EFFECTIVE SYNTHETIC DATA CURATION REQUIRES GROUP-LEVEL SIGNALS

Anonymous authors
Paper under double-blind review

ABSTRACT

Synthetic data now is essential to LLM training, used to strengthen advanced capabilities such as autonomous and long-horizon task execution. Yet recent work shows that training on it at scale can degrade model generation, making it important to decide what synthetic data is worth training on. While current data curation practices do so with *individual-level* signals (i.e., estimates of each data sample's training utility in isolation), across pre-training and post-training settings we show that this is insufficient for synthetic data, and that *group-level* signals (i.e., estimates of utility that account for interactions among data samples) are necessary for effective data curation. First, we show that individual-level signals are blind to how samples jointly affect training: synthetic datasets with different compositions can be indistinguishable under individual-level influence yet differ sharply under group-level influence, and curating by the latter yields better downstream performance, particularly in generative capability. Second, we find that group-level signals matter more as training pipelines become increasingly synthetic: among widely used data curation methods, only those incorporating them improve over baseline, with gains increasing when weights capturing relations among samples are amplified. Finally, we translate these findings into practice – for model developers under a compute budget, we offer a cheap diagnostic that prioritizes which groups of synthetic data most need group-level estimation, recovering much of the benefit of full group-level scoring at a fraction of the compute cost.

1 INTRODUCTION

Synthetic data is crucial to large language model (LLM) development, and is now heavily used across all stages of LLM training, from synthetic corpora in pre-training (Li et al., 2023; Abdin et al., 2024) to inducing reasoning post-training (Nvidia et al., 2024; Team, 2025). It supplies targeted training signal for advancing LLM capabilities that are difficult to learn from organic web data alone, such as extended-horizon coding autonomy (Pan et al., 2025), competition-level mathematics (Trinh et al., 2024; Ren et al., 2025; Hubert et al., 2026), and reliable tool use (Qin et al., 2023; Liu et al., 2024b).

Although synthetic data is essential to continued progress in LLM capabilities, using it requires care. Recent work shows that training on it at scale can degrade LLM generative capabilities (a phenomenon known as "*model collapse*") since synthetic data exhibits narrower coverage than organic web data and inherits generator biases (Dohmatob et al., 2024; Guo et al., 2024; Shumailov et al., 2024; Chen et al., 2024a). Moreover, these consequences are costly for LLM developers, who waste substantial compute to recover from collapsed training runs (Chowdhery et al., 2023; Wortsman et al., 2024; Grattafiori et al., 2024). Therefore, using synthetic data effectively requires understanding how it contributes to (or harms) model training, so that it can be selected carefully.

While current data curation practices guide this selection with individual-level signals (i.e., estimates of each data sample's training utility in isolation), we show that these signals are insufficient, and that effective synthetic data curation instead requires group-level signals (i.e., estimates of training utility that account for interactions among data samples). First, individual-level signals are blind to these interactions: using data influence measures to quantify the utility of synthetic datasets, we find that datasets with different compositions (e.g., domains and diversity) can be indistinguishable under individual-level influence yet differ under group-level influence. Curating training data by group-level influence also yields better downstream performance, particularly in *generative capability*.

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.

Next, we show that as LLM training pipelines grow increasingly reliant on synthetic data, incorporating group-level signals in standard data curation methods is necessary to enhance model performance. Analyzing several data curation methods used in practice (Trotman et al., 2014; Xia et al., 2024; Yu et al., 2025; Wang et al., 2024), we find that, in this setting, methods which use group-level signals avoid model degradation and improve over baseline performance, whereas individual-level methods do neither. Gains increase further when the weights that capture relational information among data samples are amplified, suggesting that interactions within datasets should be modeled explicitly as pipelines rely more on synthetic data.

Finally, we build on our findings to address a practical concern: group-level methods can be computationally expensive, since capturing interactions among data samples often requires additional memory and compute, and practitioners with limited resources need to know where to allocate budget for them. To guide this decision, we offer efficient diagnostics: by leveraging within-group gradient diversity, we identify groups where individual-level signals are likely to be misleading. Allocating budget to scoring these groups recovers much of the group-level signal at a fraction of the compute cost. To summarize, our contributions and findings are:

1. **Individual-level signals are insufficient for curating synthetic data.** Datasets that are indistinguishable under individual-level influence differ greatly in group-level influence, and curating by group-level influence yields better downstream performance.
2. **Group-level signals matter more as data pipelines grow increasingly synthetic.** Among widely used data curation methods, we find only methods which incorporate group-level signals improve over baseline, with gains in generative capability.
3. **Leveraging group-level signals under a compute budget.** We offer practitioners cheap diagnostics, leveraging within-group gradient diversity, to decide where group-level signals are worth computing under a tight budget.

2 PRELIMINARIES

We begin with background on data influence, specifically oracle scoring, the instrument we use in our later analyses for synthetic data curation.

Oracle scoring. Data influence measures the effect of a data sample on model behavior, and one way to quantify it is "oracle scoring", which compares a given model trained with and without the sample (Koh & Liang, 2017; Pruthi et al., 2020; Basu et al., 2020). Formally, given a training dataset $D_{\text{train}} = \{z_i\}_{i=1}^{N}$, a reference set $D_{\text{ref}} = \{z'_i\}_{i=1}^{M}$ with $M \ll N$, and a model with parameters $\theta_{D_{\text{train}}}$ obtained by training on D_{train}, we define the oracle score as:

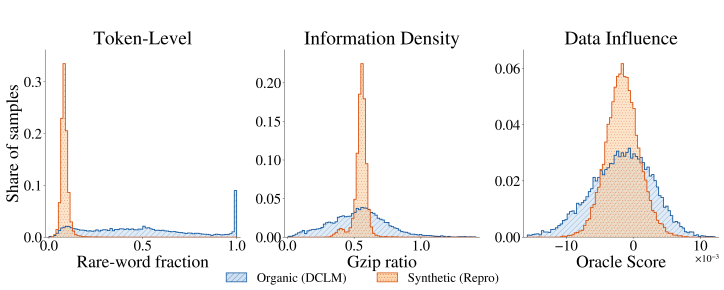
Definition 2.1 (Oracle Score: Group and Individual)**.** The *oracle score* of a set of training examples $G = \{z_j\}_{j=1}^{|G|}$ measures the change in model performance when G is included versus excluded from the training data, evaluated with respect to the model loss on a reference set D_{ref}:

$$\mathcal{I}_{\text{orcl}}(G, \theta) = L(D_{\text{ref}} \mid \theta_{D_{\text{train}} \cup G}) - L(D_{\text{ref}} \mid \theta_{D_{\text{train}}}) \,. \tag{1}$$

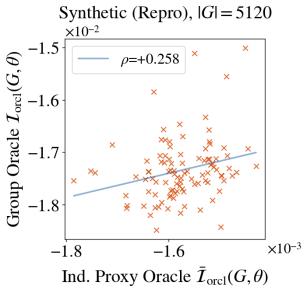
We refer to $\mathcal{I}_{\text{orcl}}(G, \theta)$ as the *group oracle score*. For a single example z ($|G| = 1$) we write $\mathcal{I}_{\text{orcl}}(z, \theta) \coloneqq \mathcal{I}_{\text{orcl}}(\{z\}, \theta)$ and call it the *individual oracle score*.

In standard data influence settings, the reference set is chosen to *reflect* a target capability (e.g., math or code text), so that loss on D_{ref} is a measurable proxy for that capability. Prior work has shown that selecting training data to reduce the oracle reference loss yields gains on the corresponding downstream tasks in LLM pre-training and post-training (Xia et al., 2024; Yu et al., 2024).

Oracle scoring for datasets. Existing data curation pipelines typically rely on scorers that operate at the *individual level* – such as heuristic deduplication filters and text quality scorers – which score each sample in isolation (Li et al., 2024; Penedo et al., 2024). Conceptually, these methods approximate the group oracle score by aggregating individual oracle scores, which we formalize as the *individual proxy oracle* below.



(1a) **Synthetic data is more homogeneous.** Spread of token-level, information density, and data influence measures over n=1000 samples each from organic (DCLM) and synthetic (Repro) data.



(1b) **Group value is not additive:** the ind. proxy oracle tracks the group oracle only weakly.

Definition 2.2 (Individual Proxy Oracle)**.** The group oracle is approximated by the mean of individual oracle scores,

$$\mathcal{I}_{\text{orcl}}(G,\theta) \;\approx\; \mathcal{I}_{\text{proxy}}(G,\theta) = \frac{1}{|G|}\sum_{z\in G}\mathcal{I}_{\text{orcl}}(z,\theta). \tag{2}$$

In practice, oracle scores are expensive since they require model training. However, for the individual proxy oracle, cheaper approximation methods exist for LLM data curation, such as LESS and LoGRA (Choe et al., 2025; Xia et al., 2024), making the individual proxy oracle in Equation (2) tractable for group-level influence estimation compared to the true group-level oracle.

With these measures in place, a natural question is why synthetic data calls for particular attention. Below, we discuss two key challenges that arise in curating synthetic datasets when we solely rely on individual-level scores.

Challenge 1: Synthetic data is homogeneous. A large synthetic corpus is often generated from a few seed prompts (Li et al., 2023; Maini et al., 2024) (e.g., instruction-following data bootstrapped from a small pool of human-written tasks (Wang et al., 2023b; Taori et al., 2023)), making it more homogeneous than an organic corpus collected from the web. This homogeneity makes it harder to distinguish the synthetic samples that most improve model performance. As shown in Figure 1a, synthetic data samples have lower variance than organic ones not only at the token-level and in information density, *but also* in individual-level data influence.

Challenge 2: A dataset's value is non-additive. Moving beyond individual data points, a dataset's value is shaped by the combined contribution of its members: interactions between samples can induce cancellation or amplification effects (Koh et al., 2019; Basu et al., 2020), making a dataset's value non-additive in its samples. These properties are invisible to an individual scorer. Figure 1b shows an example, where the individual proxy oracle does not track the group oracle, causing a mismatch between the datasets each ranks highly for model training, which we explore in the following sections.

3 Group-level Signals for Synthetic Data Influence

The previous section showed that the individual proxy oracle and the group oracle can disagree. Here, we show through controlled experiments that this disagreement matters in practice: synthetic data groups that look equivalent under the individual proxy can differ significantly in group-level influence and downstream value. The remainder of the section describes the task settings in Section 3.1, the experimental procedure in Section 3.2, and the results in Section 3.3.

3.1 Task Settings

We study the gap between individual and group-level influence in pre-training and post-training settings. The datasets and models used in both settings are described below.

Pre-training. The first setting is synthetic data pre-training, which is increasingly being incorporated into frontier LLM training pipelines (Liu et al., 2024a; Grattafiori et al., 2024; Yang et al., 2025). In this setting, we train on data from the Repro-Rephrased-72B (Yu & Xiong, 2025) dataset, which has been refined from low-quality web scrapes, and oracle scoring uses Repro-400M (trained on 6.59B Repro tokens) (Yu & Xiong, 2025) as the base model against a 128-sample FLAN (Chung et al., 2024) reference set, following prior oracle-scoring work (Yu et al., 2024; 2025; Jiao et al., 2025). We evaluate on eight standard pre-training benchmarks (ARC-Challenge, CommonsenseQA, CoQA, HellaSwag, LAMBADA, MMLU, PIQA, SQuAD).

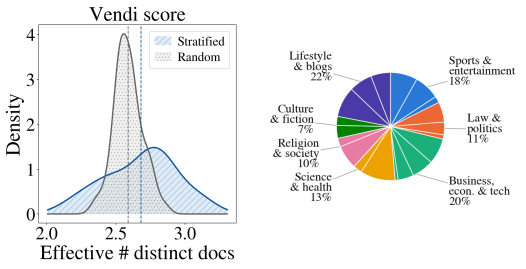
Post-training. The second setting is post-training using reinforcement learning with verifiable rewards (RLVR), a key factor in inducing reasoning in LLMs for coding, math, and agentic tasks. Here, we train on data from the GooseReason (Lu et al., 2026) dataset, and oracle scoring uses Qwen2.5-1.5B-Instruct (Qwen et al., 2025) as the base model against a 128-sample GPQA-extend (Rein et al., 2024) reference set, chosen for its graduate-level, multi-domain reasoning coverage. We evaluate on reasoning/math/coding benchmarks (MBPP, MBPP+, GPQA-D, BBH, IFEval, AIME24).

Oracle scoring parameters for both settings can be found in Appendix B.1

3.2 Experiment Setup

Below, we discuss our general experiment setup, which we instantiate identically for both settings above and which consists of three phases: candidate pool construction, data selection, and model training. We describe the overall procedure here and *defer the specific details to Appendix B.3*.

Phase 1: Candidate Pool Construction. To isolate the interactions between data samples that the individual proxy fails to capture, we form groups with (i) similar individual proxy scores and (ii) different compositions. The former holds the individual proxy fixed as a control, while the latter varies how samples within a group interact and ensures the candidate pool spans a broad range of data, covering varied levels of diversity and topics.



(a) Semantic diversity. (b) Topic diversity.

Figure 2: Semantic/topic diversity of our candidate pool on the pre-training corpus.

From each corpus (Repro and GooseReason), we build a pool of H equal-sized groups $\mathcal{G} = \{G_1, \ldots, G_H\}$. To control individual proxy scores, we draw each group's members from above and below a target score in a narrow band such that the group's mean matches the target. To vary composition, we partition the corpus into 20 semantic clusters (using BGE embeddings with k-means) and draw each group from between 1 (homogeneous) and 20 (diverse) clusters. Additional details on candidate pool construction are discussed in Appendix B.2.

Figure 2a shows that our groups span a wider semantic diversity range than uniformly random groups, and Figure 2b shows the breakdown of topics across the 20 clusters (a full breakdown of per-corpus cluster topics are in Appendix Figure 10).

Phase 2: Data selection. From the candidate pool $\mathcal{G}$, each scoring method selects a set of k groups to form the training sets $\mathcal{D}_{\text{proxy}}$ and $\mathcal{D}_{\text{orcl}}$. Following prior work that uses data influence for data selection (Yu et al., 2024; Jiao et al., 2025), we draw these groups via *Gumbel-Top-k* sampling (Kool et al., 2019) so as to better reflect the distribution of oracle scores.

$$\mathcal{D}_{\text{proxy}} \leftarrow \text{Gumbel-Top-}k\left(\{\mathcal{I}_{\text{proxy}}(G,\theta)/\lambda \mid G \in \mathcal{G}\}\right)$$
$$\mathcal{D}_{\text{orcl}} \leftarrow \text{Gumbel-Top-}k\left(\{\mathcal{I}_{\text{orcl}}(G,\theta)/\lambda \mid G \in \mathcal{G}\}\right),$$

where $\lambda \in (0, \infty)$ is the temperature.

Phase 3: Training and evaluation. We compare three selection methods under a matched budget: the group oracle ($\mathcal{D}_{\text{orcl}}$), the individual proxy oracle ($\mathcal{D}_{\text{proxy}}$), and a random baseline that draws k groups uniformly from the same pool $\mathcal{G}$. Starting from the same base model (Repro-400M for

Table 1: Results for pre- and post-training data selection. Per-task accuracy (%) on Repro-400M (400 steps) and Qwen2.5-1.5B-Instruct (100 steps), for groups selected by random sampling, the individual proxy oracle, and the group oracle. Best average within each group size is in **bold**.

(a) Pre-training: Repro-400M

Group Size	Method	HellaSwag	ARC-C	PIQA	MMLU	CSQA	LAMBADA	SQuAD	CoQA	Avg.
	Baseline	30.82	25.85	64.36	23.94	19.82	37.30	7.47	27.61	29.65
5120	Random	31.84	27.43	66.21	25.53	20.07	37.29	9.69	28.83	30.86
	Ind. Proxy Orcl.	32.01	26.96	66.59	26.01	21.05	38.09	10.25	28.62	31.20
	Group Orcl.	31.78	26.54	67.30	25.10	20.56	38.44	16.19	29.97	**31.98**
10240	Random	31.58	25.94	66.35	24.74	20.31	38.66	10.47	28.49	30.82
	Ind. Proxy Orcl.	31.53	26.54	66.05	24.30	20.15	37.63	9.65	28.49	30.54
	Group Orcl.	31.40	26.45	66.38	24.74	20.07	38.87	9.65	30.42	**31.00**

(b) Post-training: Qwen2.5-1.5B-Instruct

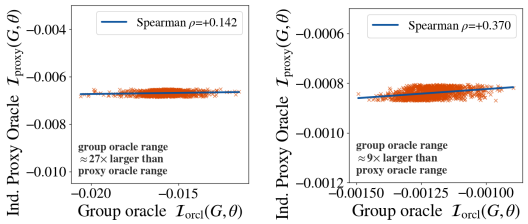
Group Size	Method	MBPP	MBPP+	GPQA-D	BBH	IFEval	AIME24	Avg.
	Baseline	41.20	45.50	25.93	37.46	44.20	3.00	32.88
128	Random	44.00	48.59	26.54	40.42	44.60	1.11	34.21
	Ind. Proxy Orcl.	42.60	46.03	26.26	40.87	44.40	1.00	33.53
	Group Orcl.	45.40	52.12	26.43	40.56	45.80	1.00	**35.22**
512	Random	43.40	48.41	25.00	40.39	45.20	1.33	33.96
	Ind. Proxy Orcl.	43.80	49.21	26.43	37.64	45.20	2.67	34.16
	Group Orcl.	44.20	50.53	26.94	40.92	46.00	2.33	**35.15**

pre-training and Qwen2.5-1.5B-Instruct for post-training), we train one model per set with identical hyperparameters and step counts, so differences in downstream performance come from selection alone. We report accuracy on the benchmarks in Section 3.1.

3.3 Results

We examine how group oracle scores vary across the candidate pool, then compare group oracle, individual proxy oracle, and random data selection on downstream model performance.

Finding 1: Group oracle recovers signal the individual proxy misses. Figure 3 shows that although all groups in the candidate pool have near-identical individual proxy scores, their group oracle scores are mismatched and span a much wider range. This mismatch matters in practice: Table 1 shows that, across all task settings and group sizes, training on data selected by the group oracle outperforms both random selection and individual proxy oracle selection.



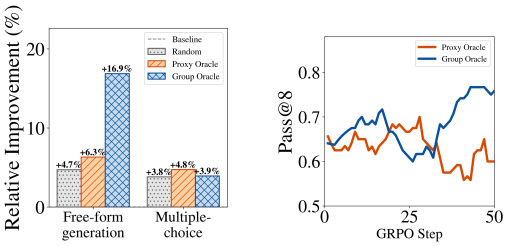
(a) Pre-train. (b) Post-train.

Figure 3: Range of group oracle and individual proxy oracle scores across our constructed groups (on groups of 5120 and 128 samples). See Appendix D.4 for full sweep over group sizes.

Finding 2: Gains concentrate on generative capability. Next, to see where the group oracle's advantage over the individual proxy oracle comes from, we examine the task types. Figure 4a breaks these gains down by task type. On free-form generation (LAMBADA, SQuAD, CoQA), group oracle selection achieves a $+16.9\%$ relative gain over the base model, far above individual proxy oracle selection ($+6.3\%$) and random selection ($+4.7\%$). On multiple-choice tasks (HellaSwag, ARC-C, PIQA, MMLU, CSQA), all three are comparable ($+3.8\%$ to $+4.8\%$). The RLVR setting shows the same pattern (Figure 4b): over the course of GRPO training, pass@8 increases under group oracle selection, while under individual proxy oracle selection it ends below its starting value.

Together, our results point to two blind spots in synthetic data curation. First, practitioners should not rely solely on individual-level scoring, as interactions between samples influence model training in ways that individual-level scoring does not capture. Instead, practitioners should leverage group-level scoring, which captures the full value of a dataset; we discuss this further in Section 5.

Second, multiple-choice benchmarks can hide this gap, since all selection methods perform similarly on them. It is therefore important to also evaluate on generative tasks, where the difference between individual- and group-level scoring becomes visible.



(a) Pre-train (size 5120). (b) Post-train (size 512).

Figure 4: *(a)* Relative improvement over the base model by task type in pre-training. *(b)* Pass@8 over GRPO training steps in post-training.

4 Group-level Signals for Synthetic Data Curation

Next, we turn to the data selection methods that model developers use in practice, and show how group-level signals become more necessary as training pipelines grow increasingly synthetic. Section 4.1 describes the experimental setup and Section 4.2 the results.

4.1 Experiment Setup

We study synthetic data recycling, where noisy web text is rephrased into clean training data. Recycling eases data scarcity by generating multiple samples per seed, and is used to pre-train recent LLMs such as Kimi K2 and Qwen3 (Kimi, 2025; Yang et al., 2025). Below, we discuss our general experiment setup, consisting of three phases: synthetic pool construction, data selection, and training/evaluation. We describe the overall procedure here and *defer specific details to Appendix B.4*.

Phase 1: Synthetic Pool Construction. We begin with a large corpus of noisy organic documents and use a rephraser model to paraphrase them, producing clean synthetic pools. Specifically, we take the organic corpus to be DCLM (Li et al., 2024) and the rephraser to be Repro (Yu & Xiong, 2025), a 4B model fine-tuned to paraphrase web text.

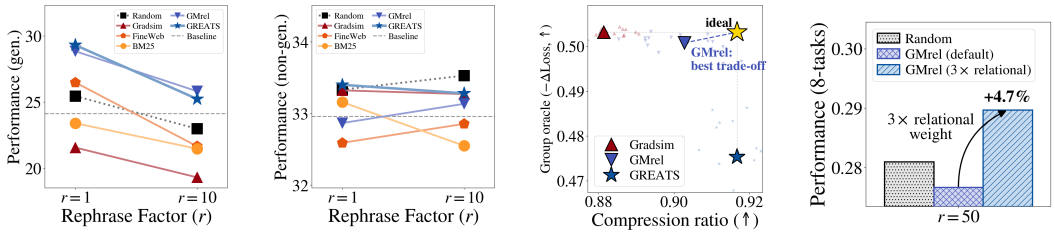
Each synthetic pool has the same fixed size $N = 716{,}800$ sequences but a different *rephrase factor* $r \in \{1, 10, 50\}$, which controls the number of rephrases per organic data sample: for a given r, we sample N/r source documents from the organic corpus and draw r paraphrases of each. Here, $r = 1$ and $r = 10$ span the range used in practice (e.g., Kimi K2 rephrases its knowledge corpus up to 10 times (Kimi, 2025)), while $r = 50$ serves as a stress test of highly synthetic training data.

Phase 2: Data selection. We compare multiple data selection methods (see Table 2) from two families: individual-level methods, which score each sample in isolation, and group-level methods, which also model relations among samples. Each method selects data from the Phase 1 pools to construct training sets. For efficiency at this scale, we adopt the cluster-based inference of Yu et al. (2025), with which each method forms 20 groups of size 5120. See Appendix B.6 for full details.

	Method	Representation
Individual	GradSim (Pruthi et al., 2020)	Gradient
	FineWeb-Edu (Penedo et al., 2024)	Classifier score
	BM25 (Trotman et al., 2014)	Lexical (tokens)
Group	GREATS (Wang et al., 2024)	Gradient
	GMRel (Yu et al., 2025)	Learned influence

Table 2: Taxonomy of individual-and group-level data selection methods. Details in Appendix B.5

Phase 3: Training and evaluation. For each method we train the same Repro-400M base model as in Section 3 (Yu & Xiong, 2025), on its formed groups under identical optimization: batch size 512 for 200 steps, peak learning rate 3×10^{-3} under a cosine schedule (no warmup, weight decay 0.033). We evaluate the trained models on the same pre-training suite as in Section 3.



(a) Generative tasks. (b) Non-Gen. Tasks. (c) Value-div. tradeoff. (d) Relational factor.

Figure 5: *(a, b)* Performance of individual-/group- level selection methods as rephrase factor increases. *(c)* Group-oracle value and diversity trade-off. *(d)* Increasing GMRel's relational weight at $r = 50$.

4.2 Results

We evaluate each selection method as the source pool becomes increasingly synthetic. We focus on the rephrase factors $r = 1$ and $r = 10$, which span the range used in frontier LLM pre-training (Kimi, 2025), and examine $r = 50$ separately below, deferring the full breakdown for all r to Appendix D.1.

Finding 1: Individual-level selection methods collapse on generative tasks. Figures 5b and 5a show downstream performance on non-generative and generative tasks, respectively, when individual-level selection methods (GradSim, FineWeb-Edu, and BM25) and group-level selection methods (GMRel and GREATS) are used to curate training data. While Figure 5b shows that all methods plateau on multiple-choice tasks, the difference between the two families is visible in Figure 5a, where the group-level selection methods outperform the individual-level selection methods, similar to the observations in Section 3.3.

Finding 2: Group-level selection methods balance value and diversity. Next, we examine the characteristics of the groups formed by individual-level and group-level selection methods. Figure 5c plots group oracle value against diversity for the methods that approximate oracle influence (GradSim, GMRel, and GREATS) at $r = 1$. GradSim greedily selects the highest-scoring samples, which gives it a high group oracle value but the least diverse groups. This lack of diversity may explain why it, like the other individual-level methods, underperforms on generative tasks. GMRel and GREATS, in contrast, form more diverse groups at a slight cost to group oracle value. This suggests that balancing value and diversity is important for effective synthetic data selection.

Finding 3: Stronger relational weighting helps with highly synthetic training data. Next, we analyze the extent to which group-level selection methods are effective in highly synthetic data pipelines, using the $r = 50$ stress test (see Table 9 in Appendix D.1). Here, Figure 5d shows that tripling GMRel's relational weight improves its relative downstream performance by 4.7%, surpassing random selection and baseline. This indicates that as the training data pipelines become more synthetic, group-level selection methods should weigh inter-sample relationships more heavily.

As in Section 3, our findings suggest that forming groups of data to train on requires careful consideration. As data pipelines come to depend on synthetic and self-generated data, group effects in data selection methods should be modeled directly to prevent suboptimal selection.

5 Diagnostics for Group-Level Data Selection

In this section, we turn to the practical cost of group-level scoring. Because group-level scores capture relational information between samples, they require more compute than individual-level scores, raising the question of where practitioners with limited compute should spend a group-scoring budget. We show that this question has a principled answer: the gap between individual and group scoring is bounded by a quantity that is cheap to compute, which serves as an efficient diagnostic that reliably predicts the gap in practice and directs group oracle calls to the groups that are most poorly reflected by individual-level scoring.

Characterizing the oracle gap. We start by considering the question: *how far is the individual proxy oracle from the true group oracle?* To answer this, we turn to Proposition 5.1, which bounds how far the individual proxy oracle can be from the true group oracle by the *within-group gradient diversity* (i.e., the spread of per-sample gradients within a group).

Proposition 5.1 (Oracle gap)**.** *The individual proxy and true group oracles differ by at most*

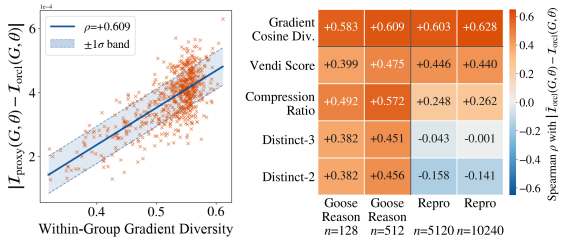
$$\left|\mathcal{I}_{\text{proxy}}(G,\theta) - \mathcal{I}_{\text{orcl}}(G,\theta)\right| \;\le\; \frac{\beta\,\eta^2}{2}\,\sigma_G^2, \tag{3}$$

where $\beta \ge \|\nabla^2 L(\theta)\|$ bounds the curvature of the reference loss and $\sigma_G^2 = \frac{1}{|G|}\sum_{z\in G}\|g_z - g_G\|^2$ is the within-group gradient diversity. Proof in Appendix C.

Intuitively, the more the samples pull the model in different directions, the more their disagreements cancel in the group's mean update, an interaction invisible to a proxy that scores samples in isolation.

Gradient diversity predicts the oracle gap. The bound in Proposition 5.1 holds empirically: Figure 6a shows that as within-group gradient diversity increases, the gap between $\mathcal{I}_{\text{proxy}}$ and $\mathcal{I}_{\text{orcl}}$ widens (see Appendix D.2 for additional settings).

Gradient diversity is also the strongest predictor of the gap among additional diversity metrics (see Appendix B.7 for details), as shown in Figure 6b. Its Spearman correlation with the gap is high ($\rho = 0.58$–0.63) across both datasets and all group sizes. By contrast, the best alternatives peak at 0.48 (Vendi) and 0.57 (compression ratio), and they vary widely. Moreover, token-level metrics are unreliable: distinct-2/-3 correlate moderately on GooseReason but not on Repro.



(a) Oracle gap vs Diversity.

(b) Oracle gap heatmap.

Figure 6: Gap between the group and individual proxy oracle diverges as within-group diversity increases.

Given these findings, we show how gradient diversity can be used as a diagnostic to decide where to spend group oracle compute.

Gradient diversity as a budget allocator. Proposition 5.1 implies that oracle calls are best spent on groups with high within-gradient diversity, where the proxy can be far off and is therefore most likely to misrank groups. This suggests a simple allocation strategy. Formally, given a pool of groups $\mathcal{G} = \{G_1, \ldots, G_H\}$ and a budget of B group oracle calls, we define a *diagnostic* as a cheap score $\nu : \mathcal{G} \to \mathbb{R}$, where a larger value signals a larger predicted oracle gap. Given ν, we audit the B top-ranked groups, denoted by $\mathcal{A}_\nu$,

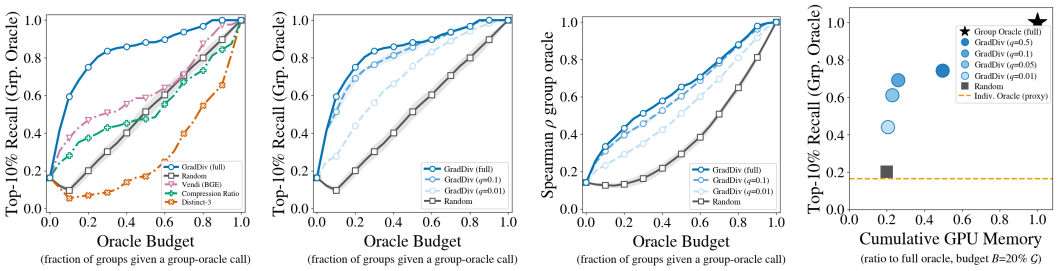
$$\mathcal{A}_\nu \;=\; \operatorname*{arg\,top}_{G\in\mathcal{G}}^{B} k \;\; \nu(G), \tag{4}$$

and score every group with the best oracle we can afford for it,

$$\widehat{\mathcal{I}}_{\mathcal{A}}(G,\theta) \;=\; \begin{cases} \mathcal{I}_{\text{orcl}}(G,\theta), & G \in \mathcal{A}_\nu, \\ \mathcal{I}_{\text{proxy}}(G,\theta), & \text{otherwise.} \end{cases} \tag{5}$$

Setup. Given the above procedure, we assess the performance of different diagnostics. The diagnostics we compare are as follows:

1. *GradDiv$_{full}$*: uses $\nu(G) = \sigma_G^2$ from Proposition 5.1
2. *GradDiv$_q$*: uses the same estimator on a uniform subsample of $\lceil q\,|G|\rceil$ samples, $q \in (0, 1]$ (i.e., $q\,|G|$ gradients per group)
3. *Gradient-free*: use Vendi, compression ratio, or distinct-n (Appendix B.7; no gradients)
4. *Random allocation*: draws $\nu(G) \sim \text{Unif}(0, 1)$ i.i.d.



(a) Choice of diagnostic. (b) Gradient subsampling. (c) Rank agreement. (d) Memory versus quality.

Figure 7: Allocating a limited group-oracle budget with cheap diagnostics. See Appendix D.5 for additional analysis plots.

Moreover, in Eqn 5 we substitute $\mathcal{I}_{\text{proxy}}$ with GradSim (from Section 4.1), as it is a cheap estimate of the individual proxy oracle more commonly used in practice (Pruthi et al., 2020; Xia et al., 2024).

We test these diagnostics on the constructed groups from Section 3, where the individual proxy oracle is blind to the group effects. We measure how well each diagnostic recovers the group oracle's ranking with two metrics: *top-k% recall*, the fraction of the group oracle's top-k% groups that also rank in the top k% under $\widehat{\mathcal{I}}_{\mathcal{A}}$, and Spearman ρ between $\widehat{\mathcal{I}}_{\mathcal{A}}$ and $\mathcal{I}_{\text{orcl}}$ over the full pool.

Finding 1: Gradient diversity can recover the group oracle. Figure 7a shows that $\text{GradDiv}_{\text{full}}$ is best at retrieving the top-ranked groups under a fixed budget: at a budget of 20% group-oracle calls, it recovers 75% of the top-10% groups, compared to 20% under random allocation. In contrast, the gradient-free diagnostics recover only part of this signal: Vendi and compression ratio reach 0.44 and 0.40 recall at the same budget, while the token-level distinct-3 falls below the random baseline.

Finding 2: Smaller gradient subsamples suffice. Next, we examine whether the diagnostic stage itself can be made cheaper. Figures 7b and 7c show that GradDiv_q gives similar benefit at $q = 0.1$ (only sampling 10% of a group's gradients) compared to $\text{GradDiv}_{\text{full}}$ on both recall (0.69 versus 0.75 at a 20% budget) and Spearman correlation (0.40 versus 0.44). This indicates that only a coarse estimate of $\text{GradDiv}_{\text{full}}$ is needed to rank groups, allowing the diagnostic to avoid much of the per-sample gradient cost.

GPU memory analysis. Finally, we examine the compute cost. Figure 7d reports the total cumulative memory usage at a 20% budget (relative to scoring all groups using the group oracle), showing that GradDiv_q exhibits a better memory-performance trade-off compared to full group oracle scoring and random allocation. In particular, at $q = 0.1$, GradDiv_q recovers 74% of the group oracle's recall while using 27% of its memory. Full details on memory computation can be found in Appendix B.8.

6 Conclusion

In this paper, we asked which signals identify what synthetic data is worth training on, and showed that *group-level* signals, which capture interactions among samples, are needed. Across pre-training and post-training, synthetic datasets indistinguishable under individual-level influence differ under group-level influence, which also yields better downstream performance, with gains concentrated in generative capability. Among widely used data curation methods, only those using group-level signals improve over baseline, with further gains when relational weights are amplified. Finally, we recommend within-group gradient diversity as a diagnostic for when group-level signals are worth their cost. Our findings motivate future work on scalable group-level curation methods that model this relational structure directly.

AI Statement

We used generative AI tools to: provide critical ingredients for proving mathematical claims, assist in the writing of proofs, design or provide feedback on research methodology or experiments, implement

methods, clean and reformat datasets, support qualitative and thematic data analysis, and create new synthetic data sets. We did not use AI tools to: help develop theoretical models or conceptual frameworks, formulate mathematical claims, propose or refine hypotheses, assist with translation, or interpret results. Additionally, we used generative AI tools to: create or modify scientific figures or images, draft parts of a research paper, source/search for information, edit a research paper to improve readability. We have reviewed all AI-assisted work: LLM-generated code was verified via unit tests and citations/literature were verified manually. We take responsibility for the final content of this work, including text, claims or artifacts produced with the aid of generative AI.

REFERENCES

Marah Abdin, Jyoti Aneja, Harkirat Behl, Sébastien Bubeck, Ronen Eldan, Suriya Gunasekar, Michael Harrison, Russell J. Hewett, Mojan Javaheripi, Piero Kauffmann, James R. Lee, Yin Tat Lee, Yuanzhi Li, Weishung Liu, Caio C. T. Mendes, Anh Nguyen, Eric Price, Gustavo de Rosa, Olli Saarikivi, Adil Salim, Shital Shah, Xin Wang, Rachel Ward, Yue Wu, Dingli Yu, Cyril Zhang, and Yi Zhang. Phi-4 technical report, 2024. URL https://arxiv.org/abs/2412.08905.

Bo Adler, Niket Agarwal, Ashwath Aithal, Dong H Anh, Pallab Bhattacharya, Annika Brundyn, Jared Casper, Bryan Catanzaro, Sharon Clay, Jonathan Cohen, et al. Nemotron-4 340b technical report. *arXiv preprint arXiv:2406.11704*, 2024.

Samyadeep Basu, Xuchen You, and Soheil Feizi. On second-order group influence functions for black-box predictions. In *International Conference on Machine Learning*, pp. 715–724. PMLR, 2020.

Hao Chen, Abdul Waheed, Xiang Li, Yidong Wang, Jindong Wang, Bhiksha Raj, and Marah I Abdin. On the diversity of synthetic data and its impact on training large language models. *arXiv preprint arXiv:2410.15226*, 2024a.

Lichang Chen, Shiyang Li, Jun Yan, Hai Wang, Kalpa Gunaratna, Vikas Yadav, Zheng Tang, Vijay Srinivasan, Tianyi Zhou, Heng Huang, et al. Alpagasus: Training a better alpaca with fewer data. In *International Conference on Learning Representations*, volume 2024, pp. 34767–34797, 2024b.

Sang Choe, Hwijeen Ahn, Juhan Bae, Kewen Zhao, Youngseog Chung, Adithya Pratapa, Willie Neiswanger, Emma Strubell, Teruko Mitamura, Jeff Schneider, Eduard Hovy, Roger Grosse, and Eric Xing. What is your data worth to gpt? llm-scale data valuation with influence functions. In D. Belgrave, C. Zhang, H. Lin, R. Pascanu, P. Koniusz, M. Ghassemi, and N. Chen (eds.), *Advances in Neural Information Processing Systems*, volume 38, pp. 145944–145985. Curran Associates, Inc., 2025. URL https://proceedings.neurips.cc/paper_files/paper/2025/file/d6d26053b977f8c589669fd201615119-Paper-Conference.pdf.

Aakanksha Chowdhery, Sharan Narang, Jacob Devlin, Maarten Bosma, Gaurav Mishra, Adam Roberts, Paul Barham, Hyung Won Chung, Charles Sutton, Sebastian Gehrmann, et al. Palm: Scaling language modeling with pathways. *Journal of machine learning research*, 24(240):1–113, 2023.

Hyung Won Chung, Le Hou, Shayne Longpre, Barret Zoph, Yi Tay, William Fedus, Yunxuan Li, Xuezhi Wang, Mostafa Dehghani, Siddhartha Brahma, et al. Scaling instruction-finetuned language models. *Journal of Machine Learning Research*, 25(70):1–53, 2024.

Elvis Dohmatob, Yunzhen Feng, Arjun Subramonian, and Julia Kempe. Strong model collapse, 2024. URL https://arxiv.org/abs/2410.04840.

Logan Engstrom, Axel Feldmann, and Aleksander Madry. Dsdm: Model-aware dataset selection with datamodels. In *International Conference on Machine Learning*, pp. 12491–12526. PMLR, 2024.

Dan Friedman and Adji Bousso Dieng. The vendi score: A diversity evaluation metric for machine learning. *arXiv preprint arXiv:2210.02410*, 2022.

Amirata Ghorbani and James Zou. Data shapley: Equitable valuation of data for machine learning. In *International conference on machine learning*, pp. 2242–2251. PMLR, 2019.

Aaron Grattafiori, Abhimanyu Dubey, Abhinav Jauhri, Abhinav Pandey, Abhishek Kadian, Ahmad Al-Dahle, Aiesha Letman, Akhil Mathur, Alan Schelten, Alex Vaughan, et al. The llama 3 herd of models. *arXiv preprint arXiv:2407.21783*, 2024.

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.

Yanzhu Guo, Guokan Shang, Michalis Vazirgiannis, and Chloé Clavel. The curious decline of linguistic diversity: Training language models on synthetic text. In *Findings of the Association for Computational Linguistics: NAACL 2024*, pp. 3589–3604, 2024.

Yuzheng Hu, Pingbang Hu, Han Zhao, and Jiaqi W. Most influential subset selection: Challenges, promises, and beyond. In A. Globerson, L. Mackey, D. Belgrave, A. Fan, U. Paquet, J. Tomczak, and C. Zhang (eds.), *Advances in Neural Information Processing Systems*, volume 37, pp. 119778–119810. Curran Associates, Inc., 2024. URL https://proceedings.neurips.cc/paper_files/paper/2024/file/d8684e49752e06ac5e4b554b60ad212a-Paper-Conference.pdf.

Thomas Hubert, Rishi Mehta, Laurent Sartran, Miklós Z Horváth, Goran Žužić, Eric Wieser, Aja Huang, Julian Schrittwieser, Yannick Schroecker, Hussain Masoom, et al. Olympiad-level formal mathematical reasoning with reinforcement learning. *Nature*, 651(8106):607–613, 2026.

Andrew Ilyas, Sung Min Park, Logan Engstrom, Guillaume Leclerc, and Aleksander Madry. Datamodels: Predicting predictions from training data, 2022. URL https://arxiv.org/abs/2202.00622.

Ruoxi Jia, David Dao, Boxin Wang, Frances Ann Hubis, Nick Hynes, Nezihe Merve Gürel, Bo Li, Ce Zhang, Dawn Song, and Costas J Spanos. Towards efficient data valuation based on the shapley value. In *The 22nd International Conference on Artificial Intelligence and Statistics*, pp. 1167–1176. PMLR, 2019.

Cathy Jiao, Yijun Pan, Emily Xiao, Daisy Sheng, Niket Jain, Hanzhang Zhao, Ishita Dasgupta, Jiaqi W. Ma, and Chenyan Xiong. Date-lm: Benchmarking data attribution evaluation for large language models. In *The Thirty-Ninth Annual Conference on Neural Information Processing Systems*, 2025. URL https://arxiv.org/abs/2507.09424.

Feiyang Kang, Newsha Ardalani, Michael Kuchnik, Youssef Emad, Mostafa Elhoushi, Shubhabrata Sengupta, Shang-Wen Li, Ramya Raghavendra, Ruoxi Jia, and Carole-Jean Wu. Demystifying synthetic data in llm pre-training: A systematic study of scaling laws, benefits, and pitfalls. In *Proceedings of the 2025 Conference on Empirical Methods in Natural Language Processing*, pp. 10750–10769, 2025.

Team Kimi. Kimi k2: Open agentic intelligence. *arXiv preprint arXiv:2507.20534*, 2025.

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

Pang Wei W Koh, Kai-Siang Ang, Hubert Teo, and Percy S Liang. On the accuracy of influence functions for measuring group effects. *Advances in neural information processing systems*, 32, 2019.

Wouter Kool, Herke van Hoof, and Max Welling. Stochastic beams and where to find them: The gumbel-top-k trick for sampling sequences without replacement. In *Proceedings of the 36th International Conference on Machine Learning*, volume 97 of *Proceedings of Machine Learning Research*, pp. 3499–3508. PMLR, 2019. URL https://proceedings.mlr.press/v97/kool19a.html.

Yongchan Kwon, Eric Wu, Kevin Wu, and James Y Zou. Datainf: Efficiently estimating data influence in lora-tuned llms and diffusion models. In *International Conference on Learning Representations*, volume 2024, pp. 21921–21942, 2024.

Jeffrey Li, Alex Fang, Georgios Smyrnis, Maor Ivgi, Matt Jordan, Samir Yitzhak Gadre, Hritik Bansal, Etash Guha, Sedrick Keh, Kushal Arora, et al. Datacomp-lm: In search of the next generation of training sets for language models, 2024. URL https://arxiv.org/abs/2406.11794.

Yuanzhi Li, Sébastien Bubeck, Ronen Eldan, Allie Del Giorno, Suriya Gunasekar, and Yin Tat Lee. Textbooks are all you need ii: phi-1.5 technical report. *arXiv preprint arXiv:2309.05463*, 2023.

Aixin Liu, Bei Feng, Bing Xue, Bingxuan Wang, Bochao Wu, Chengda Lu, Chenggang Zhao, Chengqi Deng, Chenyu Zhang, Chong Ruan, et al. Deepseek-v3 technical report. *arXiv preprint arXiv:2412.19437*, 2024a.

Zuxin Liu, Thai Hoang, Jianguo Zhang, Ming Zhu, Tian Lan, Shirley Kokane, Juntao Tan, Weiran Yao, Zhiwei Liu, Yihao Feng, Rithesh Murthy, Liangwei Yang, Silvio Savarese, Juan Carlos Niebles, Huan Wang, Shelby Heinecke, and Caiming Xiong. Apigen: automated pipeline for generating verifiable and diverse function-calling datasets. In *Proceedings of the 38th International Conference on Neural Information Processing Systems*, NIPS '24, Red Hook, NY, USA, 2024b. Curran Associates Inc. ISBN 9798331314385.

Lin Long, Rui Wang, Ruixuan Xiao, Junbo Zhao, Xiao Ding, Gang Chen, and Haobo Wang. On LLMs-driven synthetic data generation, curation, and evaluation: A survey. In Lun-Wei Ku, Andre Martins, and Vivek Srikumar (eds.), *Findings of the Association for Computational Linguistics: ACL 2024*, pp. 11065–11082, Bangkok, Thailand, August 2024. Association for Computational Linguistics. doi: 10.18653/v1/2024.findings-acl.658. URL https://aclanthology.org/2024.findings-acl.658/.

Ximing Lu, David Acuna, Jaehun Jung, Jian Hu, Di Zhang, Shizhe Diao, Yunheng Zou, Shaokun Zhang, Brandon Cui, Mingjie Liu, Hyunwoo Kim, Prithviraj Ammanabrolu, Jan Kautz, Yi Dong, and Yejin Choi. Golden goose: A simple trick to synthesize unlimited rlvr tasks from unverifiable internet text, 2026. URL https://arxiv.org/abs/2601.22975.

Pratyush Maini, Skyler Seto, Richard Bai, David Grangier, Yizhe Zhang, and Navdeep Jaitly. Rephrasing the web: A recipe for compute and data-efficient language modeling. In *Proceedings of the 62nd Annual Meeting of the Association for Computational Linguistics (Volume 1: Long Papers)*, pp. 14044–14072, 2024.

Arindam Mitra, Luciano Del Corro, Guoqing Zheng, Shweti Mahajan, Dany Rouhana, Andres Codas, Yadong Lu, Wei ge Chen, Olga Vrousgos, Corby Rosset, Fillipe Silva, Hamed Khanpour, Yash Lara, and Ahmed Awadallah. Agentinstruct: Toward generative teaching with agentic flows, 2024. URL https://arxiv.org/abs/2407.03502.

Mihai Nadăş, Laura Dioşan, and Andreea Tomescu. Synthetic data generation using large language models: Advances in text and code. *IEEE Access*, 13:134615–134633, 2025. ISSN 2169-3536. doi: 10.1109/access.2025.3589503. URL http://dx.doi.org/10.1109/ACCESS.2025.3589503.

Thao Nguyen, Yang Li, Olga Golovneva, Luke Zettlemoyer, Sewoong Oh, Ludwig Schmidt, and Xian Li. Recycling the web: A method to enhance pre-training data quality and quantity for language models. *arXiv preprint arXiv:2506.04689*, 2025.

Joel Niklaus, Atsuki Yamaguchi, Michal Štefánik, Guilherme Penedo, Hynek Kydlíček, Elie Bakouch, Lewis Tunstall, Edward Emanuel Beeching, Thibaud Frere, Colin Raffel, et al. How can we synthesize high-quality pretraining data? a systematic study of prompt design, generator model, and source data. *arXiv preprint arXiv:2604.13977*, 2026.

Nvidia, :, Bo Adler, Niket Agarwal, Ashwath Aithal, Dong H. Anh, Pallab Bhattacharya, Annika Brundyn, Jared Casper, Bryan Catanzaro, Sharon Clay, Jonathan Cohen, Sirshak Das, Ayush Dattagupta, Olivier Delalleau, Leon Derczynski, Yi Dong, Daniel Egert, Ellie Evans, Aleksander Ficek, Denys Fridman, Shaona Ghosh, Boris Ginsburg, Igor Gitman, Tomasz Grzegorzek, Robert Hero, Jining Huang, Vibhu Jawa, Joseph Jennings, Aastha Jhunjhunwala, John Kamalu, Sadaf Khan, Oleksii Kuchaiev, Patrick LeGresley, Hui Li, Jiwei Liu, Zihan Liu, Eileen Long, Ameya Sunil Mahabaleshwarkar, Somshubra Majumdar, James Maki, Miguel Martinez, Maer Rodrigues de Melo, Ivan Moshkov, Deepak Narayanan, Sean Narenthiran, Jesus Navarro, Phong Nguyen, Osvald Nitski,

Vahid Noroozi, Guruprasad Nutheti, Christopher Parisien, Jupinder Parmar, Mostofa Patwary, Krzysztof Pawelec, Wei Ping, Shrimai Prabhumoye, Rajarshi Roy, Trisha Saar, Vasanth Rao Naik Sabavat, Sanjeev Satheesh, Jane Polak Scowcroft, Jason Sewall, Pavel Shamis, Gerald Shen, Mohammad Shoeybi, Dave Sizer, Misha Smelyanskiy, Felipe Soares, Makesh Narsimhan Sreedhar, Dan Su, Sandeep Subramanian, Shengyang Sun, Shubham Toshniwal, Hao Wang, Zhilin Wang, Jiaxuan You, Jiaqi Zeng, Jimmy Zhang, Jing Zhang, Vivienne Zhang, Yian Zhang, and Chen Zhu. Nemotron-4 340b technical report, 2024. URL https://arxiv.org/abs/2406.11704.

Team OLMo, Pete Walsh, Luca Soldaini, Dirk Groeneveld, Kyle Lo, Shane Arora, Akshita Bhagia, Yuling Gu, Shengyi Huang, Matt Jordan, Nathan Lambert, Dustin Schwenk, Oyvind Tafjord, Taira Anderson, David Atkinson, Faeze Brahman, Christopher Clark, Pradeep Dasigi, Nouha Dziri, Michal Guerquin, Hamish Ivison, Pang Wei Koh, Jiacheng Liu, Saumya Malik, William Merrill, Lester James V. Miranda, Jacob Morrison, Tyler Murray, Crystal Nam, Valentina Pyatkin, Aman Rangapur, Michael Schmitz, Sam Skjonsberg, David Wadden, Christopher Wilhelm, Michael Wilson, Luke Zettlemoyer, Ali Farhadi, Noah A. Smith, and Hannaneh Hajishirzi. 2 olmo 2 furious, 2025. URL https://arxiv.org/abs/2501.00656.

Jiayi Pan, Xingyao Wang, Graham Neubig, Navdeep Jaitly, Heng Ji, Alane Suhr, and Yizhe Zhang. Training software engineering agents and verifiers with SWE-gym. In Aarti Singh, Maryam Fazel, Daniel Hsu, Simon Lacoste-Julien, Felix Berkenkamp, Tegan Maharaj, Kiri Wagstaff, and Jerry Zhu (eds.), *Proceedings of the 42nd International Conference on Machine Learning*, volume 267 of *Proceedings of Machine Learning Research*, pp. 47717–47737. PMLR, 13–19 Jul 2025. URL https://proceedings.mlr.press/v267/pan25g.html.

Sung Min Park, Kristian Georgiev, Andrew Ilyas, Guillaume Leclerc, and Aleksander Madry. Trak: Attributing model behavior at scale. In *International Conference on Machine Learning*, pp. 27074–27113. PMLR, 2023.

Guilherme Penedo, Hynek Kydlíček, Anton Lozhkov, Margaret Mitchell, Colin A Raffel, Leandro Von Werra, Thomas Wolf, et al. The fineweb datasets: Decanting the web for the finest text data at scale. *Advances in Neural Information Processing Systems*, 37:30811–30849, 2024.

Pouya Pezeshkpour, Sarthak Jain, Byron C Wallace, and Sameer Singh. An empirical comparison of instance attribution methods for nlp. In *Proceedings of the 2021 Conference of the North American Chapter of the Association for Computational Linguistics: Human Language Technologies*, pp. 967–975, 2021.

Garima Pruthi, Frederick Liu, Satyen Kale, and Mukund Sundararajan. Estimating training data influence by tracing gradient descent. *Advances in Neural Information Processing Systems*, 33: 19920–19930, 2020.

Yujia Qin, Shihao Liang, Yining Ye, Kunlun Zhu, Lan Yan, Yaxi Lu, Yankai Lin, Xin Cong, Xiangru Tang, Bill Qian, et al. Toolllm: Facilitating large language models to master 16000+ real-world apis. In *The twelfth international conference on learning representations*, 2023.

Qwen, :, An Yang, Baosong Yang, Beichen Zhang, Binyuan Hui, Bo Zheng, Bowen Yu, Chengyuan Li, Dayiheng Liu, Fei Huang, Haoran Wei, Huan Lin, Jian Yang, Jianhong Tu, Jianwei Zhang, Jianxin Yang, Jiaxi Yang, Jingren Zhou, Junyang Lin, Kai Dang, Keming Lu, Keqin Bao, Kexin Yang, Le Yu, Mei Li, Mingfeng Xue, Pei Zhang, Qin Zhu, Rui Men, Runji Lin, Tianhao Li, Tianyi Tang, Tingyu Xia, Xingzhang Ren, Xuancheng Ren, Yang Fan, Yang Su, Yichang Zhang, Yu Wan, Yuqiong Liu, Zeyu Cui, Zhenru Zhang, and Zihan Qiu. Qwen2.5 technical report, 2025. URL https://arxiv.org/abs/2412.15115.

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. In *First Conference on Language Modeling*, 2024.

Z. Z. Ren, Zhihong Shao, Junxiao Song, Huajian Xin, Haocheng Wang, Wanjia Zhao, Liyue Zhang, Zhe Fu, Qihao Zhu, Dejian Yang, Z. F. Wu, Zhibin Gou, Shirong Ma, Hongxuan Tang, Yuxuan Liu, Wenjun Gao, Daya Guo, and Chong Ruan. Deepseek-prover-v2: Advancing formal mathematical reasoning via reinforcement learning for subgoal decomposition, 2025. URL https://arxiv.org/abs/2504.21801.

Chantal Shaib, Venkata S Govindarajan, Joe Barrow, Jiuding Sun, Alexa Siu, Byron C Wallace, and Ani Nenkova. Standardizing the measurement of text diversity: A tool and comparative analysis. In Xuebo Liu and Ayu Purwarianti (eds.), *Proceedings of The 14th International Joint Conference on Natural Language Processing and The 4th Conference of the Asia-Pacific Chapter of the Association for Computational Linguistics: System Demonstrations*, pp. 36–46, Mumbai, India, December 2025. Association for Computational Linguistics. ISBN 979-8-89176-301-2. doi: 10.18653/v1/2025.ijcnlp-demo.5. URL https://aclanthology.org/2025.ijcnlp-demo.5/.

Ilia Shumailov, Zakhar Shumaylov, Yiren Zhao, Nicolas Papernot, Ross Anderson, and Yarin Gal. Ai models collapse when trained on recursively generated data. *Nature*, 631(8022):755–759, 2024.

Avi Singh, John D Co-Reyes, Rishabh Agarwal, Ankesh Anand, Piyush Patil, Xavier Garcia, Peter J Liu, James Harrison, Jaehoon Lee, Kelvin Xu, et al. Beyond human data: Scaling self-training for problem-solving with language models. *arXiv preprint arXiv:2312.06585*, 2023.

Weiwei Sun, Haokun Liu, Nikhil Kandpal, Colin Raffel, and Yiming Yang. Enhancing training data attribution with representational optimization, 2025. URL https://arxiv.org/abs/2505.18513.

Rohan Taori, Ishaan Gulrajani, Tianyi Zhang, Yann Dubois, Xuechen Li, Carlos Guestrin, Percy Liang, and Tatsunori B Hashimoto. Stanford alpaca: an instruction-following llama model (2023), 2023.

DeepSeek Team. Deepseek-r1 incentivizes reasoning in llms through reinforcement learning. *Nature*, 645(8081):633–638, 2025. ISSN 1476-4687. doi: 10.1038/s41586-025-09422-z. URL http://dx.doi.org/10.1038/s41586-025-09422-z.

Llama 3 Team. The llama 3 herd of models, 2024. URL https://arxiv.org/abs/2407.21783.

Trieu H Trinh, Yuhuai Wu, Quoc V Le, He He, and Thang Luong. Solving olympiad geometry without human demonstrations. *Nature*, 625(7995):476–482, 2024.

Andrew Trotman, Antti Puurula, and Blake Burgess. Improvements to bm25 and language models examined. In *Proceedings of the 19th Australasian Document Computing Symposium*, ADCS '14, pp. 58–65, New York, NY, USA, 2014. Association for Computing Machinery. ISBN 9781450330008. doi: 10.1145/2682862.2682863. URL https://doi.org/10.1145/2682862.2682863.

Jiachen T Wang and Ruoxi Jia. Data banzhaf: A robust data valuation framework for machine learning. In *International Conference on Artificial Intelligence and Statistics*, pp. 6388–6421. PMLR, 2023.

Jiachen Tianhao 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, 2024.

Xiao Wang, Weikang Zhou, Qi Zhang, Jie Zhou, Songyang Gao, Junzhe Wang, Menghan Zhang, Xiang Gao, Yun Wen Chen, and Tao Gui. Farewell to aimless large-scale pretraining: Influential subset selection for language model. In *Findings of the Association for Computational Linguistics: ACL 2023*, pp. 555–568, 2023a.

Yizhong Wang, Yeganeh Kordi, Swaroop Mishra, Alisa Liu, Noah A Smith, Daniel Khashabi, and Hannaneh Hajishirzi. Self-instruct: Aligning language models with self-generated instructions. In *Proceedings of the 61st annual meeting of the association for computational linguistics (volume 1: long papers)*, pp. 13484–13508, 2023b.

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

Mitchell Wortsman, Peter Liu, Lechao Xiao, Katie Everett, Alexander Alemi, Ben Adlam, John D Co-Reyes, Izzeddin Gur, Abhishek Kumar, Roman Novak, et al. Small-scale proxies for large-scale transformer training instabilities. In *International Conference on Learning Representations*, volume 2024, pp. 49844–49869, 2024.

Mengzhou Xia, Sadhika Malladi, Suchin Gururangan, Sanjeev Arora, and Danqi Chen. Less: Selecting influential data for targeted instruction tuning, 2024. URL https://arxiv.org/abs/2402.04333.

Shitao Xiao, Zheng Liu, Peitian Zhang, Niklas Muennighoff, Defu Lian, and Jian-Yun Nie. C-pack: Packed resources for general chinese embeddings. In *Proceedings of the 47th international ACM SIGIR conference on research and development in information retrieval*, pp. 641–649, 2024.

An Yang, Beichen Zhang, Binyuan Hui, Bofei Gao, Bowen Yu, Chengpeng Li, Dayiheng Liu, Jianhong Tu, Jingren Zhou, Junyang Lin, Keming Lu, Mingfeng Xue, Runji Lin, Tianyu Liu, Xingzhang Ren, and Zhenru Zhang. Qwen2.5-math technical report: Toward mathematical expert model via self-improvement, 2024. URL https://arxiv.org/abs/2409.12122.

An Yang, Anfeng Li, Baosong Yang, Beichen Zhang, Binyuan Hui, Bo Zheng, Bowen Yu, Chang Gao, Chengen Huang, Chenxu Lv, et al. Qwen3 technical report. *arXiv preprint arXiv:2505.09388*, 2025.

Zichun Yu and Chenyan Xiong. Repro: Training language models to faithfully recycle the web for pretraining, 2025. URL https://arxiv.org/abs/2510.10681.

Zichun Yu and Chenyan Xiong. Generating pretraining tokens from organic data for data-bound scaling. *arXiv preprint arXiv:2605.17849*, 2026.

Zichun Yu, Spandan Das, and Chenyan Xiong. Mates: Model-aware data selection for efficient pretraining with data influence models, 2024. URL https://arxiv.org/abs/2406.06046.

Zichun Yu, Fei Peng, Jie Lei, Arnold Overwijk, Wen-tau Yih, and Chenyan Xiong. Data-efficient pretraining with group-level data influence modeling. *arXiv preprint arXiv:2502.14709*, 2025.

Zheng Yuan, Hongyi Yuan, Chengpeng Li, Guanting Dong, Keming Lu, Chuanqi Tan, Chang Zhou, and Jingren Zhou. Scaling relationship on learning mathematical reasoning with large language models, 2023. URL https://arxiv.org/abs/2308.01825.

Eric Zelikman, Yuhuai Wu, Jesse Mu, and Noah Goodman. Star: Bootstrapping reasoning with reasoning. *Advances in Neural Information Processing Systems*, 35:15476–15488, 2022.

Ji Zeng, Dayuan Fu, Tiantian Mi, Yumin Zhuang, Yaxing Huang, Xuefeng Li, Lyumanshan Ye, Muhang Xie, Qishuo Hua, Zhen Huang, Mohan Jiang, Hanning Wang, Jifan Lin, Yang Xiao, Jie Sun, Yunze Wu, and Pengfei Liu. davinci-dev: Agent-native mid-training for software engineering, 2026. URL https://arxiv.org/abs/2601.18418.

Chi Zhang, Huaping Zhong, Kuan Zhang, Chengliang Chai, Rui Wang, Xinlin Zhuang, Tianyi Bai, Qiu Jiantao, Lei Cao, Ju Fan, et al. Harnessing diversity for important data selection in pretraining large language models. In *The Thirteenth International Conference on Learning Representations*, 2025a.

Luyang Zhang, Cathy Jiao, Beibei Li, and Chenyan Xiong. Fairshare data pricing via data valuation for large language models. *arXiv preprint arXiv:2502.00198*, 2025b.

Xuekai Zhu, Daixuan Cheng, Hengli Li, Kaiyan Zhang, Ermo Hua, Xingtai Lv, Ning Ding, Zhouhan Lin, Zilong Zheng, and Bowen Zhou. How to synthesize text data without model collapse? *arXiv preprint arXiv:2412.14689*, 2024.

APPENDIX

A Related Work **16**

B Supplementary Materials **17**

B.1 Oracle Scoring Configuration 17
B.2 Proxy-Matched Diversity Sampling 17
B.3 Data Influence Setup 19
B.4 Data Curation Setup 20
B.5 Data Selection Methods 21
B.6 Group Formation 23
B.7 Diversity Metrics 23
B.8 Budget Analysis 24

C Proofs: Invidividual Proxy Oracle vs Group Oracle **26**

C.1 Setup 26
C.2 Leading-order characterization of the gap 27
C.3 Proof of the exact bound 29

D Additional Figures and Tables **32**

D.1 Synthetic Data Curation Full Results 32
D.2 Group versus Individual Proxy Oracle Gap Across Group Sizes 33
D.3 Cluster Topic Composition 33
D.4 Additional Diversity Plots 34
D.5 Group Oracle Budget Analysis Plots 35

A RELATED WORK

Synthetic data for LLM Training. Synthetic data is now used at every stage of LLM training. In pre-training, it supplements or rewrites web text to raise its information density, either through textbook-style generation (Li et al., 2023; Abdin et al., 2024) or by rephrasing and recycling existing documents (Maini et al., 2024; Nguyen et al., 2025; Yu & Xiong, 2025). In mid-training, it is used to target specific capabilities before post-training, such as long-context, math, and code (Team, 2024; Yang et al., 2024; OLMo et al., 2025; Zeng et al., 2026). In post-training, it supplies instruction-following and reasoning supervision that is difficult to source from humans at scale, through self-generated instructions (Wang et al., 2023b; Taori et al., 2023; Mitra et al., 2024), distillation from larger teachers (Adler et al., 2024; Team, 2025), and self-improvement over verified model outputs (Zelikman et al., 2022; Singh et al., 2023; Yuan et al., 2023). It is also used to construct the environments and tasks that post-training runs over, including tool-use APIs and trajectories (Qin et al., 2023; Liu et al., 2024b) and verifiable tasks for reinforcement learning (Lu et al., 2026).

Synthetic data selection. While past work has shown that individual-level data selection is effective for organic data (including representation-based (Pezeshkpour et al., 2021), gradient-based (Grosse et al., 2023; Kwon et al., 2024; Xia et al., 2024), and lexical methods (Trotman et al., 2014)), synthetic data often exhibits different characteristics. These characteristics include lower information density (Maini et al., 2024; Kang et al., 2025), under-representation of rare entities or domains (Zhu et al.,

2024; Yu & Xiong, 2026), and more templated phrasing (Niklaus et al., 2026). Synthetic data curation therefore requires additional considerations, such as task complexity, environment construction, and model biases (Long et al., 2024; Nadăş et al., 2025). Selection strategies for synthetic data largely score each generation on its own, filtering by quality or reward models (Chen et al., 2024b; Wettig et al., 2024) or by similarity to a target task (Xia et al., 2024; Trotman et al., 2014). A smaller line of work instead scores generations against one another, promoting diversity or limiting redundancy across the retained set (Chen et al., 2024a; Zhu et al., 2024).

Group-Level Data Influence. Since data inclusion decisions are typically made over groups rather than individual samples, a line of work has examined how influence estimates behave at the group level. Group influence is not the sum of its parts: the effects of individual samples can amplify or cancel one another, and the change induced by any single sample is small relative to that of the group it belongs to (Koh et al., 2019; Basu et al., 2020; Hu et al., 2024). The most common approach nonetheless scores a group by aggregating the influence estimates of its samples (Wang et al., 2023a; Zhang et al., 2025a;b). Cooperative-game formulations instead average a sample's marginal contribution across subsets (Ghorbani & Zou, 2019; Jia et al., 2019; Wang & Jia, 2023), though the resulting scores remain per-sample and are again summed to value a group. A third line of work models the group directly, either by predicting model behavior as a function of the training subset (Ilyas et al., 2022; Park et al., 2023; Engstrom et al., 2024) or by estimating influence alongside relational information between samples in the same group (Wang et al., 2024; Sun et al., 2025; Yu et al., 2025).

B Supplementary Materials

This section presents additional details on the experiments from Sections 2, 3, 4, and 5.

B.1 Oracle Scoring Configuration

In this section, we present the parameters for the oracle scoring procedure from Section 3.1, for both the pre-training and post-training settings. Across all oracle scoring settings, we take a candidate group $G = \{z_j\}_{j=1}^{|G|}$ and perform one AdamW step on G, starting from a model with parameters $\theta_{D_{\text{train}}}$ previously obtained by training on D_{train} (in our setting, $\theta_{D_{\text{train}}}$ is a pre-trained model checkpoint). We evaluate both $\theta_{D_{\text{train}}}$ and $\theta_{D_{\text{train}} \cup G}$ on a reference set $D_{\text{ref}} = \{z'_i\}_{i=1}^{M}$. We reset the weights to $\theta_{D_{\text{train}}}$ between candidate groups. The parameter details are shown in Table 3.

Parameter	**Pre-Training Setting**	**Post-training Setting**
Probe model	Repro-400M (trained on 6.59B Repro tokens (Yu et al., 2025))	Qwen2.5-1.5B-Instruct
Tokenizer	Pythia-410m	Qwen2.5-1.5B-Instruct
Optimizer	AdamW, 1 step	AdamW, 1 step
Learning rate	`2e-5`	1e-5
Precision	`bf16`	`bf16`
Grad. accumulation	Yes	Yes
Micro batch size	`2`	`1`
Candidate group sizes	5120, 10240 samples per group	128, 512 samples per group
Sequence length	2048	2048
Reference set	FLAN (128 samples)	GPQA (128 held-out samples)

Table 3: Oracle scoring configurations for the pre-training and post-training settings.

B.2 Proxy-Matched Diversity Sampling

This section details the sampling procedure summarized in Section 3.2. The goal is to create a pool of groups in which every group falls within an individual proxy oracle score range, and diversity is the free variable. We describe the process below.

Setting	size	Quantity	$\max_j \lvert\mathcal{I}_{\text{proxy}}(G_j,\theta)-\tau_j\rvert$	ϵ	diversity range
Pretraining	5120	1000	5.0×10^{-5}	5×10^{-5}	0.095–0.212
Pretraining	10240	500	5.0×10^{-5}	5×10^{-5}	0.096–0.200
Post-training	128	1000	5.0×10^{-6}	5×10^{-6}	0.156–0.435
Post-training	512	500	4.8×10^{-6}	5×10^{-6}	0.157–0.425

Table 4: Diversity-swept candidate pools. Diversity is $1-\overline{\cos}(G)$. Every group's individual proxy oracle lands within ϵ of its assigned band target, so the pools vary in diversity at (near-)fixed individual proxy oracle.

Embedding and clustering. Given a large corpus, D_{train}, we first partition D_{train} into multiple clusters. The cluster partition $\mathcal{C}=\{C_1,\ldots,C_k\}$ that defines coverage is built once per setting, before any group creation. We embed each example with BGE-base-en-v1.5 (Xiao et al., 2024). BGE accepts at most 512 tokens whereas our sequences are 2048 tokens long, so each sequence is split into four consecutive 512-token chunks, each chunk is encoded independently, and we take the 768-dimensional [CLS] hidden state as the chunk embedding. The four chunk embeddings are averaged to give one 768-dimensional vector per sequence. We then partition the resulting matrix with k-means at $k=20$, using Euclidean distance on the embeddings. Figure 10 shows the topics covered by the resulting clusters for each corpus.

Band selection. We first score a reference pool of 100 uniformly-random groups from D_{train} and take the central half of their individual proxy-oracle distribution as the band $[\tau_{\text{lo}},\tau_{\text{hi}}]$. For the pre-training setting this gives $[-6.85,-6.57]\times10^{-3}$ (center $\tau=-6.71\times10^{-3}$, width 2.8×10^{-4}); for the post-training setting, $[-8.7,-8.1]\times10^{-4}$ (center $\tau=-8.4\times10^{-4}$, width 6×10^{-5}). Restricting to the central half rather than the full reference range keeps every group's individual proxy oracle inside the distribution a random selector would produce, so the pool is not itself an outlier population.

Coverage schedule. Each group G_j, where $j=1,\ldots,H$, is assigned a coverage level

$$r_j=\text{round}\Big(\exp\Big(\log r_{\min}+\tfrac{j-1}{H-1}\left(\log r_{\max}-\log r_{\min}\right)\Big)\Big),\qquad r_{\min}=1,\ r_{\max}=k=20.$$

Target assignment and decorrelation. We form the grid $\beta=\text{linspace}(\tau_{\text{lo}},\tau_{\text{hi}},H)$ and set $\tau_j=\beta_{\pi(j)}$ for a seeded uniform random permutation π.

Member selection. Given (r_j,τ_j), we allocate the group's examples across its r_j clusters and select within each cluster by *stratified two-sided sampling*: within a cluster, examples are sorted by individual proxy score, split at τ_j, and drawn from both sides in the mixing proportion that makes the subset mean equal τ_j, with picks spread uniformly over each side's rank range. This deliberately allows a *wide* spread of individual scores while pinning their mean to τ_j. Table 4 reports the four pools. In every case the realized individual proxy-oracle residual is within tolerance for all H groups.

B.3 Data Influence Setup

Table 5 lists the full settings for the pre-training and post-training experiments in Section 3.2, organized by phase. Each training setting is run under two configurations that differ in group size. In the *small*-group configuration, the candidate pool contains more, smaller groups (H=1000), and more groups are selected (k=40 for pre-training, k=25 for post-training). In the *large*-group configuration, the pool contains fewer, larger groups (H=500), and fewer groups are selected (k=20 and k=7).

Group sizes are integer multiples of the training batch size, so each group spans a whole number of optimizer steps (10 or 20 for pre-training, 4 or 16 for post-training). The base model is used both for oracle scoring (Phase 2) and as the training initialization (Phase 3).

Table 5: Task settings for the pre-training and post-training experiments, organized by the three phases in Section 3.2. Each training setting is run under two configurations: a *small*-group and a *large*-group setting.

	Pre-training	**Post-training**
Phase 1: Candidate pool construction		
Source dataset	Repro-Rephrased-72B	GooseReason
Semantic clusters	20	20
Group sizes s	5120 (small), 10240 (large)	128 (small), 512 (large)
Candidate pool H	1000 (small), 500 (large)	1000 (small), 500 (large)
Phase 2: Data selection		
Reference set	FLAN (128 samples)	GPQA-extend (128 samples)
Selected groups k	40 (small), 20 (large)	25 (small), 7 (large)
Gumbel Temperature λ	0.5	0.5
Phase 3: Training and evaluation		
Base model M_{θ_0}	Repro-400M checkpoint	Qwen2.5-1.5B-Instruct
Batch size	512 sequences	32 prompts ($\times$8 completions)
Optimizer	cosine (wd 0.033, no warmup)	GRPO (warmup 0.05, β=0, temp 0.6)
Learning rate	3×10^{-3}	1×10^{-6}
Training steps	400	100
Benchmarks	ARC-C, CSQA, CoQA, HellaSwag, LAMBADA, MMLU, PIQA, SQuAD	MBPP, MBPP+, GPQA-D, BBH, IFEval, AIME24

B.4 Data Curation Setup

Table 6 lists the full settings for the data curation experiments in Section 4, organized by phase.

Table 6: Experimental settings for the data curation experiments, organized by the three phases in Section 4.1. The pool is held at a fixed size while the rephrase factor r controls the degree of near-duplicate redundancy.

Phase 1: Synthetic pool construction	
Source documents	Organic web text (DCLM-RefinedWeb)
Rephraser	Repro rephraser-4B (Yu & Xiong, 2025)
Sampling	temperature 1.0, top-p 0.9
Pool size N	716,800 sequences
Rephrase factor r	$\{1, 10, 50\}$
Unique source docs (N/r)	716,800 / 71,680 / 14,336
Sequence length	2048 tokens
Phase 2: Data selection	
Selection methods	Random, GradSim, FineWeb-Edu, BM25, GMRel, GREATS
Reference set	FLAN (128 samples) (Chung et al., 2024)
Embedding model $\phi(\cdot)$	BGE
Clusters	30 (k-means), largest 20 retained
Group size s	5,120 sequences
Groups per run	20 (one per retained cluster), 102,400 sequences
Phase 3: Training and evaluation	
Base model M_{θ_0}	Repro-400M (Yu & Xiong, 2025)
Batch size	512 sequences
Training steps	200 (one epoch over the selection)
Learning rate	3×10^{-3} (cosine, no warmup)
Weight decay	0.033
Seed	124
Benchmarks	ARC-C, CSQA, CoQA, HellaSwag, LAMBADA, MMLU, PIQA, SQuAD

B.5 Data Selection Methods

This section presents additional details regarding the data selection methods from Section 4.1.

B.5.1 FineWeb-Edu

The FineWeb-Edu (Penedo et al., 2024) classifier is a lightweight encoder-based regressor (a BERT-based encoder with a single-output regression head) that rates a document's educational value from 0 to 5. We score every sequence in the pool independently. The classifier accepts at most 512 tokens. Therefore, for a given data sample, we score each chunk, and average the chunk scores to obtain a single score per sequence. For groups of samples, we average the individual sequence scores to obtain a single score per group.

B.5.2 BM25

BM25 (Trotman et al., 2014) is a standard information retrieval method that ranks training samples based on lexical similarity to a query. The reference set sequences are decoded to text and whitespace-tokenized to form a BM25-Okapi document collection of documents, built once; each candidate sequence is then decoded and whitespace-tokenized in the same way and issued as a query against that collection. The score of a sequence is the mean of its BM25 scores, so higher scores indicate greater lexical relevance to the reference set and sequences are again ranked in descending order.

B.5.3 GradSim

GradSim (Pruthi et al., 2020) estimates influence with the first-order inner product between the gradient of the loss on a training sample and the gradient of the loss on the reference set. We first compute a single reference gradient by back-propagating the token-weighted mean loss over the reference set at the base checkpoint θ; the base checkpoint is the same one used for oracle scoring in each setting (Table 3). We then compute, for each candidate sequence z, the gradient $\nabla_\theta \mathcal{L}(z, \theta)$ at that same checkpoint. Each candidate is scored by the cosine similarity between its gradient and the reference gradient.

B.5.4 GMRel

GMRel (Yu et al., 2025) trains a relational data influence model that estimates oracle data influence by weighting individual influence scores according to relationships among training data. We train one relational model per task setting, in three stages.

Architecture. The relational model is a BGE-base-en-v1.5 encoder $\psi(\cdot)$ with a one-dimensional linear head $h(\cdot)$, plus two scalars learned jointly with the encoder: a relationship temperature τ and a relational weight α. A sequence's representation $\psi(z)$ is the ℓ_2-normalized `[CLS]` state of each of its 512-token chunks, averaged over the four chunks of a 2048-token sequence, and its individual influence is $\hat{I}(z) = h(\psi(z))$. For an ordered group $G = (z_1, \ldots, z_n)$, the *relationship* between two members is the temperature-scaled representation inner product

$$R(z_i, z_j) \;=\; \frac{\langle \psi(z_i), \psi(z_j) \rangle}{\tau},$$

and the group-aware influence of the t-th member is its individual influence modulated by its average relationship to the members preceding it,

$$\hat{I}_{\mathrm{rel}}\big(z_t \mid z_1, \ldots, z_{t-1}\big) \;=\; \alpha\, \hat{I}(z_t) \left(1 - \frac{1}{t-1} \sum_{i<t} R(z_t, z_i)\right), \qquad t \geq 2,$$

with $\hat{I}_{\mathrm{rel}}(z_1) = \hat{I}(z_1)$ for the first member.

Training data. The relational model is trained on *oracle rollouts* generated on the same base model used for oracle scoring in that setting. In the pre-training setting, rollout candidates are drawn from the tokenized Repro-Rephrased-72B corpus (excluding data used for training in Section 4) (Yu & Xiong, 2025), such that each candidate contains four 2048-token sequences, giving $800{,}000$ candidate sequences; the reference set is the same 128-sample FLAN set used for oracle scoring.

In the post-training setting, candidates are drawn from GooseReason (Lu et al., 2026) against the 128-sample GPQA-extend reference set.

Each rollout samples $K = 10$ candidate sequences without replacement, resets the model to its base weights, and then trains on the candidates *sequentially* with AdamW, re-evaluating reference-set loss after each one. The recorded label for a candidate is the cumulative change in reference loss. We collect 2,400 rollouts (24,000 labeled candidates) in the pre-training setting and 800 rollouts (8,000 labeled candidates) in the post-training setting.

Training parameters. The relational model is fit by MSE between its group-aware scores $\hat{I}_{\text{rel}}$ and these per-step deltas, holding out 200 rollouts for evaluation. We select the best checkpoint by Spearman correlation on the held-out rollouts. Table 7 lists the full configuration for both the rollout generation and the model fitting.

Parameter	Pre-Training Setting	Post-training Setting
Rollout generation (supervision)		
Base model	Repro-400M (as in Table 3)	Qwen2.5-1.5B-Instruct
Candidate pool	Repro-Rephrased-72B	GooseReason
Source tokenizer	Pythia-410m	Qwen2.5-1.5B-Instruct
Reference set	FLAN (128 samples)	GPQA-extend (128 samples)
Candidates per rollout K	10	10
Inner optimizer	AdamW, 4 steps per candidate	AdamW, 1 step per candidate
Inner learning rate	`2e-5`	`1e-5`
(β_1, β_2), weight decay	$(0.9, 0.95)$, 0	$(0.9, 0.999)$, 0
Gradient clipping	1.0	none
Precision	`bf16`	`bf16`
Rollouts collected	2,400 (24,000 labeled candidates)	800 (8,000 labeled candidates)
Relational model fitting		
Encoder	BGE-base-en-v1.5	BGE-base-en-v1.5
Chunking per sequence	4×512 tokens	4×512 tokens
Objective	MSE on per-step reference-loss deltas	MSE on per-step reference-loss deltas
Learning rate	`5e-5`	`5e-5`
Weight decay	0.01	0.01
Epochs	5	5
Batch size	1, gradient accumulation 4	1, gradient accumulation 4
Warmup steps	50	50
Precision	`bf16`	`bf16`
Held-out rollouts	200	200

Table 7: GMRel relational model configurations: oracle rollout generation (which produces the supervision) and fitting of the relational influence model.

At selection time each pool sequence's representation and individual influence are computed once and cached, and the greedy of Appendix B.6 uses the checkpoint's learned scalars ($\alpha \approx 0.996$, $\tau \approx 1.010$ for the pre-training model). Because the labels are reference-loss changes, more useful data has a *more negative* score and the greedy selects the arg min. To vary relational strength we additionally form groups with τ overridden to 0.30, roughly tripling the weight on the relational term; this is the comparison in Figure 5d.

B.5.5 GREATS

GREATS (Wang et al., 2024) augments the GradSim individual score with a penalty against the sequences already selected. Writing $g_z = \nabla_\theta \mathcal{L}(z, \theta)$ and g_{ref} for the reference gradient above, the individual term is $s_z = \langle g_z, g_{\text{ref}} \rangle$ and the marginal gain of adding z to a partially formed group G is

$$\phi_z = s_z - \eta \Big\langle g_z, \sum_{z^* \in G} g_{z^*} \Big\rangle.$$

Selection starts from $G = \emptyset$, takes $\arg\max_z s_z$ in the first round, and repeatedly adds $\arg\max_z \phi_z$, so a candidate that points in the same gradient direction as the current members is discounted even if its individual score is high. Because ϕ_z has to be recomputed against G at every round, we cache each sequence's gradient once as a projection to $d = 8192$ dimensions (correlation ≥ 0.9999 with the exact gradient inner products). We set step size η to the training learning rate ($\eta = 0.003$). The gradients are the same per-sequence gradients GradSim selects on, so the two methods differ only in the selection rule.

B.6 Group Formation

In this section, we describe the group formation procedure used in Section 4.1. For each data selection method considered in that section, we construct groups using a *cluster-based inference* procedure (Yu et al., 2025). We adopt this procedure to reduce computational cost, since some data selection methods (e.g., GMRel (Yu et al., 2025) and GREATS (Wang et al., 2024)) require iterative rounds of scoring over the entire dataset.

Specifically, for each synthetic dataset from Section 4, we embed every sequence with BGE-base-en-v1.5 (Xiao et al., 2024) and partition the pool into 30 clusters via k-means. We then sample 20 clusters and use each data selection method to form one group of size $s = 5120$ from each sampled cluster, yielding 20 groups per method.

Individual-level scorers (i.e., GradSim (Pruthi et al., 2020), FineWeb-Edu (Penedo et al., 2024), and BM25 (Trotman et al., 2014)) form each group greedily by taking the top-scored samples within a cluster. GMRel (Yu et al., 2025) and GREATS (Wang et al., 2024) instead form each group through a relational greedy process: starting from an empty group, they repeatedly add the candidate with the highest score after *re-scoring against the partially formed group*, so that each addition accounts for its relation to the samples already selected, as described in Appendix B.5.

B.7 Diversity Metrics

In this section, we describe the diversity metrics evaluated in Section 5. Each metric maps a group to a scalar, and in every case a larger value indicates a more diverse (less redundant) group. We then report the Spearman rank correlation between each metric and the oracle gap $\left| \mathcal{I}_{\text{proxy}}(G, \theta) - \mathcal{I}_{\text{orcl}}(G, \theta) \right|$ over the groups in each setting.

Gradient diversity. This is the gradient-based metric. For each sample z we take the per-sample gradient $g_z = \nabla_\theta \mathcal{L}(z, \theta)$ at the same base checkpoint used for oracle scoring (Table 3), and JL-project it to $d = 8192$ dimensions with the TRAK Rademacher projector (Park et al., 2023), as in Appendix B.5. Writing $u_z = p_z / \|p_z\|$ for the unit-normalized projected gradients, the metric is one minus the mean pairwise cosine similarity,

$$\text{GradDiv}(G) \;=\; 1 - \frac{1}{|G|(|G|-1)} \sum_{z \neq z'} \langle u_z, u_{z'} \rangle.$$

We use the cosine form so that the metric depends only on gradient *directions* and is not dominated by a few high-magnitude samples.

Vendi score. The Vendi score (Friedman & Dieng, 2022) is an embedding-based metric that measures the *effective number of distinct samples* in a group. We embed each sample with BGE-base-en-v1.5 (Xiao et al., 2024) (averaging over a sample's sequences) and ℓ_2-normalize, giving $X \in \mathbb{R}^{|G| \times 768}$. Letting $\lambda_1, \ldots, \lambda_{|G|}$ be the eigenvalues of the normalized cosine-similarity kernel $K = \frac{1}{|G|} X X^\top$, the score is the exponential of the Shannon entropy of the eigenspectrum,

$$\text{Vendi}(G) \;=\; \exp\Big(- \sum_i \lambda_i \log \lambda_i \Big).$$

It ranges from 1 (all samples identical) to $|G|$ (mutually orthogonal embeddings), and unlike a mean pairwise similarity it accounts for higher-order redundancy, such as several tight clusters of near-duplicates.

Compression ratio. This is the information-theoretic metric, and captures redundancy *across* samples that a compressor can exploit (Shaib et al., 2025). Writing b_z for the raw byte string of sample z and $\mathrm{zlib}(\cdot)$ for the compressed length at level 6, we take the ratio of the compressed length of the concatenated group to the sum of the individually compressed lengths,

$$\mathrm{CompRatio}(G) \;=\; \frac{\big|\mathrm{zlib}(b_{z_1} \| \cdots \| b_{z_{|G|}})\big|}{\sum_{z \in G} \big|\mathrm{zlib}(b_z)\big|}.$$

A value near 1 means concatenating the samples yields no compression gain, i.e. they share little material; values well below 1 indicate that samples are largely restatements of one another.

Distinct n-grams. These are the token-level metrics. We pool the token sequences of all samples in the group (stripping padding), enumerate all token n-grams, and report the fraction that are unique,

$$\text{Distinct-}n(G) \;=\; \frac{\big|\{\text{unique } n\text{-grams in } G\}\big|}{\big|\{\text{all } n\text{-grams in } G\}\big|}.$$

We report Distinct-2 and Distinct-3. Because n-grams are pooled across the whole group rather than computed per sample, the metric reflects repetition between samples and not just within them.

B.8 Budget Analysis

This section details the memory computation from Section 5. We consider a pool of data groups $\mathcal{G} = \{G_1, \ldots, G_H\}$, where each group contains $|G|$ sequences, and an allocation that audits B of the H groups with the group oracle. We report *cumulative* GPU memory, i.e., memory integrated over the run (GiB·hours). A run has two phases, group-oracle calls and per-sample gradient passes, so its cumulative memory is

$$\mathrm{Mem} \;=\; \underbrace{\mathrm{mem}_{\mathrm{orcl}} \times T_{\mathrm{orcl}}}_{\text{oracle phase}} \;+\; \underbrace{\mathrm{mem}_{\mathrm{grad}} \times T_{\mathrm{grad}}}_{\text{gradient phase}}, \tag{6}$$

where mem is the GPU memory resident during a phase and T is the time spent in it. Since both phases are compute-bound on the same GPU, we substitute FLOPs for time. We detail the FLOPs analysis in Section B.8.1 and the cumulative memory analysis in Section B.8.2.

B.8.1 FLOPs Analysis

We first define the following variables for FLOPs analysis:

- F_{fwd}: FLOPs for one forward pass of the probe model over one sequence. We report all FLOPs in units of F_{fwd}.
- $F_{\mathrm{bwd}} \approx 2F_{\mathrm{fwd}}$: FLOPs for one backward pass.
- R: number of sequences in the reference set D_{ref}.

Section 5 applies two operations to a group G. A *group-oracle call* (Appendix B.1) performs one AdamW step over G and then evaluates the updated model on D_{ref}. A *gradient pass* computes the per-sample gradient g_z of every $z \in G$, each requiring one forward and backward pass. Their FLOPs are

$$\begin{aligned} F_{\mathrm{orcl}} \;&=\; \underbrace{|G|\,(F_{\mathrm{fwd}} + F_{\mathrm{bwd}})}_{\text{AdamW step over } G} \;+\; \underbrace{R\,F_{\mathrm{fwd}}}_{\text{reference evaluation}} \;=\; 3|G| + R, \\ F_{\mathrm{grad}} \;&=\; \underbrace{|G|\,(F_{\mathrm{fwd}} + F_{\mathrm{bwd}})}_{\text{per-sample gradients over } G} \;=\; 3|G|. \end{aligned}$$

A gradient pass over a fraction q of the group, as in $\mathrm{GradDiv}_q$, costs $q\,F_{\mathrm{grad}}$.[1]

[1] $\mathrm{GradDiv}_q$ samples $\lceil q|G| \rceil$ sequences; the rounding does not affect any reported value.

B.8.2 CUMULATIVE MEMORY ANALYSIS

Next, we define the following variables for memory analysis:

- W: GPU memory of one copy of the model parameters.
- A: GPU memory of the activations of one micro-batch.

A group-oracle call holds the weights, the gradient, and the two AdamW moments, plus activations. A gradient pass holds only the weights and the gradient, plus activations; no optimizer state is instantiated. The resident memory of each phase is therefore

$$\begin{aligned}\mathrm{mem}_{\mathrm{orcl}} &= \underbrace{W}_{\text{weights}} + \underbrace{W}_{\text{gradient}} + \underbrace{2W}_{\text{AdamW moments}} + \underbrace{A}_{\text{activations}} = 4W + A,\\ \mathrm{mem}_{\mathrm{grad}} &= \underbrace{W}_{\text{weights}} + \underbrace{W}_{\text{gradient}} + \underbrace{A}_{\text{activations}} = 2W + A,\end{aligned}$$

and we write $\kappa = \mathrm{mem}_{\mathrm{grad}}/\mathrm{mem}_{\mathrm{orcl}} = (2W+A)/(4W+A)$ for the memory discount of a gradient pass relative to an oracle call.

Every allocation in Section 5 audits B groups with the group oracle and runs a gradient pass over a fraction q of each of the H groups, so all of them share one cost expression:

$$\mathrm{Mem}(B,q) = \underbrace{(4W+A)\,B\,F_{\mathrm{orcl}}}_{\text{oracle calls}} + \underbrace{(2W+A)\,H\,q\,F_{\mathrm{grad}}}_{\text{gradient passes}}. \tag{7}$$

The allocations differ only in q:

- **Group oracle.** Every group receives an oracle call and no gradients are computed.
- **Random.** No diagnostic is computed, and the allocation is charged only for its oracle calls.
- **GradDiv$_{\text{full}}$.** All per-sample gradients of every group are computed to rank the groups; the top B are audited and the rest are scored with GradSim, which reuses the same gradients.
- **GradDiv$_q$.** As above, with a uniform subsample of a fraction q of each group.

Ratio to the group oracle. Figure 7d plots each allocation's cumulative memory relative to the group oracle, $\mathrm{Mem}_{\mathrm{orcl}} = \mathrm{Mem}(H, 0)$.

Proposition B.1 (Cumulative memory ratio)**.** *For an allocation that audits B groups and runs a gradient pass over a fraction q of each group,*

$$\frac{\mathrm{Mem}(B,q)}{\mathrm{Mem}_{\mathrm{orcl}}} = \frac{B}{H} + \kappa\, q\, \frac{3|G|}{3|G|+R} \approx \frac{B}{H} + \kappa\, q, \tag{8}$$

where the approximation holds when $R \ll 3|G|$.

Proof. Substituting $F_{\mathrm{orcl}} = 3|G| + R$ and $F_{\mathrm{grad}} = 3|G|$ into Equation (7), and dividing by $\mathrm{Mem}_{\mathrm{orcl}} = (4W + A)\,H\,(3|G| + R)$,

$$\begin{aligned}\frac{\mathrm{Mem}(B,q)}{\mathrm{Mem}_{\mathrm{orcl}}} &= \frac{(4W+A)\,B\,(3|G|+R) + (2W+A)\,H\,q\,3|G|}{(4W+A)\,H\,(3|G|+R)}\\ &= \frac{B}{H} + \frac{2W+A}{4W+A}\cdot q\,\frac{3|G|}{3|G|+R},\end{aligned}$$

and the middle factor is κ by definition. Only the ratio of the two memory levels enters, so the result does not depend on the absolute model size or GPU. When $R \ll 3|G|$, the last fraction is ≈ 1.

□

Table 8 summarises the resulting ratios.

Allocation	B	q	Ratio (Eq. 8)	Calculated Ratio
Group oracle	H	0	1	1.000
GradDiv$_{\text{full}}$	B	1	$\approx B/H + \kappa$	0.790
GradDiv$_q$	B	0.5		0.495
GradDiv$_q$	B	0.1	$\approx B/H + \kappa q$	0.259
GradDiv$_q$	B	0.05		0.229
GradDiv$_q$	B	0.01		0.206
Random	B	0	B/H	0.200

Table 8: Cumulative GPU memory of each allocation relative to the full group oracle. The last column is the exact ratio of Equation (8) on the pre-training pool of Figure 7d ($H = 1000$, $|G| = 5120$, $R = 128$, $B/H = 0.2$, $\kappa = 0.593$).

C Proofs: Invidividual Proxy Oracle vs Group Oracle

In this section, we prove Theorem 5.1 and analyze the gap between the two ways of assigning an oracle score to a group: averaging the per-example oracles, and scoring the group jointly.

C.1 Setup

We adopt the notation of Section 2 and use the following quantities throughout:

$g_z = \nabla_\theta \ell(z;\theta)$	gradient of the training loss on example z at θ
$g_G = \frac{1}{\lvert G\rvert}\sum_{z\in G} g_z$	mean gradient of the group
$H = \nabla^2 L(\theta)$	Hessian of the reference loss at θ
$\Sigma_G = \frac{1}{\lvert G\rvert}\sum_{z\in G}(g_z - g_G)(g_z - g_G)^\top$	within-group gradient covariance
$\sigma_G^2 = \frac{1}{\lvert G\rvert}\sum_{z\in G}\lVert g_z - g_G\rVert^2 = \mathrm{Tr}(\Sigma_G)$	within-group gradient diversity
β	smoothness constant of L ($\lVert H\rVert \le \beta$)

Let θ denote the model parameters, D_{ref} the reference set, and $G = \{z_1, \dots, z_{|G|}\}$ a group of training examples. Recall from Theorem 2.1 in Section 2 the oracle scores for a data sample z and a data group G:

$$\text{(individual)}\quad \mathcal{I}_{\text{orcl}}(z,\theta) = \mathcal{L}\big(D_{\text{ref}} \mid \theta_{D_{\text{train}}\cup\{z\}}\big) - \mathcal{L}(D_{\text{ref}} \mid \theta_{D_{\text{train}}})$$
$$\text{(group)}\quad \mathcal{I}_{\text{orcl}}(G,\theta) = \mathcal{L}(D_{\text{ref}} \mid \theta_{D_{\text{train}}\cup G}) - \mathcal{L}(D_{\text{ref}} \mid \theta_{D_{\text{train}}})$$

which represents the change in model performance on reference set D_{ref} when a training sample z or group G is included versus excluded from the training set.

Next, we introduce a helper function to capture the loss-change via parameter displacement.

Loss-Change Function: *Let d denote the number of model parameters, so that $\theta, g_z, g_G \in \mathbb{R}^d$. For a parameter displacement $v \in \mathbb{R}^d$, we define the loss-change function as:*

$$\Phi(v) \overset{def}{=} L(\theta - v) - L(\theta), \qquad \Phi : \mathbb{R}^d \to \mathbb{R}, \tag{9}$$

We rewrite the definitions in Theorem 2.1 above by abbreviating the reference loss as $L(\theta) \overset{\text{def}}{=} \mathcal{L}(D_{\text{ref}} \mid \theta)$, and including the gradient steps $\theta \to \theta - \eta g_z$ and $\theta \to \theta - \eta g_G$:

$$\begin{aligned}\mathcal{I}_{\text{orcl}}(z,\theta) &= L(\theta - \eta g_z) - L(\theta)\\ \mathcal{I}_{\text{orcl}}(G,\theta) &= L(\theta - \eta g_G) - L(\theta)\end{aligned}$$

Furthermore, the individual proxy oracle of Theorem 2.2 in Section 2 becomes:

$$\mathcal{I}_{\text{proxy}}(G,\theta) = \frac{1}{|G|}\sum_{z\in G} L(\theta - \eta g_z) - L(\theta).$$

With the loss-change function, for $\mathcal{I}_{\text{proxy}}$ and $\mathcal{I}_{\text{orcl}}$ the relevant displacements are the gradients $\eta g_z \in \mathbb{R}^d$ and $\eta g_G \in \mathbb{R}^d$, respectively. Thus, $\mathcal{I}_{\text{proxy}}$ and $\mathcal{I}_{\text{orcl}}$ are written as:

$$\boxed{\mathcal{I}_{\text{proxy}}(G,\theta) = \frac{1}{|G|}\sum_{z\in G}\Phi(\eta g_z)} \tag{10}$$

$$\boxed{\mathcal{I}_{\text{orcl}}(G,\theta) = \Phi(\eta g_G).} \tag{11}$$

Thus $\mathcal{I}_{\text{proxy}}(G,\theta)$ is the average of Φ over the points $\{\eta g_z\}$, while $\mathcal{I}_{\text{orcl}}(G,\theta)$ is Φ evaluated at the mean group gradient g_G.

C.2 Leading-order characterization of the gap

In this section, we characterize the difference between $\mathcal{I}_{\text{proxy}}$ and $\mathcal{I}_{\text{orcl}}$. We first identify the leading-order behavior of the gap via a Taylor expansion of Φ about $v = 0$, then promote it to an exact bound in Theorem 5.1.

Proposition C.1 (Leading-order gap)**.** *Let η be the gradient-step size, $H = \nabla^2 L(\theta)$ the Hessian of the reference loss at θ, and Σ_G the within-group gradient covariance. Then the leading-order gap between $\mathcal{I}_{\text{orcl}}(G,\theta)$ and $\mathcal{I}_{\text{proxy}}(G,\theta)$ is*

$$\mathcal{I}_{\text{proxy}}(G,\theta) - \mathcal{I}_{\text{orcl}}(G,\theta) \;=\; \frac{\eta^2}{2}\,\operatorname{Tr}\big(H\,\Sigma_G\big) \;+\; O(\eta^3). \tag{12}$$

In particular, the individual proxy oracle $\mathcal{I}_{\text{proxy}}$ and the true group oracle $\mathcal{I}_{\text{orcl}}$ agree to first order in η, and their leading discrepancy is a second-order effect that grows with both the loss curvature H and the within-group gradient diversity Σ_G.

Proof. We begin by expanding $\mathcal{I}_{\text{orcl}}(G,\theta)$ and $\mathcal{I}_{\text{proxy}}(G,\theta)$ using Lemma C.2.

Step 1. Expanding $\mathcal{I}_{\text{orcl}}(G,\theta)$. Using Lemma C.2 we can write Equation (11) as:

$$\begin{aligned}
\mathcal{I}_{\text{orcl}}(G,\theta) \;&=\; \Phi(\eta g_G) \\
&=\; -\nabla L(\theta)^\top(\eta g_G) \;+\; \tfrac{1}{2}(\eta g_G)^\top H(\eta g_G) \;+\; O(\|\eta g_G\|^3) \quad \text{(Lemma C.2)} \\
&=\; -\eta\,\nabla L(\theta)^\top g_G \;+\; \frac{\eta^2}{2}\,g_G^\top H g_G \;+\; O(\eta^3).
\end{aligned} \tag{13}$$

Step 2. Expanding $\mathcal{I}_{\text{proxy}}(G,\theta)$. Similarly, using Lemma C.2 we can write Equation (10) as:

$$\begin{aligned}
\mathcal{I}_{\text{proxy}}(G,\theta) \;&=\; \frac{1}{|G|}\sum_{z\in G}\Phi(\eta g_z) \\
&=\; \frac{1}{|G|}\sum_{z\in G}\left(-\eta\,\nabla L(\theta)^\top g_z \;+\; \frac{\eta^2}{2}\,g_z^\top H g_z \;+\; O(\eta^3)\right) \quad \text{(Lemma C.2)} \\
&=\; -\eta\,\nabla L(\theta)^\top g_G \;+\; \frac{\eta^2}{2}\cdot\frac{1}{|G|}\sum_{z\in G} g_z^\top H g_z \;+\; O(\eta^3).
\end{aligned} \tag{14}$$

To obtain the gap between $\mathcal{I}_{\text{proxy}}(G,\theta)$ and $\mathcal{I}_{\text{orcl}}(G,\theta)$, we subtract their expanded forms.

Step 3. Gap between $\mathcal{I}_{\mathrm{proxy}}(G,\theta)$ and $\mathcal{I}_{\mathrm{orcl}}(G,\theta)$. Subtracting the group-oracle expansion (Equation 13) from the individual proxy-oracle expansion (Equation 14) , we have:

$$\begin{aligned}\mathcal{I}_{\mathrm{proxy}}(G,\theta) - \mathcal{I}_{\mathrm{orcl}}(G,\theta) \;&=\; \left(-\eta\,\nabla L(\theta)^\top g_G \;+\; \frac{\eta^2}{2}\cdot\frac{1}{|G|}\sum_{z\in G} g_z^\top H g_z \;+\; O(\eta^3)\right)\\ &\quad -\; \left(-\eta\,\nabla L(\theta)^\top g_G \;+\; \frac{\eta^2}{2}\, g_G^\top H g_G \;+\; O(\eta^3)\right)\\ &=\; \frac{\eta^2}{2}\cdot\frac{1}{|G|}\sum_{z\in G} g_z^\top H g_z \;-\; \frac{\eta^2}{2}\, g_G^\top H g_G \;+\; O(\eta^3)\\ &=\; \frac{\eta^2}{2}\left[\frac{1}{|G|}\sum_{z\in G} g_z^\top H g_z \;-\; g_G^\top H g_G\right] + O(\eta^3), \end{aligned} \tag{15}$$

Note that the linear terms $-\eta\,\nabla L(\theta)^\top g_G$ are identical in both expansions (i.e., the two quantities agree to first order) and therefore cancel.

Step 4. Within-group gradient variance. Next, we simplify the bracketed term in Equation (15). We begin by writing each gradient as the group mean plus a deviation,

$$g_z \;=\; g_G + \delta_z, \qquad \delta_z \overset{\text{def}}{=} g_z - g_G, \tag{16}$$

where, by the definition $g_G \;=\; \frac{1}{|G|}\sum_{z\in G} g_z$, the deviations necessarily sum to zero: $\frac{1}{|G|}\sum_{z\in G}\delta_z = g_G - g_G = 0$, a fact we use below.

Next, we expand $g_z^\top H g_z$ using the symmetry of H (note: the Hessian $H = \nabla^2 L(\theta)$ is symmetric because L is twice continuously differentiable),

$$\begin{aligned} g_z^\top H g_z \;&=\; (g_G+\delta_z)^\top H (g_G+\delta_z)\\ &=\; g_G^\top H g_G \;+\; g_G^\top H\delta_z \;+\; \delta_z^\top H g_G \;+\; \delta_z^\top H\delta_z\\ &=\; g_G^\top H g_G \;+\; 2\,\delta_z^\top H g_G \;+\; \delta_z^\top H\delta_z. \end{aligned} \tag{17}$$

Averaging over $z\in G$, we obtain:

$$\begin{aligned}\frac{1}{|G|}\sum_{z\in G} g_z^\top H g_z \;&=\; g_G^\top H g_G \;+\; 2\Big(\tfrac{1}{|G|}\sum_{z\in G}\delta_z\Big)^\top H g_G \;+\; \frac{1}{|G|}\sum_{z\in G}\delta_z^\top H\delta_z\\ &=\; g_G^\top H g_G \;+\; \frac{1}{|G|}\sum_{z\in G}\delta_z^\top H\delta_z \quad \left(\text{since } \frac{1}{|G|}\sum_{z\in G}\delta_z = 0\right). \end{aligned} \tag{18}$$

Applying the cyclic trace identity $a^\top H a = \mathrm{Tr}(H\,aa^\top)$ with $a=\delta_z$, recasts the above equation as a trace:

$$\begin{aligned}\frac{1}{|G|}\sum_{z\in G} g_z^\top H g_z \;-\; g_G^\top H g_G \;&=\; \frac{1}{|G|}\sum_{z\in G}\delta_z^\top H\delta_z\\ &=\; \frac{1}{|G|}\sum_{z\in G}\mathrm{Tr}\big(H\,\delta_z\delta_z^\top\big)\\ &=\; \mathrm{Tr}\Big(H\cdot\tfrac{1}{|G|}\sum_{z\in G}\delta_z\delta_z^\top\Big)\\ &=\; \mathrm{Tr}\big(H\,\Sigma_G\big), \end{aligned} \tag{19}$$

where $\Sigma_G \overset{\text{def}}{=} \frac{1}{|G|}\sum_{z\in G}\delta_z\delta_z^\top = \frac{1}{|G|}\sum_{z\in G}(g_z-g_G)(g_z-g_G)^\top$ is the within-group gradient covariance. Substituting this back into Equation (15),

$$\mathcal{I}_{\mathrm{proxy}}(G,\theta) - \mathcal{I}_{\mathrm{orcl}}(G,\theta) \;=\; \frac{\eta^2}{2}\,\mathrm{Tr}\big(H\,\Sigma_G\big) \;+\; O(\eta^3), \tag{20}$$

which is Equation (12). As the linear terms cancelled in Step 3, the leading discrepancy is the $O(\eta^2)$ curvature term, so the two quantities agree to first order.

□

The gap is therefore a second-order effect, governed by the product of the reference-loss curvature H and the within-group gradient covariance Σ_G: it vanishes when the group is perfectly coherent ($\Sigma_G = 0$) or the loss is locally flat ($H = 0$).

Lemma C.2 (Quadratic expansion of Φ)**.** *Writing $H \overset{def}{=} \nabla^2 L(\theta)$ for the Hessian of the reference loss at θ, the function Φ in Equation* (9) *admits the second-order expansion*

$$\Phi(v) \;=\; -\nabla L(\theta)^\top v \;+\; \tfrac{1}{2}\, v^\top H\, v \;+\; O(\|v\|^3). \tag{21}$$

Proof. Recall from Equation (9) that the loss-change function is

$$\Phi(v) \;=\; L(\theta - v) - L(\theta).$$

The only term that depends on v is the loss L evaluated at the displaced parameters $\theta - v$, so we expand L via its general second-order Taylor expansion about the base point θ. Assuming L is three-times continuously differentiable in a neighborhood of θ, for any point x near θ, the general second-order Taylor expansion is

$$L(x) \;=\; L(\theta) \;+\; \nabla L(\theta)^\top (x - \theta) \;+\; \tfrac{1}{2}\,(x-\theta)^\top \nabla^2 L(\theta)\,(x-\theta) \;+\; O(\|x-\theta\|^3). \tag{22}$$

Setting $x = \theta - v$ we have

$$\begin{aligned} L(\theta - v) \;&=\; L(\theta) \;+\; \nabla L(\theta)^\top (-v) \;+\; \tfrac{1}{2}\,(-v)^\top \nabla^2 L(\theta)\,(-v) \;+\; O(\|v\|^3) \\ &=\; L(\theta) \;-\; \nabla L(\theta)^\top v \;+\; \tfrac{1}{2}\, v^\top H\, v \;+\; O(\|v\|^3), \end{aligned}$$

where $H \overset{\text{def}}{=} \nabla^2 L(\theta)$ is the Hessian of the reference loss L evaluated at θ. Substituting this back into $\Phi(v) = L(\theta - v) - L(\theta)$ from Equation (9) gives

$$\Phi(v) \;=\; -\nabla L(\theta)^\top v \;+\; \tfrac{1}{2}\, v^\top H\, v \;+\; O(\|v\|^3), \tag{23}$$

which is Equation (21).

□

C.3 Proof of the exact bound

The leading-order statement of Theorem C.1 becomes the exact inequality of Theorem 5.1 under β-smoothness.

Proposition 5.1 (Individual-vs-Group Oracle gap, restated). *If the reference loss is β-smooth (i.e. $\|H\| \le \beta$, where H is its Hessian), then*

$$\left|\, \mathcal{I}_{\text{proxy}}(G, \theta) - \mathcal{I}_{\text{orcl}}(G, \theta) \,\right| \;\le\; \frac{\beta\, \eta^2}{2}\, \sigma_G^2,$$

where $\sigma_G^2 = \frac{1}{|G|} \sum_{z \in G} \|g_z - g_G\|^2$ is the within-group diversity of the per-example gradients.

Proof. First, write $m \stackrel{\text{def}}{=} \eta g_G$ for the mean of the points. Substituting the boxed identities Equations (10) and (11) for $\mathcal{I}_{\text{proxy}}(G,\theta)$ and $\mathcal{I}_{\text{orcl}}(G,\theta)$, the Jensen gap of Φ over those points is:

$$\begin{aligned}\mathcal{I}_{\text{proxy}}(G,\theta) - \mathcal{I}_{\text{orcl}}(G,\theta) &= \frac{1}{|G|}\sum_{z\in G}\Phi(\eta g_z) \;-\; \Phi(\eta g_G) \\ &= \frac{1}{|G|}\sum_{z\in G}\Phi(\eta g_z) \;-\; \Phi(m) \qquad (24)\\ &= \frac{1}{2|G|}\sum_{z\in G}(\eta g_z - m)^\top \nabla^2\Phi(\xi_z)\,(\eta g_z - m). \quad \text{(Lemma C.3)} \qquad (25)\end{aligned}$$

Here, Lemma C.3 applies directly to Equation (24), and $\xi_z = m + t_z(\eta g_z - m)$ with $t_z \in [0,1]$ are points on the segment between m and ηg_z.

Step 1. Bound each quadratic form. By Theorem C.2 the Hessian of Φ is $\nabla^2\Phi(\cdot) = H(\cdot)$, and β-smoothness of L means $\|H(\cdot)\| \le \beta$ at every point, including each ξ_z; thus $\|\nabla^2\Phi(\xi_z)\| \le \beta$. For any symmetric matrix A and vector u, the operator-norm inequality gives $|u^\top A u| \le \|A\|\,\|u\|^2$, so with $u = \eta g_z - m$,

$$\left|(\eta g_z - m)^\top \nabla^2\Phi(\xi_z)\,(\eta g_z - m)\right| \;\le\; \beta\,\|\eta g_z - m\|^2. \qquad (26)$$

Applying the triangle inequality to the average in Equation (25),

$$\begin{aligned}\left|\mathcal{I}_{\text{proxy}}(G,\theta) - \mathcal{I}_{\text{orcl}}(G,\theta)\right| &\le \frac{1}{2|G|}\sum_{z\in G}\beta\,\|\eta g_z - m\|^2 \\ &= \frac{\beta}{2}\cdot\frac{1}{|G|}\sum_{z\in G}\|\eta g_z - \eta g_G\|^2. \qquad (27)\end{aligned}$$

Step 2. Identify the variance. Factoring out η^2 and using the definition $\sigma_G^2 \stackrel{\text{def}}{=} \frac{1}{|G|}\sum_{z\in G}\|g_z - g_G\|^2$ of the within-group gradient diversity,

$$\begin{aligned}\frac{1}{|G|}\sum_{z\in G}\|\eta g_z - \eta g_G\|^2 &= \eta^2\cdot\frac{1}{|G|}\sum_{z\in G}\|g_z - g_G\|^2 \\ &= \eta^2\,\sigma_G^2. \qquad (28)\end{aligned}$$

Combining with Step 1,

$$\left|\mathcal{I}_{\text{proxy}}(G,\theta) - \mathcal{I}_{\text{orcl}}(G,\theta)\right| \;\le\; \frac{\beta\,\eta^2}{2}\,\sigma_G^2, \qquad (29)$$

which is Equation (3). The bound is exact in η; the agreement to first order follows separately from Theorem C.1.

□

Lemma C.3 (Exact second-order Jensen gap)**.** *Let $\Phi : \mathbb{R}^d \to \mathbb{R}$ be the loss-change function Equation* (9)*, and assume it is twice continuously differentiable. For a group G with per-example gradients $\{g_z\}_{z\in G}$, mean gradient $g_G = \frac{1}{|G|}\sum_{z\in G} g_z$, and step $m \stackrel{\text{def}}{=} \eta g_G$, there exist points $\xi_z = m + t_z(\eta g_z - m)$ with $t_z \in [0,1]$, each on the segment between m and ηg_z, such that*

$$\frac{1}{|G|}\sum_{z\in G}\Phi(\eta g_z) \;-\; \Phi(m) \;=\; \frac{1}{2|G|}\sum_{z\in G}(\eta g_z - m)^\top \nabla^2\Phi(\xi_z)\,(\eta g_z - m). \qquad (30)$$

Proof. We recall Taylor's theorem with the Lagrange form of the remainder. For a twice continuously differentiable $f : \mathbb{R}^d \to \mathbb{R}$ and any two points $a, b \in \mathbb{R}^d$, there exists a point ξ on the line segment joining a and b, that is, $\xi = a + t(b - a)$ for some $t \in [0, 1]$, such that

$$f(b) \;=\; f(a) \;+\; \nabla f(a)^\top (b - a) \;+\; \tfrac{1}{2}\,(b - a)^\top \nabla^2 f(\xi)\,(b - a). \tag{31}$$

We apply Equation (31) to $f = \Phi$, with $a = m$ and $b = \eta g_z$ for each $z \in G$. This produces, for each z, a point $\xi_z = m + t_z(\eta g_z - m)$, $t_z \in [0, 1]$, on the segment between m and ηg_z such that

$$\Phi(\eta g_z) \;=\; \Phi(m) \;+\; \nabla\Phi(m)^\top (\eta g_z - m) \;+\; \tfrac{1}{2}\,(\eta g_z - m)^\top \nabla^2\Phi(\xi_z)\,(\eta g_z - m). \tag{32}$$

No higher-order terms are dropped: this is an equality, with the second-order term evaluated at ξ_z. Averaging over $z \in G$, the linear term vanishes because, with $m = \eta g_G$, the deviations sum to zero:

$$\begin{aligned}
\frac{1}{|G|}\sum_{z\in G} \nabla\Phi(m)^\top (\eta g_z - m) \;&=\; \nabla\Phi(m)^\top \Big(\tfrac{1}{|G|}\sum_{z\in G} \eta g_z - m\Big) \\
&=\; \nabla\Phi(m)^\top (\eta g_G - m) \\
&=\; 0.
\end{aligned}$$

Hence $\frac{1}{|G|}\sum_{z\in G}\Phi(\eta g_z) = \Phi(m) + \frac{1}{2|G|}\sum_{z\in G}(\eta g_z - m)^\top \nabla^2\Phi(\xi_z)(\eta g_z - m)$, and subtracting $\Phi(m)$ gives Equation (30).

□

D Additional Figures and Tables

This section contains supporting plots and figures.

D.1 Synthetic Data Curation Full Results

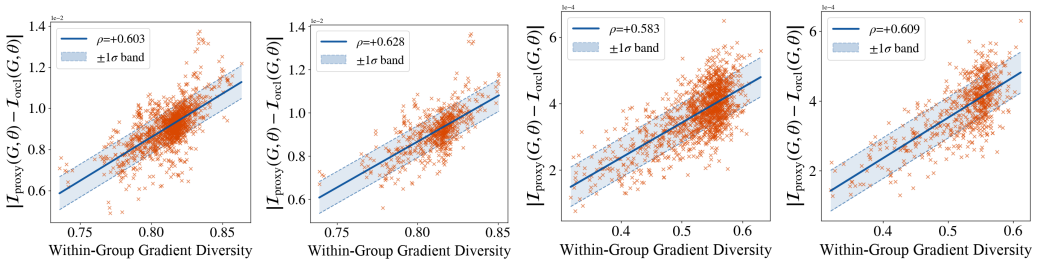
This section shows the full results from the experiments in Section 4.

Table 9: Per-task accuracy at the final step (200 steps) for the Repro-400M continuation, by selection method and rephrase factor r. Best average within each r is in **bold**, and second best is underlined.

Method	HellaSwag	ARC-C	PIQA	MMLU	CSQA	LAMBADA	SQuAD	CoQA	Avg
Rephrase factor $r = 1$									
Random	30.95	26.96	66.16	23.19	19.41	41.04	9.82	25.52	30.38
GradSim	30.81	26.28	65.34	22.98	21.21	36.37	3.65	24.69	28.92
FineWeb-Edu	30.89	25.51	64.53	22.87	19.16	38.09	17.79	23.60	30.31
BM25	31.14	26.45	65.34	23.21	19.66	38.15	5.40	26.68	29.50
GMRel ($\tau \approx 1.01$)	30.41	25.94	65.02	23.15	19.82	36.76	25.17	24.62	<u>31.36</u>
GREATS	30.76	26.79	65.13	23.85	20.48	38.87	24.20	24.87	**31.87**
Rephrase factor $r = 10$									
Random	31.09	26.45	65.13	24.10	20.88	35.49	6.23	27.24	29.58
GradSim	30.77	27.39	64.15	24.48	19.57	30.76	5.86	21.34	28.04
FineWeb-Edu	30.75	26.54	63.87	22.97	20.15	34.29	7.24	23.47	28.66
BM25	30.41	26.30	63.35	22.81	19.88	33.95	7.13	23.37	28.40
GMRel ($\tau \approx 1.01$)	30.77	26.45	64.20	24.44	19.82	32.66	20.61	24.21	**30.40**
GREATS	30.52	26.09	65.89	24.17	19.74	35.55	16.49	23.69	<u>30.27</u>
Rephrase factor $r = 50$									
Random	31.77	27.13	64.09	25.66	21.62	28.00	6.13	20.42	**28.10**
GradSim	31.12	26.19	62.24	24.28	23.42	25.33	7.24	16.09	26.99
FineWeb-Edu	30.22	25.85	62.84	26.03	20.97	25.58	6.47	16.36	26.79
BM25	29.97	26.02	63.06	24.38	20.39	29.67	11.66	19.03	<u>28.02</u>
GMRel ($\tau \approx 1.01$)	31.01	26.88	63.44	24.24	20.23	28.53	3.92	23.09	27.67
GREATS	30.50	26.28	62.95	25.13	21.13	29.54	6.50	21.20	27.90
Baseline	30.82	25.85	64.36	23.94	19.82	37.30	7.47	27.61	29.65

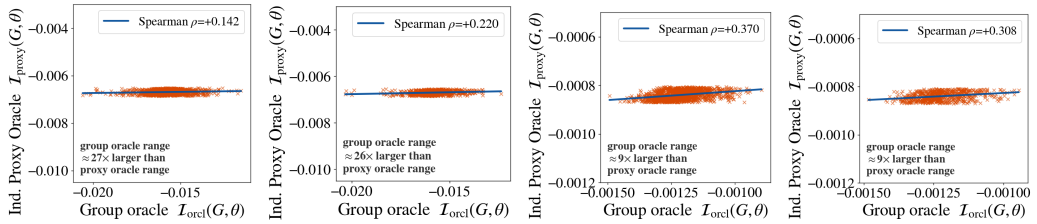
D.2 Group versus Individual Proxy Oracle Gap Across Group Sizes

This section contains the full sweep of the discrepancy between the group and individual proxy oracle scores across group sizes, summarized in Figure 6 of the main body.



(a) Pre-train (size 5120). (b) Pre-train (size 10240). (c) Post-train (size 128). (d) Post-train (size 512).

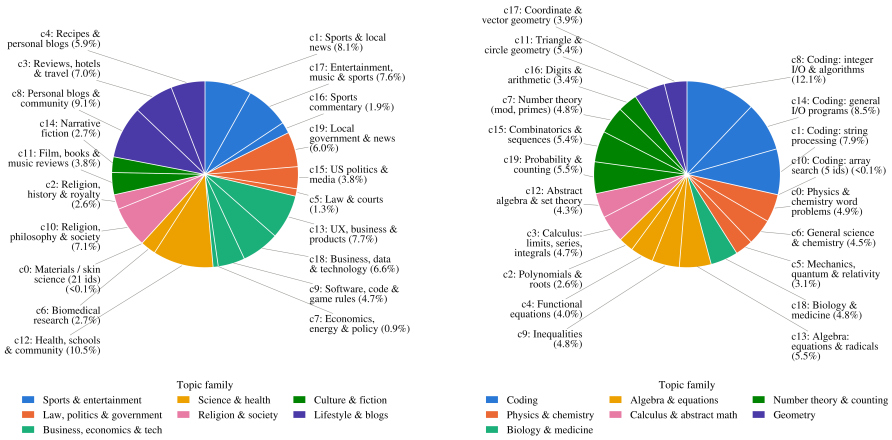
Figure 8: Discrepancy between the group oracle $\mathcal{I}_{\text{orcl}}(G,\theta)$ and individual proxy oracle $\mathcal{I}_{\text{proxy}}(G_j,\theta)$ scores across varying sizes of both the synthetic pretraining and the RLVR data groups.



(a) Pre-train (size 5120). (b) Pre-train (size 10240). (c) Post-train (size 128). (d) Post-train (size 512).

Figure 9: Range of group oracle and individual proxy oracle scores across our constructed groups, for each group size in the pre-training and post-training settings.

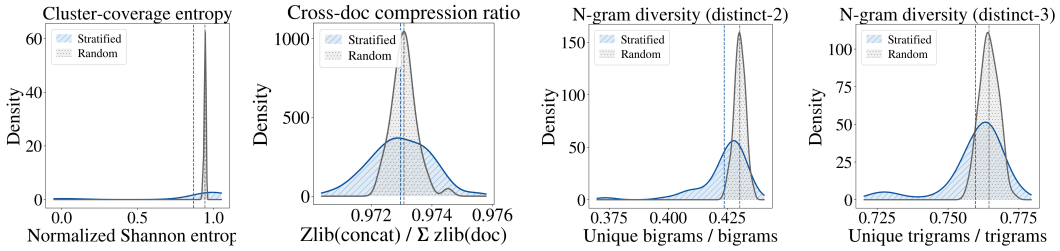
D.3 Cluster Topic Composition



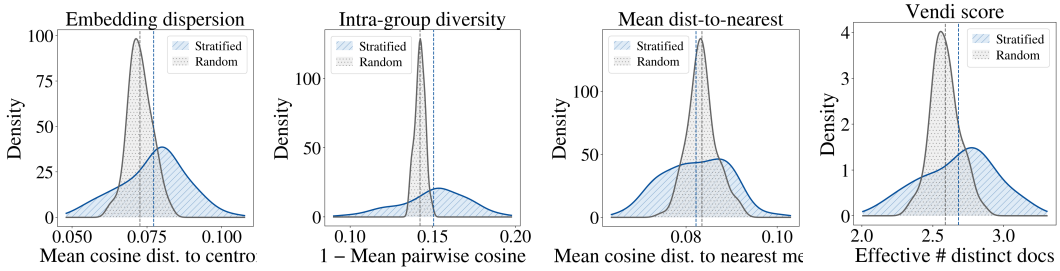
(a) Pre-training (Repro). (b) RLVR (GooseReason).

Figure 10: Topic composition of the $k = 20$ semantic clusters used for proxy-matched diversity sampling. Each slice is one cluster, labeled with its topic and share of the corpus, and colored by topic family.

D.4 Additional Diversity Plots

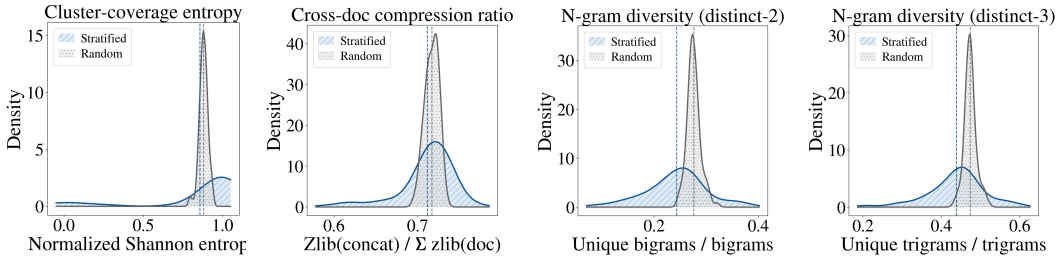


(a) Cluster entropy. (b) Compression ratio. (c) Distinct-2. (d) Distinct-3.

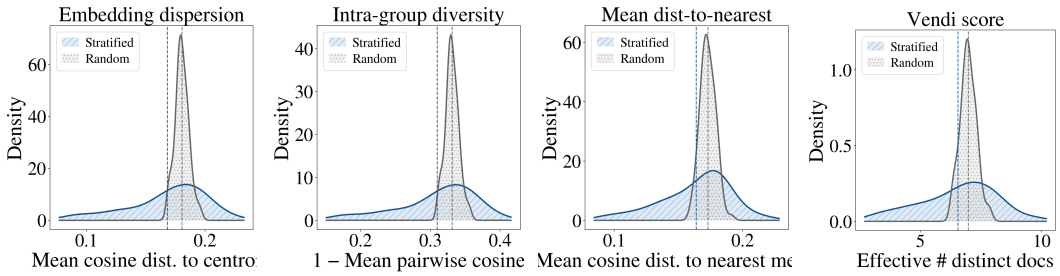


(e) Embedding dispersion. (f) Intra-group diversity. (g) Mean distance to nearest. (h) Vendi score.

Figure 11: Diversity metrics on our constructed groups versus randomly formed groups in the pre-training setting.



(a) Cluster entropy. (b) Compression ratio. (c) Distinct-2. (d) Distinct-3.



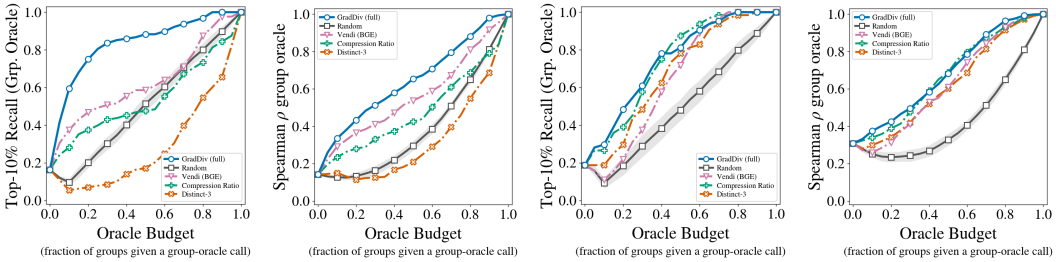
(e) Embedding dispersion. (f) Intra-group diversity. (g) Mean distance to nearest. (h) Vendi score.

Figure 12: Diversity metrics on our constructed groups versus randomly formed data groups in the post-training setting.

D.5 Group Oracle Budget Analysis Plots

This section contains the full set of group-oracle budget allocation plots, summarized in Figure 7 of the main body.

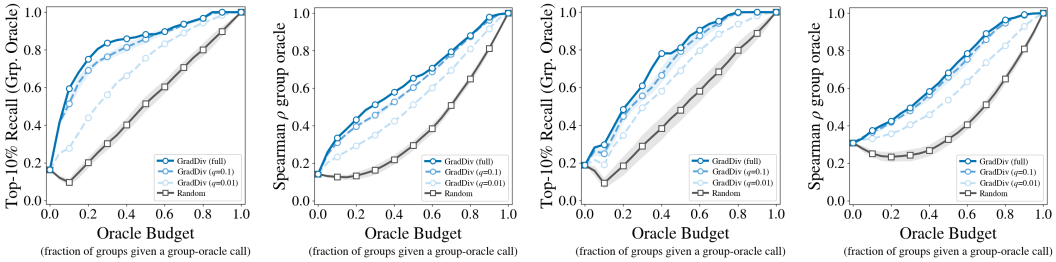
Choice of diagnostic. Figure 13 compares GradDiv against cheap diversity heuristics as the diagnostic used to decide which groups receive a group-oracle call.



(a) Pre-train, top-10% recall. (b) Pre-train, Spearman ρ. (c) Post-train, top-10% recall. (d) Post-train, Spearman ρ.

Figure 13: Choice of diagnostic for group-oracle budget allocation, for pre-train and post-train data (group sizes of 5120 and 512 samples, respectively).

Gradient subsampling. Figure 14 shows the effect of subsampling gradients when computing the GradDiv diagnostic.



(a) Pre-train, top-10% recall. (b) Pre-train (5120), Spearman ρ. (c) Post-train, top-10% recall. (d) Post-train, Spearman ρ.

Figure 14: Effect of gradient subsampling on group-oracle budget allocation, comparing the full GradDiv diagnostic against subsampled variants ($q = 0.1$, $q = 0.01$) for pre-train and post-train data (group sizes of 5120 and 512 samples, respectively).