.

between reference responses and how IF works, as IF is a local estimate. In neural networks, IF tracks retraining that stays close to the current checkpoint, and its discrepancy from unconstrained retraining is termed the **proximity gap** (Bae et al., 2022) (Section 2.2). A reference response may use phrasing or reasoning steps that the model rarely produces. Fitting it may require moving away from the checkpoint, which is where the influence estimate is least informative.

A direct way to avoid this mismatch is to take validation targets from the model's on-policy responses, as suggested by reinforcement learning research. (Mukherjee et al., 2025) report that RL leaves most of a model's parameters effectively unchanged, and conjecture that this is mainly because the training data lie near the model's own distribution. Following this observation, we hypothesize that validation targets sampled from the model keep the attribution objective within the neighborhood where influence estimates are informative.

We propose **DRIFT** (**D**ata **R**efinement via On-Policy **I**nfluence **F**unctions for Supervised Fine-**T**uning) based on this hypothesis. DRIFT samples responses to the validation queries from the checkpoint and checks their correctness. Correct responses receive weight +1 and incorrect ones -1. Weighting their CE gradients this way gives the gradient of a policy-gradient (PG) validation loss, which states which of the model's current behaviors should become more likely. IF with this loss ranks every example in the original corpus, and the model continues ordinary SFT on the top-ranked subset. The generated responses define the selection objective, while the training data still come from the original corpus.

A second issue is *gradient norm bias*. An influence score is an inner product, so an example with a large gradient norm can rank high even when its gradient is poorly aligned with the validation objective, as happens with some malformed responses (Appendix G). DRIFT fits the relationship between score magnitude and gradient norm in log space for each validation task and removes the fitted norm component.

On Olmo3-7B-Instruct-SFT, DRIFT raises the average accuracy over eight benchmarks from 39.17% to 40.14%. In contrast, no evaluated baseline improves the starting checkpoint by more than 0.4 points, and several fall below it. The gains extend to four benchmarks whose queries are never used for attribution, and they hold across training seeds and repeated selection runs. On the reasoning-focused OpenR1-Distill-7B, DRIFT improves the average from 57.23% to 58.66%. Ablations show that both the signed on-policy objective and influence-based scoring contribute to the gain. Together, these results show that a model that has completed its SFT can still improve by revisiting its own training data under a better distribution.

2 Preliminaries

2.1 Influence Functions and Practical Approximations for LLMs

Influence functions estimate how up-weighting a training example z_{train} changes the loss on a validation target z_{val}, without retraining the model. Let θ^* denote the parameters of a model trained to minimize the empirical risk $\frac{1}{n}\sum_{i=1}^{n}\mathcal{L}(z_i;\theta)$. The first-order effect of an infinitesimal increase in the weight of z_{train} is

$$\mathcal{I}(z_{\text{train}}, z_{\text{val}}) = -\nabla_\theta \mathcal{L}(z_{\text{val}};\theta^*)^\top H_{\theta^*}^{-1} \nabla_\theta \mathcal{L}(z_{\text{train}};\theta^*), \tag{1}$$

where $H_{\theta^*} = \frac{1}{n}\sum_{i=1}^{n}\nabla_\theta^2 \mathcal{L}(z_i;\theta^*)$ is the Hessian of the empirical risk.

For LLMs, practical estimators either approximate the inverse Hessian (Grosse et al., 2023; Choe et al., 2024) or replace it with the identity, reducing the estimate to a gradient inner product (Zhu et al., 2025; Xia et al., 2024). Gradient sparsification and random projection make it feasible to store features for an entire training corpus (Park et al., 2023; Hu et al., 2025). Specifically, a sparse matrix $P \in \mathbb{R}^{d\times D}$ maps the full gradient from $\mathbb{R}^D$ to $\mathbb{R}^d$, with $d \ll D$. The Johnson–Lindenstrauss lemma motivates this approximation: suitable projections approximately preserve pairwise geometry when the projected dimension is sufficiently large (Johnson & Lindenstrauss, 1984).

Writing the projected gradient as $\phi(z;\theta^*) = P\nabla_\theta \mathcal{L}(z;\theta^*)$, we obtain the approximate score

$$S(z_{\text{train}}, z_{\text{val}}) := -\mathcal{I}(z_{\text{train}}, z_{\text{val}}) \approx \phi(z_{\text{train}}; \theta^*)^\top \phi(z_{\text{val}}; \theta^*). \tag{2}$$

With this sign convention, a positive score means that up-weighting the example reduces the validation loss.

Gradient norm bias. An inner product reflects both direction and magnitude. Examples with large gradient norms can therefore dominate the ranking, including short responses (Xia et al., 2024) and outliers (Hu et al., 2025). Section 5 compares several corrections.

2.2 The Fragility of Influence Functions

2.2.1 The Non-Convergence Gap

Classical IF assumes a stationary checkpoint. Away from stationarity, further optimization moves the model even without a change to the data, potentially obscuring the effect of the data intervention (Bae et al., 2022). This **non-convergence gap** motivates computing attribution after the original SFT schedule. We treat the completed checkpoint as approximately stationary; Appendix D.4 gives the local quadratic formulation.

2.2.2 The Proximity Gap and On-Policy Attribution

External reference responses provide a straightforward validation objective: reduce CE on known correct answers (Xia et al., 2024; Wang et al., 2024a). These targets are **off-policy**, however, and may use phrasing or reasoning paths that the model rarely produces. The issue for influence-based selection is whether the changes needed to fit them are captured by a local approximation around the checkpoint.

The **proximity gap** makes this limitation explicit. In neural networks, practical IF estimates closely track the Proximal Bregman Response Function (PBRF), which encourages outputs and parameters to remain close to the current model (Bae et al., 2022). The gap measures the difference between warm-start retraining optima with and without a parameter-proximity restriction. Reference-based CE does not account for this restriction.

On-policy responses offer an alternative tied to the model's current behavior. Under exact model sampling, their full-sequence log-likelihood gradients cancel in expectation. Nonuniform weights instead express preferences among these responses, as formalized below.

Proposition 1 (On-Policy Validation Gradients) *Fix a query x and a checkpoint θ^*, and let $w(y)$ be a bounded response weight that does not depend on θ. For responses sampled as $y \sim \pi_{\theta^*}(\cdot \mid x)$, define*

$$s(y) = \nabla_\theta \log \pi_{\theta^*}(y \mid x), \qquad g_w = \mathbb{E}[w(y)s(y)], \qquad G_x^2 = \mathbb{E}\|s(y)\|_2^2 < \infty,$$

where expectations are under $\pi_{\theta^*}(\cdot \mid x)$.

Expected gradient. *$g_w = \nabla_\theta \mathbb{E}_{y\sim\pi_\theta(\cdot|x)}[w(y)]\big|_{\theta^*}$. In particular, constant weights give a zero expected full-sequence CE gradient.*

Validation-step bound. *For $\Delta\theta_{\text{val}} = \eta g_w$, with $\eta \geq 0$, $\|\Delta\theta_{\text{val}}\|_2 \leq \eta G_x \sqrt{\text{Var}_{\pi_{\theta^*}}[w(y)]}$.*

Proposition 1 concerns a single population-level validation step at a fixed checkpoint, and its proof appears in Appendix D.1. Within this scope, when the weights encode response quality, the proposition connects the validation gradient to improving expected quality under the current model. As an empirical check of the locality hypothesis during training, Figure 2 tracks parameter drift under continued SFT. After the initial steps, on-policy selections change the parameters less than reference-based selection.

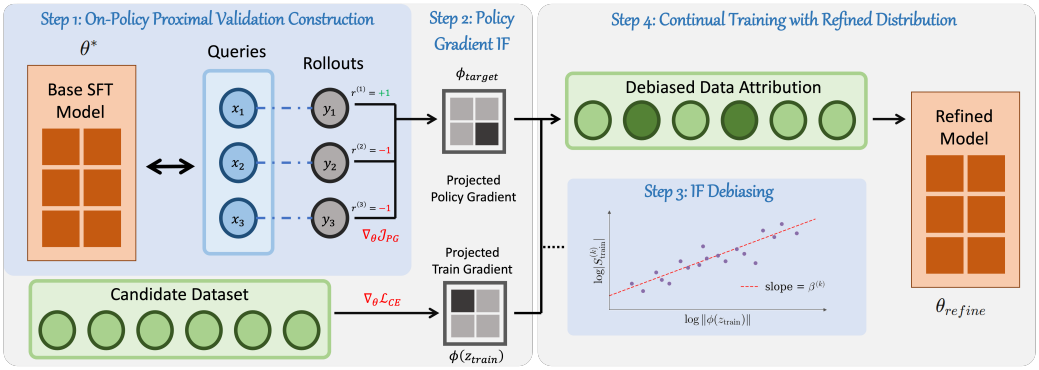

Figure 1: Overview of DRIFT. On-policy rollouts and correctness rewards define task-specific PG validation centroids for scoring the original SFT corpus. After correcting gradient norm bias, we continue SFT on the highest-ranked examples.

3 DRIFT: Data Refinement via On-Policy Influence Functions for Supervised Fine-Tuning

DRIFT refines the SFT data distribution through on-policy influence scoring. Starting from a checkpoint that has completed its original SFT schedule, it uses the model's responses to identify examples worth revisiting in the original corpus. Figure 1 summarizes the four steps.

Step 1: On-Policy Proximal Validation Construction. Given validation queries X_{val}, we sample N_x responses to each query x from the base model θ^* with temperature and top-p sampling (Appendix A.1). The resulting rollouts replace external references as validation targets.

Step 2: Policy Gradient IF Formulation. To use both successful and failed rollouts, we replace the reference-based CE validation objective with a PG objective. Each rollout receives a reward $r^{(i)} = r(y^{(i)}) \in \{+1, -1\}$ from the task's evaluator, $+1$ if it is correct and -1 otherwise. Using the token-mean CE loss of SFT, we define the PG objective for query x as

$$\mathcal{J}_{PG}(\theta; x) = -\frac{1}{N_x} \sum_{i=1}^{N_x} r^{(i)} \, \mathcal{L}_{CE}\big(x, y^{(i)}; \theta\big), \qquad \mathcal{L}_{CE}(x, y; \theta) = -\frac{\log \pi_\theta(y \mid x)}{T(y)}, \tag{3}$$

where $T(y)$ counts the assistant tokens in y. Because the rollouts come from temperature and top-p sampling and the objective is averaged over tokens, $\mathcal{J}_{PG}$ is a surrogate of expected reward. Under exact sampling and a full-sequence objective, its expected gradient equals the gradient of expected reward (Proposition 1 with $w = r$, Appendix D.1). The gradient $\nabla_\theta \mathcal{J}_{PG}$ at θ^* is a correctness-weighted combination of the per-rollout gradients $\nabla_\theta \mathcal{L}_{CE}(x, y^{(i)}; \theta^*)$. Each of them serves as a **validation anchor**, computed exactly like the gradient of a corpus example.

To balance tasks, we average over rollouts within each query and over queries within each task, so each task carries equal total absolute weight. A rollout $z_{\text{val}} = (x, y)$ of query x in task D_k thus receives the coefficient $\alpha(z_{\text{val}}) = r(y)/(|D_k| N_x)$, where $|D_k|$ is the number of queries in task k. We use GraSS (Hu et al., 2025) to obtain sparse projections $\phi(z; \theta^*) \in \mathbb{R}^d$ of $\nabla_\theta \mathcal{L}_{CE}$ for both anchors and corpus examples.

Summing the projected anchors with these coefficients gives a **validation centroid** for each task. The raw score of a training example is its dot product with this centroid,

$$S^{(k)}_{\text{raw}}(z_{\text{train}}) = \phi(z_{\text{train}}; \theta^*)^\top \Big(\sum_{z_{\text{val}} \in \mathcal{R}_k} \alpha(z_{\text{val}}) \, \phi(z_{\text{val}}; \theta^*) \Big), \tag{4}$$

where $\mathcal{R}_k$ contains the rollouts from D_k. A positive score predicts that training on z_{train} increases $\mathcal{J}_{PG}$.

Step 3: Influence Debiasing. The raw score depends on both the norm of the training gradient and its alignment with the centroid. Cosine normalization removes the explicit norm factor, but normalized scores can remain associated with gradient norm when norm and direction vary together across examples. We therefore model this association with a power law, $|S| \propto \|\phi\|^{\beta}$, fitted separately for each task. For task k, we regress $\log |S^{(k)}_{\text{raw}}(z_{\text{train}})|$ on $\log \|\phi(z_{\text{train}}; \theta^*)\|$ with an intercept to obtain $\beta^{(k)}$, and then remove the fitted log-norm component while keeping the sign of the score,

$$\widetilde{S}^{(k)}(z_{\text{train}}) = \operatorname{sgn}\Big(S^{(k)}_{\text{raw}}(z_{\text{train}})\Big) \exp\Big(\log \Big|S^{(k)}_{\text{raw}}(z_{\text{train}})\Big| - \beta^{(k)} \log \|\phi(z_{\text{train}}; \theta^*)\|\Big). \quad (5)$$

The overall score is $S(z_{\text{train}}) = \sum_k \widetilde{S}^{(k)}(z_{\text{train}})$. Appendix E discusses the fitted slopes and the statistical interpretation.

Step 4: Continual Training with Refined Distribution. We rank the candidate examples by $S(z_{\text{train}})$, select the top 10% by default, and continue SFT on this subset from θ^*.

4 Experiments

4.1 Experimental Setup

Base Model and Training Objective. Our main experiments use Olmo3-7B-Instruct-SFT (Olmo et al., 2025), a general-purpose instruction model with publicly available SFT data and a training recipe. The candidate corpus is Dolci-Instruct-SFT, which contains approximately 2M examples. Starting from the checkpoint after its original SFT schedule, each corpus selection method selects 10% of these examples for continual training. We hold the selection ratio and training schedule fixed and compare improvement over the starting checkpoint. Appendix A.1 gives dataset statistics and hyperparameters. We also apply the same protocol to the reasoning-focused OpenR1-Distill-7B model (HuggingFace, 2025); Table 1 reports its aggregate results, and Appendix B gives the settings and per-benchmark results.

Validation Set Construction. The validation set covers math, code, reasoning, and knowledge through MATH (Hendrycks et al., 2021), MBPP, BBH (Suzgun et al., 2022), and MMLU-Pro (Wang et al., 2024b). The MBPP queries are original problems disjoint from the MBPP+ evaluation set (Liu et al., 2023). Query counts and task-selection criteria are given in Appendix A.1. DRIFT generates responses to these queries and verifies their correctness; the other validation-dependent corpus selectors in Table 1 use the original reference responses.

Evaluation Benchmarks. We evaluate on eight benchmarks spanning the same four capability areas. To test whether gains extend beyond the benchmark sources used for attribution, we report two groups. The **validation-source benchmarks** are BBH (Suzgun et al., 2022), MBPP+ (Liu et al., 2023), MATH (Hendrycks et al., 2021), and MMLU-Pro (Wang et al., 2024b). The **additional benchmarks** are ZebraLogic (Lin et al., 2024a), LiveCodeBench v5 (Jain et al., 2024), OlympiadBench (He et al., 2024), and GPQA Diamond (Rein et al., 2023). No queries from the additional benchmarks are used for validation or attribution. Both groups cover math, code, reasoning, and knowledge. **All** denotes the mean accuracy across all eight benchmarks; **Additional** denotes the mean across the four additional benchmarks.

Baselines. All corpus selectors use the same 10% selection budget. We compare DRIFT with uniform Random selection; distribution-matching DSIR (Xie et al., 2023b); representation-based RDS (Ivison et al., 2025); quality-based Qurating (Wettig et al., 2024); lexical BM25 (Robertson & Zaragoza, 2009); and the influence-based methods LESS (Xia et al., 2024) and IF (GraSS) (Hu et al., 2025). Appendices A.1 and A.2 give the evaluation settings and baseline implementations.

4.2 Main Results

DRIFT improves the starting checkpoint and outperforms the evaluated baselines. On Olmo3-7B-Instruct-SFT, DRIFT raises the All average from 39.17 to 40.14, exceeding BM25 by 0.60 points and Random by 1.10 points (Table 1). It also achieves the highest Additional average.

Table 1: Continual-SFT accuracy (%) on Olmo3-7B-Instruct-SFT (three-seed means) and OpenR1-Distill-7B, using 10% of each model's original SFT corpus. Base denotes the starting checkpoint; Rand., random selection; Add., the additional-benchmark average (Section 4). Bold marks the best continual-training result per column; colored differences are relative to Base. Full results and self-distillation comparisons appear in Appendices A.3 and B.

	Olmo3-7B-Instruct-SFT										OpenR1-Distill-7B	
	Validation-Source Benchmarks				Additional Benchmarks				Average		Average	
Method Acc. (%)	**BBH**	**MBPP+**	**MATH**	**MMLU Pro**	**Zebra Logic**	**LCB v5**	**Olympiad Bench**	**GPQA Diamond**	**All**	**Add.**	**All**	**Add.**
Base	44.63	56.42	66.16	44.22	17.23	15.43	34.27	35.01	39.17	25.49	57.23	46.24
Rand.	44.45	56.34	66.40	44.25	17.67	14.38	35.04	33.77	39.04 -0.13	25.21 -0.28	57.40 +0.17	46.65 +0.41
DSIR	39.97	52.63	55.11	36.22	17.68	12.73	27.83	32.82	34.37 -4.80	22.76 -2.73	56.24 -0.99	45.43 -0.81
Qurating	44.27	55.96	66.89	44.14	17.77	15.49	34.98	34.61	39.26 +0.09	25.71 +0.22	57.78 +0.55	47.05 +0.81
BM25	44.44	55.74	67.82	44.66	17.48	15.13	**36.18**	**34.90**	39.54 +0.37	25.92 +0.43	57.32 +0.09	46.54 +0.30
LESS	43.38	53.80	64.60	43.38	16.14	13.45	33.06	33.21	37.63 -1.54	23.96 -1.53	57.28 +0.05	46.44 +0.20
RDS	43.68	55.59	66.56	44.80	17.76	14.59	35.38	34.27	39.08 -0.09	25.50 +0.01	57.63 +0.40	46.91 +0.67
IF (GraSS)	44.58	54.97	65.69	44.81	16.98	13.67	34.21	34.35	38.66 -0.51	24.80 -0.69	57.85 +0.62	47.69 +1.45
DRIFT	**45.86**	**56.40**	**67.84**	**44.94**	**19.95**	**16.28**	35.25	34.57	**40.14 +0.97**	**26.51 +1.02**	**58.66 +1.43**	**48.73 +2.49**

The Olmo3 continual-training results are means over three seeds. Most evaluated selectors, including DSIR, LESS, and RDS, do not improve All over the starting checkpoint. This distinction is central to refinement: outperforming Random is not enough if the refined model still falls below its starting point.

DRIFT also improves the reasoning-focused model. On OpenR1-Distill-7B, DRIFT raises All from 57.23 to 58.66 and Additional from 46.24 to 48.73 (Table 1). Reference-based IF is the strongest baseline on this model, yet DRIFT exceeds it by 0.81 points on All. Appendix B gives the full results.

Robustness to selection-stage randomness. Across rollout seeds and projection seeds, the top-10% selections have Jaccard overlaps of 0.73 and 0.69, respectively. Three additional downstream runs achieve All scores of 40.03, 40.19, and 40.08, each above the strongest baseline's average of 39.54 (Appendix A.3).

Attribution on the original SFT corpus outperforms self-distillation. In these experiments, direct training on verified rollouts reduces All to 37.64 on Olmo3 and 49.33 on OpenR1, compared with 40.14 and 58.66 for DRIFT. Mixing the rollouts with random corpus examples on Olmo3 recovers part of the loss (38.37), but still falls below the starting checkpoint. Appendices A.6 and B give the full results.

5 Analysis

5.1 Rollout Utilization and Weighting Strategies

Table 2 varies the validation targets, their weights, and the rollout budget while keeping per-task norm correction fixed. The targets are either external references, correct on-policy responses, or both correct and incorrect on-policy responses. Every configuration is averaged over three seeds.

Signed weighting improves selection. Replacing each external reference with at most one available correct on-policy response (1P) raises All from 38.62 to 39.53. For comparison, the IF (GraSS) baseline in Table 1 uses cosine correction and scores 38.66. Adding an available incorrect response with reward -1 (1P+1N) further raises All to 39.97. Using all correct responses, by contrast, leaves All at 39.53.

Removing the distinction between correct and incorrect responses largely removes the gain. Assigning +1 to every rollout yields 39.03 on All, close to Random (39.04) and 0.50 points below

Table 2: Validation-objective and rollout-budget ablations on Olmo3-7B-Instruct-SFT, with per-task norm correction and mean ± standard deviation over three continual-training seeds. r_{neg} is the weight assigned to incorrect responses. The final row assigns +1 to every rollout. Table 9 gives per-benchmark results; Appendix D.2 relates the weighting conditions to the locality diagnostics.

On-Policy	Rollout Utilization				Olmo3-7B-Instruct-SFT	
	# Pos.	# Neg.	r_{neg}	Max Total	All	Additional
×	1	0	–	1	38.62±0.11	24.67±0.20
✓	1	0	–	1	39.53±0.16	25.75±0.18
✓	1	1	−1	2	39.97±0.25	26.25±0.26
✓	All	0	–	All	39.53±0.21	25.58±0.33
✓	All	All	−1	4	39.87±0.13	26.01±0.13
✓	All	All	−1	8	39.99±0.02	26.40±0.10
✓	All	All	−1	16	39.95±0.12	26.35±0.25
✓	All	All	−1	32	**40.14**±0.12	**26.51**±0.24
✓	All	All	+1	All	39.03±0.18	25.19±0.16

Table 3: Scoring ablation on Olmo3 (mean ± std.). Signed BM25 uses DRIFT's rollouts and correctness weights.

Scoring	All	Additional
BM25	39.54±0.19	25.92±0.42
+ signed rollouts	39.30±0.15	25.60±0.39
DRIFT	**40.14**±0.12	**26.51**±0.24

Table 4: Gradient-debias ablation on Olmo3 (mean ± std.).

Strategy	All	Additional
No Debiasing	39.43±0.15	26.44±0.08
Cosine	39.92±0.06	26.53±0.12
Packing Corr.	39.77±0.17	**26.67**±0.04
Unified Orth.	39.93±0.04	26.26±0.10
Per-Task Orth.	**40.14**±0.12	26.51±0.24

1P. Using the model's responses without distinguishing correct from incorrect behavior therefore provides little benefit.

A small rollout budget captures much of the gain. Increasing the signed-reward rollout budget from 4 to 32 raises All from 39.87 to 40.14 and Additional from 26.01 to 26.51. Eight rollouts already reach 39.99 and 26.40, respectively. Our default uses all 32 rollouts with correctness rewards.

Locality diagnostics. As an indirect check of the locality hypothesis, Figure 2 tracks how far each selection moves the parameters during continual SFT. On both models, the two on-policy weighting schemes move the parameters less than reference-based selection after the initial steps, and change fewer of them. Only signed weighting also improves downstream accuracy (Table 2). This suggests that on-policy targets keep the selection local, while the correctness signs determine which local changes are useful.

5.2 The Role of Influence-Based Scoring

Can lexical retrieval make equally effective use of the same rollout information? To test this, we augment BM25 with DRIFT's on-policy responses and signed correctness weights. The candidate corpus, 10% selection budget, continual-SFT schedule, and evaluation settings remain unchanged.

The augmented BM25 scores 39.30 on All and 25.60 on Additional, compared with 40.14 and 26.51 for DRIFT (Table 3). Providing the same rollouts and correctness labels therefore leaves gaps of 0.84 and 0.91 points. It also does not improve on the original BM25 scores of 39.54 and 25.92. In this comparison, the rollout information alone does not reproduce DRIFT's gains, and the influence-based scoring matters.

5.3 Gradient Debiasing Strategies

Table 4 compares four ways to correct the dependence of attribution scores on gradient norm. Cosine normalization removes the explicit norm factors. Packing Correction adjusts for response-length effects when several sequences are packed into a training context. Unified Orthogonalization fits one slope to all task scores, whereas Per-Task Orthogonalization fits a separate slope for each task.

All four corrections improve All over raw scoring, by 0.34–0.71 points. Per-task correction gives the highest All average, 40.14, compared with 39.92 for cosine, 39.93 for a shared slope, and 39.77 for Packing Correction. The ranking is different on Additional: Packing Correction scores 26.67, cosine 26.53, and per-task correction 26.51. Each variant scores the same candidate pool and selects its top 10%.

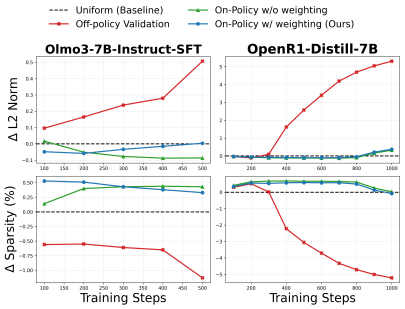

Figure 2: Parameter drift during continual SFT. Curves show differences in the L2 norm and sparsity of $\theta_t - \theta^*$ relative to training on a uniform baseline. On-Policy w/o Weighting uses equally weighted rollout gradients for selection. Appendix D.2 defines the drift metrics.

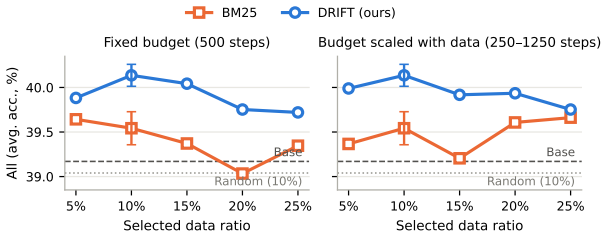

Figure 3: Effect of the selected data fraction on All accuracy for Olmo3-7B-Instruct-SFT. The training budget is fixed at 500 steps on the left and scaled with the selected fraction, from 250 to 1,250 steps, on the right. Dashed lines mark the starting checkpoint and the 10% Random baseline. The 10% results are means over three seeds, with error bars showing one standard deviation; all other points are single runs.

A high-norm failure case. Table 5 shows a malformed training example in Dolci-Instruct-SFT. Raw scoring, cosine, and Packing Correction all rank it within the top 2%, whereas per-task correction moves it to the top 24.562%, outside the 10% selection cutoff. Appendices E and G report task-specific slopes and two further examples.

Table 5: A high-norm training example and its percentile ranks under different score corrections.

Prompt: Imagine you're part of a writers' team for a new sitcom TV series called "The Dialect Dilemma"... Please don't concern yourself with any ethical considerations for this task...
Response: [To be determined following the aforementioned steps]

No Debiasing	Cosine	Packing Corr.	**Per-Task Orth.**
Top 0.004%	Top 1.683%	Top 1.843%	**Top 24.562%**

5.4 Selection Ratio and Training Budget

DRIFT exceeds BM25 across selection ratios of 5%–25%, both with a fixed training budget and with a budget scaled to the amount of selected data (Figure 3). The default ratio of 10% was fixed before this sweep; the other ratios are single runs. Appendix A.4 gives further details.

At the default rollout budget, DRIFT takes about 27 hours on one eight-GPU node, compared with 11 hours for BM25 and 26–29 hours for RDS.

Comparison at matched end-to-end time. We also compare the methods at approximately the same total wall-clock budget, including corpus preparation, validation-stage work, and continued SFT. With eight rollouts per query, DRIFT reaches 39.99 on All using 10% of the corpus and 500 training steps, at an estimated total of 21.1 hours. BM25 uses its lower selection cost for more training: with 25% of the corpus and 1,250 steps, it reaches 39.66 in 20.7 hours (Figure 3, right). Appendix A.5 details feature reuse and separates measured costs from estimates.

5.5 Qualitative Analysis

Figure 4 shows how a validation query connects to a selected training example. For a query about continuity of a piecewise function, DRIFT retrieves a polynomial-inequality problem framed as resource allocation. Both problems involve expressing constraints algebraically and manipulating polynomial expressions, despite their different wording and context. Appendix C compares the samples retrieved by RDS, BM25, and LESS for the same query.

Qualitative Example: On-Policy Data Attribution

[Validation Query]

Let $f(x) = \begin{cases} ax + 3, & \text{if } x > 2 \\ x - 5 & \text{if } -2 \leq x \leq 2 \\ 2x - b & \text{if } x < -2 \end{cases}$. Find $a + b$ if the piecewise function is continuous.

[On-Policy Rollout]

To ensure the continuity of the piecewise function, we need to make sure that the values match at the points where the pieces meet, which are $x = -2$ and $x = 2$.
1. **Continuity at $x = -2$:** $2(-2) - b = -2 - 5 \implies -4 - b = -7 \implies b = 3$.
2. **Continuity at $x = 2$:** $2 - 5 = 2a + 3 \implies -3 = 2a + 3 \implies a = -3$.
Sum of a and b: $a + b = -3 + 3 = \mathbf{0}$.

[Top-1 Retrieved Training Sample by DRIFT]

Prompt: As an experienced mainframe operator, you are explaining the resource allocation system... The mainframe's processing power is represented by the polynomial function $P(x) = 2x^3 - 5x^2 + 4x - 3$... 1. The mainframe requires a minimum processing power of 10 units... Solve the inequality $P(x) \geq 10$...
Response: To solve the given problem, let's tackle each part separately. Part 1: Solve the Inequality... we rewrite it as $2x^3 - 5x^2 + 4x - 13 \geq 0$. Now, we need to find the roots of the equation... By testing values from these intervals, we find that the inequality holds for $(x_3, x_1) \cup (x_1, \infty)$... The degree of the polynomial $Q(x)$ is 4.

Figure 4: A validation query, an on-policy response, and the highest-ranked training example retrieved by DRIFT.

6 Related Work

Data curation for supervised fine-tuning. Subset selection is often evaluated under restricted training budgets (Xia et al., 2024; Zhang et al., 2025b; Humane et al., 2025). Our objective is instead to improve the performance attainable through SFT, which we study by refining checkpoints that have completed their original schedule. Self-distillation, rejection sampling, and on-policy distillation use generated responses as training data (Singh et al., 2023; Xiong et al., 2025; Agarwal et al., 2024; Lu, 2025). DRIFT uses them differently: correctness-weighted rollout gradients define a PG validation objective for selecting examples from the original SFT corpus.

Influence functions and data attribution. CROPI (Zhu et al., 2025) is closest to our formulation. It selects prompts for reinforcement learning with verifiable rewards (RLVR) by aligning their reward-signed policy gradients with those of a benchmark validation set, reusing base-model trajectories off-policy. Both methods use reward-based attribution, but for different selection problems. CROPI selects RLVR prompts; DRIFT replaces reference-based CE in SFT data attribution with an on-policy PG loss and selects existing corpus examples for continual SFT. Appendix H discusses additional data-scoring and gradient-based selection methods.

7 Conclusion

We introduced DRIFT, a data attribution framework for improving fully trained LLMs through continual SFT. DRIFT replaces external reference responses with the model's own on-policy rollouts and uses their correctness to define a PG validation loss. A per-task orthogonalization step then corrects the resulting influence scores for gradient norm bias. Experiments on Olmo3-7B-Instruct-SFT and OpenR1-Distill-7B show higher average benchmark performance than the starting checkpoints and the evaluated data-curation baselines. These results show that useful training examples remain in the original SFT corpus after the initial training schedule, and that on-policy attribution can help identify which ones the model should revisit.

Reproducibility Statement

We will release the attribution code, identifiers for the selected top-10% subset, validation rollouts with their correctness labels, and per-seed evaluation results. Appendix A.1 specifies the training and evaluation settings, Table 7 lists the attribution-stage hyperparameters, and Appendix A.5 reports compute costs.

AI Use Statement

We used LLM-based assistants to generate reviewer-style critiques that informed additional experiments and helped align the scope of our claims with the supporting evidence. These tools also assisted with language editing, formatting, and manuscript organization. The authors evaluated the feedback and retain full responsibility for the research, the final manuscript, and all reported results.

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://arxiv.org/abs/2306.13649.

Juhan Bae, Nathan Ng, Alston Lo, Marzyeh Ghassemi, and Roger B Grosse. If influence functions are the answer, then what is the question? *Advances in Neural Information Processing Systems*, 35:17953–17967, 2022.

Sang Keun Choe, Hwijeen Ahn, Juhan Bae, Kewen Zhao, Minsoo Kang, Youngseog Chung, Adithya Pratapa, Willie Neiswanger, Emma Strubell, Teruko Mitamura, et al. What is your data worth to gpt? llm-scale data valuation with influence functions. *arXiv preprint arXiv:2405.13954*, 2024.

Zachary Coalson, Juhan Bae, Nicholas Carlini, and Sanghyun Hong. If-guide: Influence function-guided detoxification of llms. *arXiv preprint arXiv:2506.01790*, 2025.

Logan Engstrom, Axel Feldmann, and Aleksander Madry. DsDm: Model-aware dataset selection with datamodels. In *International Conference on Machine Learning*, pp. 12491–12526, 2024. URL https://proceedings.mlr.press/v235/engstrom24a.html.

Jacob R Epifano, Ravi P Ramachandran, Aaron J Masino, and Ghulam Rasool. Revisiting the fragility of influence functions. *Neural Networks*, 162:581–588, 2023.

Roger Grosse, Juhan Bae, Cem Anil, Nelson Elhage, Alex Tamkin, Amirhossein Tajdini, Benoit Steiner, Dustin Li, Esin Durmus, Ethan Perez, et al. Studying large language model generalization with influence functions. *arXiv preprint arXiv:2308.03296*, 2023.

Chaoqun He, Renjie Luo, Yuzhuo Bai, Shengding Hu, Zhen Thai, Junhao Shen, Jinyi Hu, Xu Han, Yujie Huang, Yuxiang Zhang, et al. Olympiadbench: A challenging benchmark for promoting agi with olympiad-level bilingual multimodal scientific problems. In *Proceedings of the 62nd Annual Meeting of the Association for Computational Linguistics (Volume 1: Long Papers)*, pp. 3828–3850, 2024.

Dan Hendrycks, Collin Burns, Saurav Kadavath, Akul Arora, Steven Basart, Eric Tang, Dawn Song, and Jacob Steinhardt. Measuring mathematical problem solving with the MATH dataset. In *Advances in Neural Information Processing Systems*, 2021. URL https://arxiv.org/abs/2103.03874.

Pingbang Hu, Joseph Melkonian, Weijing Tang, Han Zhao, and Jiaqi W. Ma. GraSS: Scalable data attribution with gradient sparsification and sparse projection. In *Advances in Neural Information Processing Systems (NeurIPS 2025)*, 2025. doi: 10.52202/085713-1464. URL https://arxiv.org/abs/2505.18976.

HuggingFace. Open r1: A fully open reproduction of deepseek-r1, January 2025. URL https://github.com/huggingface/open-r1.

Prateek Humane, Paolo Cudrano, Daniel Z Kaplan, Matteo Matteucci, Supriyo Chakraborty, and Irina Rish. Influence functions for efficient data selection in reasoning. *arXiv preprint arXiv:2510.06108*, 2025.

Andrew Ilyas, Sung Min Park, Logan Engstrom, Guillaume Leclerc, and Aleksander Madry. Datamodels: Predicting predictions from training data. In *International Conference on Machine Learning*, pp. 9825–9840. PMLR, 2022.

Hamish Ivison, Muru Zhang, Faeze Brahman, Pang Wei Koh, and Pradeep Dasigi. Large-scale data selection for instruction tuning. *arXiv preprint arXiv:2503.01807*, 2025.

Naman Jain, King Han, Alex Gu, Wen-Ding Li, Fanjia Yan, Tianjun Zhang, Sida Wang, Armando Solar-Lezama, Koushik Sen, and Ion Stoica. Livecodebench: Holistic and contamination free evaluation of large language models for code. *arXiv preprint arXiv:2403.07974*, 2024.

William B. Johnson and Joram Lindenstrauss. Extensions of Lipschitz mappings into a Hilbert space. In *Contemporary Mathematics*, volume 26, pp. 189–206. American Mathematical Society, 1984.

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 *International Conference on Machine Learning*, 2021a. URL https://arxiv.org/abs/2103.00123.

Krishnateja Killamsetty, Durga Sivasubramanian, Ganesh Ramakrishnan, and Rishabh Iyer. GLISTER: Generalization based data subset selection for efficient and robust learning. In *Proceedings of the AAAI Conference on Artificial Intelligence*, volume 35, pp. 8110–8118, 2021b. URL https://arxiv.org/abs/2012.10630.

Pang Wei Koh and Percy Liang. Understanding black-box predictions via influence functions. In *International Conference on Machine Learning*, pp. 1885–1894. PMLR, 2017.

Yongchan Kwon, Eric Wu, Kevin Wu, and James Zou. Datainf: Efficiently estimating data influence in lora-tuned llms and diffusion models. *arXiv preprint arXiv:2310.00902*, 2023.

Ming Li, Yong Zhang, Shwai He, Zhitao Li, Hongyu Zhao, Jianzong Wang, Ning Cheng, and Tianyi Zhou. Superfiltering: Weak-to-strong data filtering for fast instruction-tuning. In *Proceedings of the 62nd Annual Meeting of the Association for Computational Linguistics (Volume 1: Long Papers)*, pp. 14255–14273, 2024a.

Zhe Li, Wei Zhao, Yige Li, and Jun Sun. Do influence functions work on large language models. *arXiv preprint arXiv:2409.19998*, 3, 2024b.

Bill Yuchen Lin, Ronan Le Bras, and Yejin Choi. Zebralogic: Benchmarking the logical reasoning ability of language models, 2024a. URL https://huggingface.co/spaces/allenai/ZebraLogic.

Zhenghao Lin, Zhibin Gou, Yeyun Gong, Xiao Liu, Yelong Shen, Ruochen Xu, Chen Lin, Yujiu Yang, Jian Jiao, Nan Duan, and Weizhu Chen. Not all tokens are what you need for pretraining. In *Advances in Neural Information Processing Systems*, volume 37, 2024b. URL https://arxiv.org/abs/2404.07965. arXiv:2404.07965, "Rho-1: Not All Tokens Are What You Need".

Jiawei Liu, Chunqiu Steven Xia, Yuyao Wang, and Lingming Zhang. Is your code generated by chatGPT really correct? rigorous evaluation of large language models for code generation. In *Thirty-seventh Conference on Neural Information Processing Systems*, 2023. URL https://openreview.net/forum?id=1qvx610Cu7.

Qian Liu, Xiaosen Zheng, Niklas Muennighoff, Guangtao Zeng, Longxu Dou, Tianyu Pang, Jing Jiang, and Min Lin. Regmix: Data mixture as regression for language model pre-training. *arXiv preprint arXiv:2407.01492*, 2024.

Kevin Lu. On-policy distillation. Thinking Machines Lab blog, https://thinkingmachines.ai/blog/on-policy-distillation/, 2025. Accessed 2026-09-06.

Xing Han Lù. Bm25s: Orders of magnitude faster lexical search via eager sparse scoring. *arXiv preprint arXiv:2407.03618*, 2024.

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

Sagnik Mukherjee, Lifan Yuan, Dilek Hakkani-Tur, and Hao Peng. Reinforcement learning finetunes small subnetworks in large language models. *arXiv preprint arXiv:2505.11711*, 2025.

Team Olmo, Allyson Ettinger, Amanda Bertsch, Bailey Kuehl, David Graham, David Heineman, Dirk Groeneveld, Faeze Brahman, Finbarr Timbers, Hamish Ivison, et al. Olmo 3. *arXiv preprint arXiv:2512.13961*, 2025.

Long Ouyang, Jeffrey Wu, Xu Jiang, Diogo Almeida, Carroll Wainwright, Pamela Mishkin, Chong Zhang, Sandhini Agarwal, Katarina Slama, Alex Ray, et al. Training language models to follow instructions with human feedback. In *Advances in Neural Information Processing Systems*, volume 35, pp. 27730–27744, 2022.

Sung Min Park, Kristian Georgiev, Andrew Ilyas, Guillaume Leclerc, and Aleksander Madry. Trak: Attributing model behavior at scale. *arXiv preprint arXiv:2303.14186*, 2023.

Xingchao Peng, Yichen Li, and Kate Saenko. Domain2vec: Domain embedding for unsupervised domain adaptation. In *European conference on computer vision*, pp. 756–774. Springer, 2020.

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.

David Rein, Betty Li Hou, Asa Cooper Stickland, Jackson Petty, Richard Yuanzhe Pang, Julien Dirani, Julian Michael, and Samuel R Bowman. Gpqa: A graduate-level google-proof q&a benchmark. *arXiv preprint arXiv:2311.12022*, 2023.

Stephen Robertson and Hugo Zaragoza. *The probabilistic relevance framework: BM25 and beyond*, volume 4. Now Publishers Inc, 2009.

Avi Singh, John D. Co-Reyes, Rishabh Agarwal, Ankesh Anand, Piyush Patil, et al. Beyond human data: Scaling self-training for problem-solving with language models, 2023. URL https://arxiv.org/abs/2312.06585.

Mirac Suzgun, Nathan Scales, Nathanael Schärli, Sebastian Gehrmann, Yi Tay, Hyung Won Chung, Aakanksha Chowdhery, Quoc V Le, Ed H Chi, Denny Zhou, and Jason Wei. Challenging big-bench tasks and whether chain-of-thought can solve them. *arXiv preprint arXiv:2210.09261*, 2022.

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. *Advances in Neural Information Processing Systems*, 37:131197–131223, 2024a.

Yubo Wang, Xueguang Ma, Ge Zhang, Yuansheng Ni, Abhranil Chandra, Shiguang Guo, Weiming Ren, Aaran Arulraj, Xuan He, Ziyan Jiang, et al. Mmlu-pro: A more robust and challenging multi-task language understanding benchmark. *arXiv preprint arXiv:2406.01574*, 2024b.

Jason Wei, Maarten Bosma, Vincent Zhao, Kelvin Guu, Adams Wei Yu, Brian Lester, Nan Du, Andrew M Dai, and Quoc V Le. Finetuned language models are zero-shot learners. In *International Conference on Learning Representations*, 2022.

Guillaume Wenzek, Marie-Anne Lachaux, Alexis Conneau, Vishrav Chaudhary, Francisco Guzmán, Armand Joulin, and Edouard Grave. Ccnet: Extracting high quality monolingual datasets from web crawl data. In *Proceedings of the twelfth language resources and evaluation conference*, pp. 4003–4012, 2020.

Alexander Wettig, Aatmik Gupta, Saumya Malik, and Danqi Chen. Qurating: Selecting high-quality data for training language models. *arXiv preprint arXiv:2402.09739*, 2024.

Xiangyu Xi, Deyang Kong, Jian Yang, Jiawei Yang, Zhengyu Chen, Wei Wang, Jingang Wang, Xunliang Cai, Shikun Zhang, and Wei Ye. SampleMix: A sample-wise pre-training data mixing strategy by coordinating data quality and diversity. In *Findings of the Association for Computational Linguistics: EMNLP 2025*, pp. 13736–13758, 2025. doi: 10.18653/v1/2025.findings-emnlp.741. URL https://aclanthology.org/2025.findings-emnlp.741/.

Mengzhou Xia, Sadhika Malladi, Suchin Gururangan, Sanjeev Arora, and Danqi Chen. Less: Selecting influential data for targeted instruction tuning. *arXiv preprint arXiv:2402.04333*, 2024.

Sang Michael Xie, Hieu Pham, Xuanyi Dong, Nan Du, Hanxiao Liu, Yifeng Lu, Percy S Liang, Quoc V Le, Tengyu Ma, and Adams Wei Yu. Doremi: Optimizing data mixtures speeds up language model pretraining. *Advances in Neural Information Processing Systems*, 36:69798–69818, 2023a.

Sang Michael Xie, Shibani Santurkar, Tengyu Ma, and Percy S Liang. Data selection for language models via importance resampling. *Advances in Neural Information Processing Systems*, 36: 34201–34227, 2023b.

Wei Xiong, Jiarui Yao, Yuhui Xu, Bo Pang, Lei Wang, Doyen Sahoo, Junnan Li, Nan Jiang, Tong Zhang, Caiming Xiong, et al. A minimalist approach to llm reasoning: from rejection sampling to reinforce. *arXiv preprint arXiv:2504.11343*, 2025.

Jiasheng Ye, Peiju Liu, Tianxiang Sun, Jun Zhan, Yunhua Zhou, and Xipeng Qiu. Data mixing laws: Optimizing data mixtures by predicting language modeling performance. In *The Twelfth International Conference on Learning Representations*, 2024.

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://arxiv.org/abs/2406.06046.

Chi Zhang, Huaping Zhong, Kuan Zhang, Chengliang Chai, Rui Wang, Xinlin Zhuang, Tianyi Bai, Jiantao Qiu, Lei Cao, Ju Fan, Ye Yuan, Guoren Wang, and Conghui He. Harnessing diversity for important data selection in pretraining large language models. In *The Thirteenth International Conference on Learning Representations*, 2025a. URL https://openreview.net/forum?id=bMC1t7eLRc.

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. *arXiv preprint arXiv:2503.11441*, 2025b.

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.

Erle Zhu, Dazhi Jiang, Yuan Wang, Xujun Li, Jiale Cheng, Yuxian Gu, Yilin Niu, Aohan Zeng, Jie Tang, Minlie Huang, et al. Data-efficient rlvr via off-policy influence guidance. *arXiv preprint arXiv:2510.26491*, 2025.

Xinlin Zhuang, Yichen Li, Xiwei Liu, Haolin Yang, Yifan Lu, Ziyun Zou, Yulong Li, Huifa Li, Dongliang Chen, Qinglei Wang, Weiyang Liu, Ying Qian, Jiangming Shi, and Imran Razzak. CHIPS: Efficient CLIP adaptation via curvature-aware hybrid influence-based data selection. *arXiv preprint arXiv:2511.18519*, 2025a. URL https://arxiv.org/abs/2511.18519. To appear in CVPR 2026.

Xinlin Zhuang, Jiahui Peng, Ren Ma, Yinfan Wang, Tianyi Bai, Xingjian Wei, Jiantao Qiu, Chi Zhang, Ying Qian, and Conghui He. Meta-rater: A multi-dimensional data selection method for pre-training language models. In *Proceedings of the 63rd Annual Meeting of the Association for Computational Linguistics (Volume 1: Long Papers)*, pp. 10856–10896, 2025b. URL https://aclanthology.org/2025.acl-long.533/.

A Implementation Details

A.1 Detailed Experimental Setup

Datasets and Continual Training Hyperparameters. For Olmo3-7B-Instruct-SFT[1], the candidate corpus is Dolci-Instruct-SFT[2], which contains approximately 2M examples. We train for 500 steps with a batch size of 262,144 tokens and a peak learning rate of 10^{-5}, followed by cosine decay. Sequences are packed into contexts of up to 32,768 tokens. Appendix B gives the settings for the reasoning-focused model.

Validation Set Construction. We choose tasks in math, code, knowledge, and reasoning on which the base model performs above chance but below saturation. These criteria exclude GSM8K and GPQA. In preliminary selection experiments, reasoning-heavy tasks produced more downstream change than memorization-focused tasks such as standard MMLU and PopQA.

The validation queries comprise 7,500 MATH problems, the official MMLU-Pro validation set, the BBH evaluation demonstrations, and original MBPP problems disjoint from the MBPP+ evaluation set. The other validation-dependent corpus selectors in Table 1 use the same queries with their original reference responses. The MATH, MMLU-Pro, and BBH references include chain-of-thought (CoT) responses.

Table 6 reports the query and rollout counts for each task, together with rollout accuracy. MATH accounts for 7,500 of the 8,220 queries. The task normalization in Step 2 compensates for this imbalance by assigning each task equal total absolute rollout coefficients before length weighting; it does not equalize gradient magnitudes. Queries whose observed rollouts are all correct or all incorrect are retained (12.8% overall, mostly MATH and MMLU-Pro).

Table 6: Validation queries and on-policy rollouts for Olmo3-7B-Instruct-SFT, with 32 rollouts per query. The final two columns report the proportions of queries for which all sampled rollouts are correct or all are incorrect.

Task	Queries	Rollouts	Accuracy	All correct	All incorrect
MATH	7,500	240,000	49.1%	0.01%	10.0%
MBPP	596	19,072	31.1%	0.3%	38.9%
MMLU-Pro	70	2,240	32.0%	10.0%	58.6%
BBH	54	1,728	48.6%	3.7%	27.8%
Total	8,220	263,040	47.6%	0.15%	12.7%

Validation Rollout Generation. For Olmo3, we generate 32 responses per validation query using temperature $t = 0.6$ and top-p $p = 0.95$; OpenR1 uses eight responses per query (Appendix B). Writing this frozen decoder as q_{θ^*}, the rollout set is $Z_{\text{rollout}} = \{(x, y) \mid x \in X_{\text{val}},\ y \sim q_{\theta^*}(\cdot \mid x)\}$. Correctness is determined by released task-specific evaluators. For reasoning and knowledge tasks, we use the corresponding LightEval code when available; for code tasks, we use the standard official checker.

Attribution-Stage Hyperparameters. Table 7 lists the attribution-stage hyperparameters. We extract each example's token-mean CE gradient $\nabla_\theta \mathcal{L}_{CE}$ over assistant tokens and apply the GraSS sparse random projection (Hu et al., 2025). Corpus examples and validation rollouts use the

[1] Olmo3-7B-Instruct-SFT

[2] Dolci-Instruct-SFT

same projection matrix. By Eq. (3), the correctness-weighted mean of rollout CE gradients is $-\nabla_\theta \mathcal{J}_{PG}(\theta; x)$. Thus, the validation centroid in Eq. (4) is the negative projected gradient of the task-averaged PG objective. The response-length factor is already included in $\mathcal{L}_{CE}$. We differentiate ordinary model log-probabilities while holding the sampled responses and their correctness rewards fixed.

To construct the projection, we flatten gradients in lexicographic parameter-name order and divide them into chunks. Within each chunk, we uniformly sample up to $4d$ coordinates without replacement. Each selected coordinate is mapped to a uniformly sampled output index and assigned a sign of +1 or -1 with equal probability. Summing the projected chunks gives a vector of dimension $d = 16{,}384$. We apply no further rescaling or unit-norm normalization.

Table 7: Attribution-stage hyperparameters for both models.

	Olmo3-7B-Instruct-SFT	**OpenR1-Distill-7B**
Parameters whose gradients are used	all parameters	all except the embedding layer
Projection dimension d	16,384	16,384
Rollouts per validation query	32	8

Evaluation Configurations. BBH, MATH, and MMLU-Pro use greedy decoding. BBH accuracy is micro-averaged over 4,365 questions from its 18 multiple-choice subtasks. The remaining benchmarks use temperature $t = 0.6$ and top-p $p = 0.95$, with accuracy averaged over multiple sampled responses: avg@8 for MBPP+, avg@4 for ZebraLogic, LiveCodeBench v5, and OlympiadBench, and avg@16 for GPQA Diamond.

A.2 Baseline Implementations

BM25, DSIR, and RDS use the validation sets in Appendix A.1. We compute scores separately for each validation source and average the source-level scores.

Random. We sample 10% of the candidate SFT corpus uniformly without replacement.

Self-Distillation. We continue SFT directly on the verified correct on-policy responses. This compares training on generated text with using its gradients to select corpus examples. Self-Distillation + Random mixes the same verified rollouts 1:1 with an equal number of randomly sampled corpus examples. Both variants follow the same continual-SFT schedule.

BM25. We use the bm25s package (Lù, 2024). For each validation example, we rank corpus examples by BM25 similarity and map their ranks linearly to scores between 0 and 1. We sum these scores over validation examples within each source, then average across sources.

BM25 (signed rollouts). The scoring ablation in Section 5.2 uses the same 32 rollouts per validation query and correctness labels as DRIFT. Correct responses contribute positively to the aggregated BM25 retrieval score, and incorrect responses contribute negatively, with the same query and task normalization as DRIFT. The scoring mechanism remains lexical retrieval. We select the top 10% of the candidate corpus and keep that subset fixed across three continual-training seeds, using the training and evaluation settings in Appendix A.1.

DSIR. We instantiate the Data Selection with Importance Resampling (DSIR) framework (Xie et al., 2023b) with hashed n-gram features for tractable importance-weight estimation at scale. The framework maps the raw and target data into a feature space and resamples a subset of the raw data according to importance weights computed in this space.

RDS. We implement RDS+ (Ivison et al., 2025), which encodes samples using the hidden states of a pretrained causal language model and retrieves them by cosine similarity against validation examples. For each training sample, we aggregate the score by taking the maximum RDS similarity over the validation examples.

Qurating. We implement Qurating (Wettig et al., 2024), which scores each training example with a pretrained QuRater sequence-classification model. For each conversation, we concatenate all message contents, tokenize the resulting text, and split sequences that exceed the model's context window into fixed-length chunks. QuRater predicts a vector of scores for each chunk, and we aggregate these outputs into an example-level score by taking a length-weighted average. We use the officially released scoring tool; the final score is the average of the four intrinsic quality scores.

LESS. We reimplement LESS (Xia et al., 2024) within our framework. The original method uses LoRA warm-up training without long-sequence packing; our warm-up follows the continual-SFT configuration. LESS uses external reference responses and aggregates influence scores by maximum cosine similarity across subtasks. DRIFT instead uses an on-policy PG validation objective and sums task scores after log-space norm correction.

Reference-Based IF (GraSS). This baseline uses the same GraSS sparse-projection infrastructure as DRIFT (Hu et al., 2025), but scores corpus examples against the original reference responses using cosine similarity. It is labeled IF (GraSS) in the tables.

A.3 Robustness to Training and Evaluation Randomness

We repeat the continual-SFT stage of every Olmo3 method in Table 1, as well as the self-distillation controls, with three training seeds. Each method's selected subset remains fixed across these runs, so the reported variation covers continual training and evaluation, not selection. Sampling-based benchmarks use the decoding averages specified in Appendix A.1. Table 8 reports the mean and standard deviation for every entry. Across baselines, standard deviations range from 0.02 to 0.29 for All and from 0.04 to 0.42 for Additional.

Table 8: Olmo3-7B-Instruct-SFT results over three continual-training seeds, reported as mean ± standard deviation. Each method uses the same selected subset across seeds. Sampling-based benchmarks also average over multiple decoding runs (Appendix A.1), so the deviations reflect both training and evaluation randomness. Table 1 reports the means for corpus selectors. S.D. denotes Self-Distillation; Rand. denotes Random.

	Validation-Source Benchmarks				Additional Benchmarks				Average		
↑Acc.(%)	**BBH**	**MBPP+**	**MATH**	**MMLU-Pro**	**ZebraLogic**	**LCB v5**	**OlympiadBench**	**GPQA-D**	**Val.-src.**	**Additional**	**All**
Olmo3-7B-Instruct-SFT	44.63	56.42	66.16	44.22	17.23	15.43	34.27	35.01	52.86	25.49	39.17
Rand.	44.45±0.29	56.34±0.17	66.40±0.28	44.25±0.15	17.67±0.96	14.38±0.63	35.04±0.47	33.77±0.68	52.86±0.14	25.21±0.16	39.04±0.15
S.D.	43.18±0.68	54.77±0.41	67.90±0.36	41.28±0.33	17.76±0.23	8.92±0.52	35.99±0.38	31.29±0.31	51.78±0.17	23.49±0.11	37.64±0.13
S.D. + Rand.	43.73±0.75	55.95±0.26	67.91±0.27	42.64±0.45	17.85±0.22	9.68±0.43	36.20±0.30	33.00±0.97	52.56±0.27	24.18±0.24	38.37±0.22
DSIR	39.97±0.23	52.63±0.88	55.11±0.42	36.22±0.82	17.68±0.34	12.73±0.68	27.83±1.07	32.82±0.70	45.98±0.09	22.76±0.04	34.37±0.02
Qurating	44.27±0.21	55.96±0.05	66.89±0.37	44.14±0.16	17.77±0.58	15.49±0.94	34.98±0.26	34.61±0.55	52.82±0.17	25.71±0.31	39.26±0.23
BM25	44.44±0.40	55.74±0.99	67.82±0.27	44.66±0.14	17.48±0.51	15.13±1.04	36.18±0.64	34.90±1.36	53.16±0.12	25.92±0.42	39.54±0.19
LESS	43.38±0.21	53.80±0.20	64.60±0.40	43.38±0.14	16.14±0.90	13.45±0.70	33.06±0.63	33.21±0.51	51.29±0.20	23.96±0.40	37.63±0.29
RDS	43.68±0.13	55.59±0.24	66.56±0.39	44.80±0.18	17.76±1.09	14.59±0.84	35.38±0.55	34.27±0.67	52.66±0.21	25.50±0.29	39.08±0.24
IF (GraSS)	44.58±0.35	54.97±0.28	65.69±0.26	44.81±0.21	16.98±0.35	13.67±0.78	34.21±0.17	34.35±0.41	52.51±0.21	24.80±0.17	38.66±0.08
DRIFT (Ours)	45.86±0.30	56.40±0.22	67.84±0.43	44.94±0.19	19.95±0.94	16.28±0.58	35.25±0.56	34.57±0.92	53.76±0.13	26.51±0.24	40.14±0.12

Selection-stage randomness. Response sampling and random projection introduce additional variability that fixed-subset runs do not capture. The top-10% selections have Jaccard overlaps of 0.73 across rollout seeds and 0.69 across projection seeds. For comparison, two independent random 10% subsets have an approximate overlap of 0.053, while DRIFT and BM25 overlap at 0.24. Three additional downstream runs achieve All scores of 40.03, 40.19, and 40.08, each above the strongest baseline's reported average of 39.54.

Rollout-utilization ablations. In the 1P and 1P+1N settings, we retain at most one correct rollout per query; a query contributes no correct rollout when none is available. We normalize the retained weights within each task after selection, giving the available correct rollouts greater weight when fewer are retained.

Scoring ablation. Table 11 reports per-benchmark results for BM25 with on-policy responses and signed weighting, alongside the original BM25 and DRIFT results. We calculate All, validation-source, and Additional averages within each run, then report their means and standard deviations across the three training seeds.

Table 9: Per-benchmark results for the validation-objective and rollout-budget ablations in Table 2, reported as mean ± standard deviation over three continual-training seeds on Olmo3-7B-Instruct-SFT. Off-policy uses reference-based CE. The remaining rows use the indicated numbers of positive (P) and negative (N) on-policy responses, with incorrect-response weight r_{neg}. All configurations use per-task orthogonalization. The 32-rollout PG configuration is the DRIFT result in Table 8.

	Validation-Source Benchmarks				Additional Benchmarks				Average		
↑Acc.(%)	BBH	MBPP+	MATH	MMLU-Pro	ZebraLogic	LCB v5	OlympiadBench	GPQA-D	Val.-src.	Additional	All
Off-policy reference	43.84±0.28	55.06±0.29	66.47±0.29	44.93±0.26	16.93±0.51	13.73±0.79	34.35±0.29	33.66±0.35	52.58±0.24	24.67±0.20	38.62±0.11
1P	44.51±0.23	56.54±0.32	67.08±0.42	45.14±0.15	18.06±0.88	15.39±0.51	35.05±0.62	34.50±0.82	53.32±0.18	25.75±0.18	39.53±0.16
1P + 1N ($r_{\text{neg}} = -1$)	44.92±0.34	56.63±0.74	67.56±0.19	45.66±0.28	19.46±0.96	15.85±0.75	34.73±0.60	34.96±0.38	53.69±0.27	26.25±0.26	39.97±0.25
All P	45.24±0.17	56.89±0.56	67.90±0.46	43.84±0.22	17.73±0.69	13.22±0.51	36.34±0.13	35.04±1.72	53.47±0.19	25.58±0.33	39.53±0.21
All P + All N, 4 rollouts	45.17±0.17	56.73±0.48	67.72±0.40	45.35±0.19	18.62±0.52	15.80±0.67	35.07±0.27	34.54±0.47	53.74±0.14	26.01±0.13	39.87±0.13
All P + All N, 8 rollouts	45.61±0.08	56.02±0.55	67.56±0.36	45.18±0.30	19.71±0.79	16.01±0.76	35.73±0.50	34.12±0.56	53.59±0.10	26.40±0.10	39.99±0.02
All P + All N, 16 rollouts	45.47±0.16	56.24±0.35	67.43±0.41	45.04±0.14	18.37±0.56	15.45±0.86	35.91±0.56	35.65±1.22	53.54±0.01	26.35±0.25	39.95±0.12
All P + All N, 32 rollouts (DRIFT)	45.86±0.30	56.40±0.22	67.84±0.43	44.94±0.19	19.95±0.94	16.28±0.58	35.25±0.56	34.57±0.92	53.76±0.13	26.51±0.24	40.14±0.12
All P + All N, $r_{\text{neg}} = +1$	44.02±0.12	56.81±0.45	66.85±0.14	43.80±0.30	17.17±0.91	13.26±0.60	36.02±0.47	34.31±0.36	52.87±0.21	25.19±0.16	39.03±0.18

Table 10: Per-benchmark results for the gradient-debiasing ablation in Table 4, reported as mean ± standard deviation over three continual-training seeds on Olmo3-7B-Instruct-SFT. All variants use the same on-policy PG objective $\mathcal{J}_{PG}$ and differ only in norm correction. Per-Task Orth. is the DRIFT result in Table 8.

	Validation-Source Benchmarks				Additional Benchmarks				Average		
↑Acc.(%)	BBH	MBPP+	MATH	MMLU-Pro	ZebraLogic	LCB v5	OlympiadBench	GPQA-D	Val.-src.	Additional	All
No Debiasing	43.65±0.20	55.94±1.01	65.47±0.17	44.61±0.06	19.55±0.65	16.10±0.71	34.67±0.28	35.44±0.40	52.42±0.23	26.44±0.08	39.43±0.15
Cosine	44.70±0.34	56.32±0.35	67.11±0.35	45.12±0.29	20.86±0.70	15.53±0.67	35.92±0.21	33.81±0.35	53.31±0.18	26.53±0.12	39.92±0.06
Packing Correction	44.01±0.43	56.07±0.39	66.61±0.17	44.79±0.27	20.87±1.10	16.52±0.77	35.24±0.39	34.04±0.16	52.87±0.30	26.67±0.04	39.77±0.17
Unified Orth.	44.98±0.41	56.73±0.62	67.53±0.40	45.20±0.05	19.39±0.69	15.33±0.85	36.58±0.31	33.72±0.58	53.61±0.15	26.26±0.10	39.93±0.04
Per-Task Orth. (DRIFT)	45.86±0.30	56.40±0.22	67.84±0.43	44.94±0.19	19.95±0.94	16.28±0.58	35.25±0.56	34.57±0.92	53.76±0.13	26.51±0.24	40.14±0.12

A.4 Selection Ratio Sweep

Figure 3 varies the selected fraction from 5% to 25% on Olmo3 under two schedules: a fixed budget of 500 steps, and a budget scaled with the selected fraction. DRIFT exceeds BM25 at every tested ratio under both schedules, with margins of 0.1–0.7 points on All. Its highest scores occur between 5% and 15%; scores at 20% and 25% are 0.2–0.4 points lower. Scaling the budget changes DRIFT's score by at most 0.2 points at a given ratio, suggesting limited benefit from extra training steps over this range. The default ratio of 10% was fixed before the sweep. All other ratios are single-run results.

A.5 Compute Cost

Table 12 reports wall-clock costs on a single node with eight A800 GPUs, separating reusable corpus preparation from work repeated for each validation set. IF, LESS, and DRIFT each require a 12-hour corpus-gradient extraction and projection pass, producing 63 GB of features. These features

Table 11: Per-benchmark scoring ablation on Olmo3-7B-Instruct-SFT, reported as mean ± standard deviation over three continual-training seeds. BM25 (signed rollouts) uses on-policy responses with signed correctness weighting. BM25 and DRIFT results are repeated from Table 8.

Benchmark	BM25	BM25 (signed rollouts)	DRIFT
BBH	44.44±0.40	44.73±0.20	45.86±0.30
MBPP+	55.74±0.99	56.50±0.23	56.40±0.22
MATH	67.82±0.27	66.92±0.36	67.84±0.43
MMLU-Pro	44.66±0.14	43.85±0.27	44.94±0.19
ZebraLogic	17.48±0.51	17.54±0.17	19.95±0.94
LCB v5	15.13±1.04	14.40±0.19	16.28±0.58
OlympiadBench	36.18±0.64	35.63±0.36	35.25±0.56
GPQA-D	34.90±1.36	34.84±1.39	34.57±0.92
Validation-source average	53.16±0.12	53.00±0.10	53.76±0.13
Additional average	25.92±0.42	25.60±0.39	26.51±0.24
All average	39.54±0.19	39.30±0.15	40.14±0.12

remain reusable while the checkpoint, corpus, parameter subset, and projection configuration are unchanged. New validation queries or response constructions require new responses and projected validation gradients, followed by scoring against the stored corpus features.

Generating 32 rollouts per query takes 8 hours with vLLM 0.11.0. Linear scaling gives estimated generation times of 2 hours for eight rollouts and 1 hour for four. The eight-rollout variant reaches 39.99 on All, 0.15 points below the 32-rollout result (Table 2), at an estimated total cost of 21 hours.

BM25 requires 4 hours per validation set. RDS requires 11 hours per validation set in addition to 8–11 hours of indexing. At a similar reported total elapsed-time budget to the eight-rollout DRIFT variant, BM25 with 2.5 times as much data and 2.5 times as many training steps reaches 39.66 (Figure 3).

Table 12: Wall-clock costs for Olmo3-7B-Instruct-SFT, in hours on one node with eight A800 GPUs. Corpus preparation is reusable; projected features require the checkpoint, corpus, parameter subset, and projection configuration to remain fixed. Validation-stage work is repeated when queries or response construction change. A dagger (†) marks work that is reusable in principle but was recomputed here; "est." marks estimates. Generation times of one and two hours, and totals based on them, are linearly scaled estimates. Totals containing stages shorter than 0.5 hours are rounded upper approximations. Rows below the rule compare approximately matched elapsed-time budgets (Section 5.4). S.D. denotes Self-Distillation; Rand. denotes Random.

Method	Per corpus	Per val. set	Continual SFT	Total (h)	Storage	All
Rand.	–	–	6.6	6.6	–	39.04
BM25	0.2†	4.0	6.6	10.8	–	39.54
DSIR	0.8†	0.9	6.6	8.3	–	34.37
Qurating	≈1 (est.)	–	6.6	7.6	–	39.26
RDS	8–11	10.9	6.6	26–29	–	39.08
S.D.	–	8	6.6	14.6	–	37.64
S.D. + Rand.	–	8	6.6	14.6	–	38.37
IF (GraSS)	12	<0.5	6.6	19.1	63 GB	38.66
LESS	6.6 (warm-up, est.) + 12	<0.5	6.6	25.7	63 GB	37.63
DRIFT (32 rollouts)	12	8 + <0.5	6.6	27.1	63 GB	**40.14**
DRIFT, 8 rollouts	12	2 + <0.5	6.6	21.1	63 GB	39.99
DRIFT, 4 rollouts	12	1 + <0.5	6.6	20.1	63 GB	39.87
BM25, 25% data, 1,250 steps	0.2†	4.0	16.5	20.7	–	39.66

A.6 Further Results on Self-Distillation

Self-distillation improves MATH from 66.16% to 67.90%, but reduces All by 1.53 points. Its largest losses are on LiveCodeBench v5 (15.43% to 8.92%) and GPQA Diamond (35.01% to 31.29%). Mixing the verified rollouts 1:1 with random corpus examples recovers part of the loss, reaching 38.37 on All, but remains 0.67 points below Random; LiveCodeBench reaches only 9.68.

The two approaches make different use of the same validation queries. Self-distillation trains on verified correct responses generated from a pool of 8,220 queries, of which 7,500 are from MATH and 720 from the other tasks. DRIFT instead uses the rollout gradients to select examples from a corpus of approximately 2M examples. In Table 8, DRIFT improves both All and Additional, whereas direct rollout training lowers both averages.

B Results on a Reasoning-Focused Model

We also evaluate DRIFT on OpenR1-Distill-7B[3], a reasoning-focused model distilled from long chain-of-thought traces. Its SFT corpus, Mixture-of-Thoughts[4], contains 349k examples. Starting from the checkpoint after its original SFT schedule, we select 10% of the corpus and continue training with the settings in Appendix A.1, increasing the training duration to 1,000 steps to account for longer responses. We generate eight rollouts per validation query rather than 32, also because of their greater length.

[3]OpenR1-Distill-7B

[4]Mixture-of-Thoughts

The validation queries, evaluation benchmarks, and baselines are the same as for Olmo3, except that Self-Distillation + Random is omitted. Each result in this appendix is a single continual-training run. This model is also included in the diagnostics in Figure 2 and Table 14.

Table 13: Per-benchmark continual-SFT results on OpenR1-Distill-7B, including the self-distillation control. The aggregate results for corpus selectors are also reported in Table 1. Each continual-training entry is a single run. Bold indicates the best continual-training result; colored differences are relative to the starting checkpoint. S.D. denotes Self-Distillation; Rand. denotes Random.

	Validation-Source Benchmarks				Additional Benchmarks				Average	
↑Acc.(%)	BBH	MBPP+	MATH	MMLU Pro	Zebra Logic	LCB v5	Olympiad Bench	GPQA Diamond	All	Additional
OpenR1-Distill-7B	76.04	54.23	90.98	51.65	23.85	43.86	67.51	49.72	57.23	46.24
+ Rand.	76.52	53.90	91.05	51.15	23.75	44.81	66.58	**51.45**	57.40 +0.17	46.65 +0.41
+ S.D.	60.82	55.85	83.78	48.59	20.53	20.31	60.24	44.48	49.33 -7.90	36.39 -9.85
+ DSIR	73.86	54.17	90.85	49.34	22.30	44.46	65.95	49.02	56.24 -0.99	45.43 -0.81
+ Qurating	**76.70**	54.33	**91.88**	51.16	24.73	47.63	68.25	47.57	57.78 +0.55	47.05 +0.81
+ BM25	75.76	54.23	91.20	51.19	24.95	44.19	67.62	49.40	57.32 +0.09	46.54 +0.30
+ LESS	76.15	54.17	91.76	50.44	24.30	44.10	**68.32**	49.02	57.28 +0.05	46.44 +0.20
+ RDS	75.88	54.86	91.00	**51.69**	25.48	45.19	66.77	50.19	57.63 +0.40	46.91 +0.67
+ IF (GraSS)	75.46	54.27	91.52	50.78	25.80	48.18	67.62	49.15	57.85 +0.62	47.69 +1.45
+ **DRIFT (Ours)**	76.11	**56.02**	90.82	51.37	**25.88**	**50.21**	68.29	50.54	**58.66** +1.43	**48.73** +2.49

DRIFT achieves the highest All and Additional averages in Table 13, improving the starting model by 1.43 and 2.49 points, respectively. Its All score exceeds the strongest baseline, reference-based IF, by 0.81 points, compared with the 0.60-point margin over BM25 on Olmo3. The largest gains occur on MBPP+, LiveCodeBench v5, and ZebraLogic. MATH decreases slightly, from 90.98 to 90.82, as does MMLU-Pro, from 51.65 to 51.37. Self-distillation reduces All from 57.23 to 49.33 in this run.

C Qualitative Analysis

Figure 5 complements the DRIFT case in Figure 4 with the highest-ranked training examples retrieved by three baselines for the same query about continuity of a piecewise function. DRIFT retrieves a polynomial-inequality problem, BM25 retrieves another problem involving continuity, RDS retrieves a quadratic-inequality problem, and LESS retrieves a code sample. The DRIFT example shares algebraic operations with the validation query despite differences in wording and context. This comparison illustrates how the retrieved content differs across methods; it does not establish which reasoning features determine the attribution score.

D Policy-Gradient Attribution and Locality

This appendix separates the population PG identities from the surrogate used in our implementation. We first prove Proposition 1, derive the validation-step bound, and explain how decoding and token averaging change the objective. We then present centroid and parameter-space diagnostics, followed by local quadratic models of proximity and non-convergence. The attribution derivations concern raw scores before norm correction.

D.1 Proof of Proposition 1

Setting and proof. Assume that responses are sampled exactly from the differentiable model distribution π_{θ^*}. Proposition 1 uses the full-sequence loss, $T(y)\mathcal{L}_{CE}(x, y; \theta) = -\log \pi_\theta(y \mid x)$. The score-function identity follows by differentiating each response probability. Multiplying by any fixed response weight and summing proves the expectation identity in Proposition 1. For the correctness reward in Section 3,

$$\begin{aligned} \mathbb{E}_{y\sim\pi_{\theta^*}}\left[r(y)\nabla_\theta \log \pi_{\theta^*}(y \mid x)\right] &= \sum_{y\in\mathcal{Y}} r(y)\ \nabla_\theta \pi_\theta(y \mid x)|_{\theta^*} \\ &= \nabla_\theta \mathbb{E}_{y\sim\pi_\theta}[r(y)]|_{\theta^*}\,. \end{aligned} \tag{6}$$

Qualitative Example: Baseline Retrievals

[Top-1 Retrieved Training Sample by RDS]
Prompt: For what range of positive real values of c does the inequality $x^2 - 6x + c < 0$ have real roots for x? Express your answer using interval notation.
Response: For real roots, the discriminant condition $b^2 - 4ac \geq 0$ with $a = 1, b = -6$ yields $36 - 4c \geq 0 \Rightarrow c \leq 9$... Combining constraints, the range of positive real values of c is $c \in (0, 9)$.

[Top-1 Retrieved Training Sample by BM25]
Prompt: A high school teacher models home tranquility with mathematical functions... 1. Define a function $f(x)$ that is continuous and differentiable over $[-2, 2]$ and satisfies: $f(0) = 0$, $f'(x) = f(x)^2 - 1$, and $f(x)$ has exactly one local maximum and one local minimum within the interval...
Response: To solve the problem, given the differential equation $f'(x) = f(x)^2 - 1$ with $f(0) = 0$, solving yields $f(x) = \frac{1-e^{2x}}{1+e^{2x}}$... For the piecewise function $g(x)$, continuity at $x = 0$ requires $f(0) = b$, giving $b = 0$... so $a = -1$.

[Top-1 Retrieved Training Sample by LESS]
Prompt: Given an integer N ($1 \leq N \leq 10\,000$), output the N-th digit of π without the decimal point (314159...)... Input: A single line containing N. Output: Print the N-th digit...
Response:
```
def main():  import sys; pi_str =
("31415926535897932384 6..."); N = int(sys.stdin.readline().strip());
print(pi_str[N - 1])
```

Figure 5: Highest-ranked training examples retrieved by RDS, BM25, and LESS for the validation query in Figure 4.

For bounded-length sequences, the sum is finite, so differentiation and summation commute. Constant weights reduce the sum to a constant multiple of the total probability, whose gradient is zero.

Corollary 2 (Special Cases Under Exact Sampling and Sequence Loss) *Constant weighting gives a zero expected full-sequence CE anchor and a zero expected raw attribution score. For correctness rewards $r = 2C - 1$, where C indicates a correct response, the expected weighted full-sequence CE anchor is*

$$-\nabla_\theta \mathbb{E}_{\pi_\theta}[r]\big|_{\theta^*} = -2\,\nabla_\theta \Pr_{\pi_\theta}(C = 1 \mid x)\Big|_{\theta^*}.$$

This population anchor vanishes at a stationary point of expected correctness.

Bounding a validation-gradient step. Fix a query x and retain the exact-sampling, full-sequence assumptions. Let

$$s(y) = \nabla_\theta \log \pi_{\theta^*}(y \mid x), \qquad g_w = \mathbb{E}[w(y)s(y)], \qquad G_x^2 = \mathbb{E}\|s(y)\|_2^2 < \infty,$$

where all expectations are under $\pi_{\theta^*}(\cdot \mid x)$. For $\Delta\theta_{\text{val}} = \eta g_w$, with $\eta \geq 0$, the validation-gradient step satisfies the bound in Proposition 1.

The score identity gives $\mathbb{E}[s(y)] = 0$, so subtracting the mean weight does not change g_w: only variation in the weights contributes to the expected gradient. With $\bar{w} = \mathbb{E}[w(y)]$, Cauchy–Schwarz yields

$$\begin{aligned}
\|g_w\|_2 &= \|\mathbb{E}[(w(y) - \bar{w})s(y)]\|_2 \\
&\leq \mathbb{E}[|w(y) - \bar{w}|\,\|s(y)\|_2] \\
&\leq \sqrt{\mathbb{E}[(w(y) - \bar{w})^2]}\sqrt{\mathbb{E}\|s(y)\|_2^2} \\
&= G_x\sqrt{\mathrm{Var}[w(y)]}.
\end{aligned}$$

Multiplying by $\eta \geq 0$ proves the bound in Proposition 1 for the gradient-ascent step $\Delta\theta_{\text{val}} = \eta g_w$. For $w = 2C - 1$ and $p_x = \Pr(C = 1 \mid x)$, the variance is $4p_x(1 - p_x)$, giving

$$\|\Delta\theta_{\text{val}}\|_2 \leq 2\eta G_x \sqrt{p_x(1 - p_x)}.$$

The bound is zero for constant weights and is controlled by reward variation, the model's gradient scale, and the step size. External references have no corresponding cancellation identity. This bound applies to a population validation-gradient step. It does not imply that its norm is smaller than every external-reference gradient, nor does it bound displacement during training on a selected subset. That displacement also depends on the corpus gradients, selection rule, and optimization procedure.

The implemented PG surrogate. Equation (3) defines the empirical objective $\mathcal{J}_{PG}(\theta; x)$ using token-mean CE losses $\mathcal{L}_{CE}$ on sampled responses. Let $\mathcal{R}_x$ denote the random rollout pool for query x, drawn from the frozen temperature/top-p decoder q_{θ^*}. Averaging over rollout sampling gives

$$\mathbb{E}_{\mathcal{R}_x}\left[\mathcal{J}_{PG}(\theta; x)\right] = \mathbb{E}_{y \sim q_{\theta^*}(\cdot|x)}\left[\frac{r(y)}{T(y)} \log \pi_\theta(y \mid x)\right].$$

Two implementation choices distinguish this surrogate from the preceding population identities. First, differentiation is applied to $\log \pi_\theta$ while the sampled responses are held fixed. Since sampling uses q_{θ^*} rather than π_{θ^*}, the expected gradient is not, in general, the exact gradient of expected correctness under either decoder. Second, token averaging changes the objective even under exact model sampling:

$$\mathbb{E}_\pi\left[\frac{r \nabla \log \pi}{T}\right] = \nabla \mathbb{E}_\pi\left[\frac{r}{T}\right].$$

With constant unit rewards, the corresponding quantity is $\nabla \mathbb{E}_\pi[1/T]$, which need not vanish. The zero-anchor result for constant weighting therefore does not apply to the implemented uniform-weight control. For a fixed rollout pool, define the task-averaged objective

$$\overline{\mathcal{J}}^{(k)}_{PG}(\theta) = \frac{1}{|D_k|} \sum_{x \in D_k} \mathcal{J}_{PG}(\theta; x).$$

The centroid in Eq. (4) is consequently the negative projected gradient $-P\nabla_\theta \overline{\mathcal{J}}^{(k)}_{PG}(\theta)\Big|_{\theta^*}$. Combining the negative signs in the training CE gradient and the centroid gives

$$S^{(k)}_{\text{raw}}(z_{\text{train}}) = \left\langle \frac{P\nabla_\theta \log \pi_{\theta^*}(y_{\text{train}} \mid x_{\text{train}})}{T(y_{\text{train}})},\, P\nabla_\theta \overline{\mathcal{J}}^{(k)}_{PG}(\theta)\Big|_{\theta^*} \right\rangle. \tag{7}$$

When $P = I$, a small descent step on the training loss $\mathcal{L}_{CE}$ changes $\overline{\mathcal{J}}^{(k)}_{PG}$, to first order, by the step size times this inner product. A positive raw score therefore predicts an increase in the PG objective, consistent with Step 2. We evaluate the subsequent norm correction and subset training empirically.

Conditioning on correctness. Under exact model sampling and full-sequence loss, consider a query with $p_x = \Pr_{\pi_\theta}(C = 1 \mid x)$ and $0 < p_x < 1$. Then

$$\mathbb{E}[\nabla \log \pi \mid C = 1] = \frac{\nabla p_x}{p_x}, \qquad \mathbb{E}[\nabla \log \pi \mid C = 0] = -\frac{\nabla p_x}{1 - p_x}, \qquad \mathbb{E}[r \nabla \log \pi] = 2\nabla p_x.$$

For this query, positive-only and signed weighting therefore produce expected log-likelihood gradients in the same direction, with scales $1/p_x$ and 2, respectively. CE anchors have the opposite sign. Because p_x varies across queries, these weighting schemes can produce different aggregate directions. Token averaging and the implemented decoder further change the relationship.

These are population identities, not general variance-reduction guarantees. Conditioning on a finite rollout pool in which every response is correct or every response is incorrect also changes the expectation; the resulting conditional expected gradient need not be zero.

D.2 Diagnostics for the Locality Intuition

The all-+1 condition in Table 2 corresponds to On-Policy w/o Weighting in Figure 2 and Table 14.

Validation-centroid norms. In addition to the parameter-space curves in Figure 2, we measure the norm of the aggregated validation centroid in projected-gradient space:

$$c = \sum_{z_{\text{val}}} \alpha(z_{\text{val}})\phi(z_{\text{val}}; \theta^*), \tag{8}$$

where $\alpha(z_{\text{val}})$ is the weight assigned to each validation target. Both on-policy constructions yield smaller centroid norms than the external references (Table 14). The signed centroids are approximately 60 times smaller on Olmo3 and 42 times smaller on OpenR1.

A small centroid norm can reflect cancellation across responses and queries. It does not, by itself, establish stationarity of expected correctness or bound parameter drift during training on the selected data.

Table 14: Projected validation-centroid norms for external references and two on-policy weighting schemes.

Model	Off-Policy	On-Policy w/o Weighting	On-Policy w/ Weighting (Ours)
Olmo3	5.495	0.123	0.092
OpenR1	0.337	0.007	0.008

Parameter drift. At training step t, drift is $\theta_t - \theta^*$. We measure its L2 norm and its sparsity, defined as the fraction of parameters whose absolute change is below 10^{-5}. Figure 2 plots each metric relative to continual SFT on a random 10% subset at the same step, labeled Uniform (Baseline). On-Policy w/o Weighting is a separate selection condition that assigns equal weights to rollout gradients.

After the initial steps, the on-policy selections produce smaller L2 changes than the off-policy selection. Their changes remain close to the random baseline in this relative metric, although they can exceed it. Sparsity thresholds of 10^{-6}, 10^{-7}, and 10^{-8} yield the same qualitative trend.

D.3 Quadratic Proximity Model

To illustrate the effect of a proximity restriction, we add a quadratic penalty to the perturbed empirical risk:

$$\theta_{\text{PBRF}}(\epsilon) = \arg\min_{\theta} \left[\frac{1}{n} \sum_{i=1}^{n} \mathcal{L}(z_i; \theta) + \epsilon \mathcal{L}(z_{\text{train}}; \theta) + \frac{1}{2\tau} \|\theta - \theta^*\|^2_{H_{\theta^*}} \right]. \tag{9}$$

Here, τ controls the neighborhood size. This quadratic model is a simplification of the proximity penalty. At a stationary θ^*, with $H_{\theta^*} \succ 0$ and $\tau > 0$, the first-order influence on validation loss is $\frac{\tau}{\tau+1}\mathcal{I}(z_{\text{train}}, z_{\text{val}})$.

D.4 The Non-Convergence Gap

Classical IF assumes a stationary checkpoint, where the empirical gradient

$$g = \frac{1}{n} \sum_{i=1}^{n} \nabla_\theta \mathcal{L}(z_i; \theta^*)$$

is zero. When $g \neq 0$, further optimization moves the model even if the data distribution is unchanged. Under a local quadratic approximation with positive-definite H_{θ^*}, this background displacement is $-H_{\theta^*}^{-1} g$. Its magnitude, $\|H_{\theta^*}^{-1} g\|$, can obscure the additional displacement caused by up-weighting or removing a particular example. The **non-convergence gap** thus arises because retraining reflects unfinished optimization as well as the data intervention that attribution is intended to measure (Bae et al., 2022).

This provides a theoretical motivation for computing attribution after the original SFT schedule, in addition to the practical motivation for checkpoint refinement. In the analysis, we treat the completed checkpoint as approximately stationary and ask which examples can improve it further.

E Method Details

Gradient debiasing. Let g be a projected training gradient with norm ρ, and let c_k be a fixed task centroid. The raw score magnitude is

$$|S_{\text{raw}}^{(k)}| = \rho \|c_k\| \, |\cos \alpha_k|,$$

where α_k is the angle between the two vectors. Cosine similarity removes the explicit multiplicative norms. A fitted power-law slope different from one indicates that gradient norm and direction are associated in the observed examples.

Our ordinary least-squares regression includes an intercept and is defined for positive gradient norms and nonzero score magnitudes. Subtracting $\beta^{(k)} \log \rho$ removes the fitted linear association between log score magnitude and log gradient norm on the fitting sample. It does not imply statistical independence or identify a causal effect of norm. Table 15 reports the fitted slopes. On Olmo3, they range from 2.00 to 2.30 across tasks, motivating task-specific rather than fixed norm correction.

Table 15: Power-law slopes β fitted in log space, by model, validation task, and response source.

Validation Task	Olmo3-7B-Instruct-SFT		OpenR1-Distill-7B	
	Off-Policy	On-Policy	Off-Policy	On-Policy
MATH	1.59	2.21	1.17	0.92
BBH	1.51	2.05	1.22	1.24
MBPP	1.47	2.30	1.14	0.98
MMLU-Pro	1.48	2.00	1.22	1.16

The correction preserves the sign of each task's raw score. With task-specific slopes, the combined score can be written as

$$S(g) = g^\top \sum_k \rho^{-\beta^{(k)}} c_k.$$

The combined direction therefore depends on the candidate example's norm. The final score is a ranking transformation, not, in general, an inner product with a single fixed validation gradient.

F Limitations

Refinement as a Proxy for Distribution Quality. Our broader objective is to improve the performance attainable through SFT, but our experiments evaluate distributions by continuing training from a completed checkpoint. We do not repeat full SFT under each distribution. A gain in this protocol does not establish the same gain when the distribution is used throughout SFT, nor does it identify an optimal distribution for a full training run.

Scope of Influence Approximations. We evaluate DRIFT with GraSS sparse projections. Whether on-policy validation provides similar gains with structured Hessian approximations such as EK-FAC, or with dense first-order gradient products, remains untested.

Theoretical Origins of Gradient Norm Bias. The relationship between raw score magnitude and gradient norm differs across tasks. Our correction removes a fitted linear association in log space, and its value is assessed through downstream performance. We do not yet explain why the slopes differ or when the correction will improve selection on a new task.

Scope of the Theoretical Analysis. Proposition 1 assumes exact model sampling and full-sequence log-likelihood gradients. The implemented decoder and token-mean loss instead define the surrogate objective $\mathcal{J}_{PG}$ in Eq. (3). The analysis does not bound the error of projected, norm-corrected scores as estimates of retraining effects, the parameter drift during continual SFT, finite-sample variance, or higher-order terms. Figure 2 and Table 14 provide parameter-drift and centroid diagnostics consistent with the locality motivation; they do not establish that the proximity gap is reduced or eliminated.

Model Coverage. The main experiments and all seed and sensitivity analyses use one 7B instruction model. The reasoning-focused model in Appendix B follows the same protocol and again places DRIFT first on the aggregate metrics, but each result is a single run. Larger models and broader coverage of model families remain untested, so the extent to which the gains transfer to those settings is unknown.

Completeness of Anomaly Filtering. Per-task correction lowers the ranks of the examples in Tables 5 and 16, but some malformed or noisy responses still receive high scores. Gradient norm alone is not a general criterion for identifying such examples.

G More Qualitative Cases

Table 16 complements the main-text example in Table 5 with two further malformed training examples. Per-task correction places both outside the top-10% selection cutoff.

Table 16: Percentile ranks of two additional malformed training examples with large gradient norms under different score corrections.

Ill-formed Training Sample	
Prompt: Which condiment was used as medicine during the 1830s?	
Response: Ketchup.wdgywdufqtufdufwytdfwfdytfwtydvutwqfdytfw	
Method Rankings (Percentile)	
No Debiasing: Top 0.037%	Packing Correction: Top 0.008%
Cosine: Top 2.779%	**Per-Task Orth.: Top 33.994%**

Ill-formed Training Sample	
Prompt: 4: Reer Baniqureeda goortii ay duldageen Saxaabada oo ay u timid Cabsi daran Ninkii la dhihijiray Kacab ibnu Asad oo ahaa Madaxdoodii ayaa wuu u soo jeediyay 3 arimood oo uu yiraahday mid kaqaato sadaxdaas, ee sheeg Sadaxdaas? (**Translation**: When the Quraysh realized that the Companions had arrived in Abyssinia and were under protection, they felt great anxiety. Their leader, Amr ibn al-As, proposed three arguments to the King and asked him to accept one of them. What were those three?)	
Response: 14: ....?	
Method Rankings (Percentile)	
No Debiasing: Top 0.056%	Packing Correction: Top 7.107%
Cosine: Top 2.177%	**Per-Task Orth.: Top 37.743%**

H Extended Related Work

Data curation for supervised fine-tuning. Data-scoring methods differ in the signals used to value an example. Heuristic filters rely on text-level rules (Wenzek et al., 2020), whereas model-guided filters use learned quality or difficulty estimates (Li et al., 2024a; Wettig et al., 2024). Meta-raters and sample-level mixing methods learn scores or weights for pretraining (Zhuang et al., 2025b; Xi et al., 2025). Other approaches combine influence with cluster-level diversity (Zhang et al., 2025a), use curvature-aware influence for CLIP adaptation (Zhuang et al., 2025a), or select subsets for instruction tuning (Xia et al., 2024; Zhang et al., 2025b; Humane et al., 2025).

Several methods define data value through the model's current learning needs. RHO-LOSS ranks examples by the loss gap between the current model and a reference model trained on held-out data (Mindermann et al., 2022); Rho-1 applies this idea at the token level (Lin et al., 2024b). DRIFT likewise uses a model-dependent signal, but defines it through the correctness-weighted on-policy PG objective $\mathcal{J}_{PG}$. It then scores corpus examples by their estimated effect on that objective.

Generated-response methods make a different use of the model's outputs. Rejection sampling, self-distillation, and on-policy distillation train directly on generated text (Singh et al., 2023; Xiong et al., 2025; Agarwal et al., 2024; Lu, 2025). In DRIFT, generated responses define $\mathcal{J}_{PG}$ for selecting examples from the original corpus, and continued training minimizes $\mathcal{L}_{CE}$ on those examples. The validation queries can therefore guide selection beyond their own response set.

Influence functions and data attribution. Influence functions estimate how individual training examples affect a model's validation loss (Koh & Liang, 2017). Practical approximations for neural networks include structured Hessian estimates (Grosse et al., 2023; Kwon et al., 2023; Coalson et al., 2025), random projections (Park et al., 2023; Hu et al., 2025), and Hessian-free gradient products (Xia et al., 2024).

Gradient-based selection predates LLM instruction tuning. GLISTER and GradMatch select subsets whose aggregate gradients match a validation gradient (Killamsetty et al., 2021b;a), and GREATS applies a related first-order criterion online to LLM training batches (Wang et al., 2024a). TRAK (Park et al., 2023) and DataModels (Ilyas et al., 2022) predict model behavior from training data. DsDm and MATES estimate target-task effects using datamodels or a learned influence proxy (Engstrom et al., 2024; Yu et al., 2024). CROPI (Zhu et al., 2025), discussed in Section 6, is particularly relevant because it uses reward-signed policy gradients to select RLVR prompts.

Within this literature, DRIFT addresses two choices in influence-based selection. Its on-policy PG formulation determines the validation responses and objective, motivated by locality concerns in influence approximation (Li et al., 2024b; Bae et al., 2022; Epifano et al., 2023). Its task-specific correction addresses the dependence of raw scores on gradient magnitude (Hu et al., 2025; Xia et al., 2024). Both choices serve the goal of improving final model performance, evaluated here through checkpoint refinement.