Influence Functions for Efficient Data Selection in Reasoning

Prateek Humane[1,2,†] **Paolo Cudrano**[3] **Daniel Z. Kaplan**[4]
Matteo Matteucci[3] **Supriyo Chakraborty**[5] **Irina Rish**[1,2]

[1]Mila, Québec AI Institute, [2]Université de Montréal, [3]Politecnico di Milano,
[4]Sage Group, [5]Capital One
[†]prateek.humane@mila.quebec

Abstract

Fine-tuning large language models (LLMs) on chain-of-thought (CoT) data shows that a small amount of high-quality data can outperform massive datasets. Yet, what constitutes "quality" remains ill-defined. Existing reasoning methods rely on indirect heuristics such as problem difficulty or trace length, while instruction-tuning has explored a broader range of automated selection strategies—but rarely in the context of reasoning. We propose to define reasoning data quality using influence functions, which measure the causal effect of individual CoT examples on downstream accuracy, and introduce influence-based pruning, which consistently outperforms perplexity and embedding-based baselines on math reasoning within a model family.

1 Introduction

Large language models (LLMs) show strong reasoning gains when fine-tuned on chain-of-thought (CoT) data (DeepSeek-AI and Others, 2025). Recent studies suggest that data quality often outweighs quantity (Rodchenko et al., 2025; Yuan et al., 2025; Ye et al., 2025), yet what constitutes "quality" remains unclear.

Prior work on reasoning has used proxies for data quality, such as question difficulty (Chen et al., 2025) or CoT length (Yuan et al., 2025), which offer only indirect signals of downstream performance. In contrast, instruction-tuning studies have systematically evaluated automated data selection methods—perplexity filtering, embedding similarity, gradient-based scoring (Yin and Rush, 2025; Ivison et al., 2025)—but these have not been fully applied to reasoning data.

What ultimately matters is not whether an example appears challenging or diverse, but whether fine-tuning on it *causally improves reasoning performance*. Influence functions (IFs) provide a principled way to capture such causal effects by estimating the contribution of individual training examples to downstream accuracy. While IFs have been explored in instruction tuning and knowledge tracing, they remain unused for reasoning-specific pruning.

In this work, we extend IFs to define reasoning data quality through causal impact on downstream accuracy. We introduce IF-based pruning, which surpasses common baselines on math reasoning within a model family. However, transfer across families remains inconclusive, raising the open question of whether data quality is intrinsic or model-specific.

2 Related Works

Let $\mathcal{D}$ denote a training dataset and $\mathcal{V}$ a validation set sampled from the evaluation distribution. Data selection methods assign scores to examples in $\mathcal{D}$ in order to prune/reweight the training pool. These

39th Conference on Neural Information Processing Systems (NeurIPS 2025) Workshop: Efficient Reasoning.

methods either (1) score examples directly, $s : \mathcal{D} \to \mathbb{R}$, or (2) compute pairwise scores with respect to $\mathcal{V}$, $s : \mathcal{V} \times \mathcal{D} \to \mathbb{R}$, then aggregate across $v \in \mathcal{V}$ to obtain $s(d)$ for each $d \in \mathcal{D}$. We group existing approaches accordingly.

Reasoning heuristics Heuristic methods define scores $s(d)$ directly from inherent properties of each training example $d \in \mathcal{D}$, independent of $\mathcal{V}$ or a specific model. Recent works take into account heuristics such as estimated question difficulty, CoT length, and problem diversity (Ye et al., 2025; Muennighoff et al., 2025). LIMO (Ye et al., 2025) also incorporates human evaluation, selecting solutions that exhibit desirable behaviors such as elaboration and self-verification. Select2Reason (Yang et al., 2025) provides a systematic study, showing that higher $s(d)$ assigned to examples with longer traces and harder problems yields stronger downstream performance, whereas diversity has limited benefit.

Perplexity based These methods score each example $s(d)$ by its negative log-likelihood under the base model. *Top-PPL* selects high-perplexity (hard) cases, while *Mid-PPL* favors mid-range examples to avoid trivial or outlier data. These scores are cheap to compute and have shown to improve sample efficiency in the instruction tuning setting (Yin and Rush, 2025; Ankner et al., 2024).

Embedding based Embedding-based methods use data representations to assign scores. S1 (Muennighoff et al., 2025) enforces coverage by classifying problems into topics with an LLM and sampling across them. RDS+ (Ivison et al., 2025) computes pairwise scores $s(v, d) = \langle E(v), E(d) \rangle$ between validation and training examples, selecting examples most relavent to each $v \in \mathcal{V}$ in a round-robin fashion.

Gradient based Gradient and influence based methods compute $s(v, d)$ by estimating the causal contribution of training examples on validation loss. TracIn (Pruthi et al., 2020) sums gradient inner products across training checkpoints and LESS (Xia et al., 2024) scales this approach to large models using gradient datastores. InfDist (Nikdan et al., 2025) learns optimal per-example weights so that a single gradient step minimizes validation loss.

Our influence function approach Unlike gradient-similarity methods that track training dynamics, influence functions measure the effect of a training example on converged model parameters and downstream objectives $f(\theta^\star)$. Formally, the influence of a training example $d \in \mathcal{D}$ is $\mathcal{I}_f(d) = -\nabla_\theta f(\theta^\star)^\top H^{-1} \nabla_\theta L(d, \theta^\star)$ where H is the Hessian of the training objective at $\theta^\star$. Approximations such as EK-FAC have made this tractable for LLMs (Grosse et al., 2023). Ruis et al. (2025) shows that examples with high estimated influence on cross-entropy loss also causally affect downstream accuracy, and their removal causes larger performance drops than random pruning or TracIn.

We extend this idea to reasoning, scoring examples by their influence on whether fine-tuning improves or harms correctness. This approach yields scores $s(d)$ that capture how each training example $d \in \mathcal{D}$ steers the model toward or away from correct reasoning behavior.

3 Methods

3.1 Influence Functions Scoring

Let θ be a pretrained model and $\mathcal{D}$ the LIMO training dataset (Ye et al., 2025). We fine-tune θ on $\mathcal{D}$ for 10 epochs following the settings of Ye et al. (2025), obtaining θ^{LIMO}. We then evaluate on the MATH500 benchmark $\mathcal{V}$ for both model parameters θ and θ^{LIMO} (Hendrycks et al., 2021).

To analyze how fine-tuning changes behavior, we compare correctness before and after fine-tuning for each query $(q, a) \in \mathcal{V}$. This yields two disjoint subsets $C, I \subseteq \mathcal{V}$:

$$C = \{(q, a) \in \mathcal{V} : \text{Acc}(f(q; \theta), a) = 0 \wedge \text{Acc}(f(q; \theta^{\text{LIMO}}), a) = 1\}$$
$$I = \{(q, a) \in \mathcal{V} : \text{Acc}(f(q; \theta), a) = 1 \wedge \text{Acc}(f(q; \theta^{\text{LIMO}}), a) = 0\}$$

Here $\text{Acc}(f(q; \theta), a)$ is a binary indicator of whether the model $f(\cdot; \theta)$ greedy sampling completion for query q matches the ground-truth answer a. Thus, C collects queries where fine-tuning improves correctness, while I collects queries where fine-tuning degrades correctness.

Our goal is to estimate the causal influence of training examples $d \in \mathcal{D}$ on these outcome sets. Because influence functions require a differentiable proxy, we follow Grosse et al. (2023) and

compute influence with respect to cross-entropy loss on completions. For each validation query $v \in C \cup I$ with completion y_c, the influence of a training point d is:

$$\mathcal{I}(d;(q,a)) = -\nabla_\theta \mathcal{L}_{\text{CE}}(f(q;\theta^{\text{LIMO}}),\theta)^\top H^{-1} \nabla_\theta \mathcal{L}_{\text{CE}}(d,\theta^{\text{LIMO}})$$

where $\mathcal{L}_{\text{CE}}(f(q;\theta^{\text{LIMO}}),\theta)$ is the cross-entropy loss on the generated completion and $\mathcal{L}_{\text{CE}}(d,\theta^{\text{LIMO}})$ is the training loss of d at convergence.

By aggregating $\mathcal{I}(d;v)$ across queries $v \in C$ and $v \in I$, we obtain scores $s(d)$ that quantify whether training on d tends to push the model toward correct completions (beneficial influence) or toward incorrect completions (harmful influence). These scores provide a principled, influence-based ranking of reasoning examples in $\mathcal{D}$.

3.2 Data Pruning

We first aggregate influence scores across validation queries v:

$$s_C(d) = \frac{1}{|C|}\sum_{v\in C}\mathcal{I}(d;v), \quad s_I(d) = \frac{1}{|I|}\sum_{v\in I}\mathcal{I}(d;v).$$

To prevent bias from queries with disproportionately large scores, we complement raw influence scores $s(d)$ with rank-based measures $r(d)$. For each query v, we sort training examples by their influence values in descending order and assign ranks. The final rank for a datapoint is its average position across queries:

$$r_C(d) = \frac{1}{|C|}\sum_{v\in C}\text{rank}(\mathcal{I}(d;v)), \quad r_I(d) = \frac{1}{|I|}\sum_{v\in I}\text{rank}(\mathcal{I}(d;v)).$$

Pruning is then performed by **intersecting thresholds on both scores and ranks**, ensuring that selected datapoints are consistently high or low influence across queries, rather than dominated by outliers.

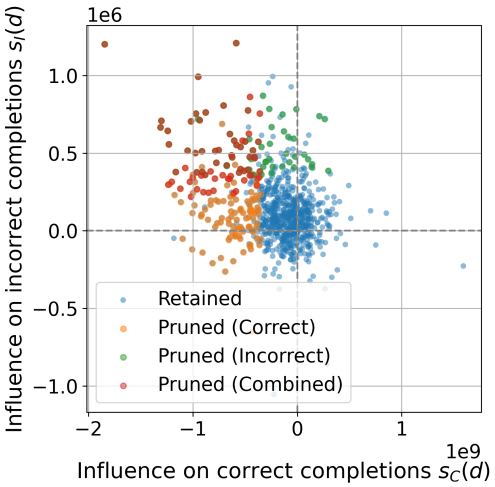


Figure 1: Influence scores $s_C(d), s_I(d)$ for training examples $d \in \mathcal{D}$. Points farther left contribute less to correct answers, while points higher up push the model toward wrong answers. Interestingly, the majority of outliers appear in the top-left quadrant, where lie the datapoints we predict to be most harmful. *Note: as the final pruning is affected also by rank scores $r(d)$, few unpruned points may also be found in such quadrant.*

We evaluate three pruning strategies (See Figure 1). **(i) *Correct***: prune the bottom 20% of examples by intersecting low $s_C(d)$ and $r_C(d)$, removing examples that contribute least to correct completions; **(ii) *Incorrect***: prune the top 10% of examples by intersecting high $s_I(d)$ and $r_I(d)$, removing examples that most strongly push the model toward incorrect completions; **(iii) *Combined***: prune 10% of examples that satisfy both of the above criteria.

Interestingly, histograms of the score distributions in Figure 2 show long tails: in the correct case, many examples have negligible or negative benefit, while in the incorrect case, outliers with high influence dominate harmful behavior.

4 Experiments and Results

We perform supervised fine-tuning (SFT) on pruned versions of the LIMO dataset, using two models: LLaMA-3-8B-Instruct (Grattafiori et al., 2024), also used during pruning, and Qwen2.5-Math-7B-Instruct (Yang et al., 2024), external to the pruning process. We ran full finetuning following the training settings from LIMO for 10 epochs, saving checkpoints at every other epoch. Given that we already were optimizing for MATH500, we treated it as a validation set and chose the checkpoint which performed best on MATH500. We chose this strategy as for different datasets, optimal checkpoints were not consistent.

Evaluation focuses on the math reasoning benchmarks: AIME24, AMC23, OlympiadBench (He et al., 2024), GSM8k (Cobbe et al., 2021). Additionally we include GPQA (Rein et al., 2024) to assess transfer to non-math reasoning. We compare three IF-based pruning strategies (*Correct*, *Incorrect*,

Table 1: **Results on LLaMA-3-8B-Instruct.** Pass@1 with N=8 samples, zero-shot CoT. We fine-tune on math-pruned subsets selected by influence functions (*Correct*, *Incorrect*, *Combined*) and compare against baselines (*Random*, *Mid-PPL*, *RDS+*), as well as the unpruned dataset and base model. Evaluation covers math reasoning benchmarks (AIME24, AMC23, OlympiadBench, GSM8k) and GPQA for non-math transfer. Best in **bold**; best-after-SFT underlined. Amount of samples not pruned in [brackets].

LLAMA3-8B-INSTRUCT

PRUNING	GSM8K	OLYMPIADBENCH	AMC23	AIME24	GPQA
Combined [90%] (ours)	**0.8113**	**0.0880**	**0.1156**	0.0000	0.1654
Incorrect [90%] (ours)	0.7878	0.0798	0.1094	0.0042	0.1439
Correct [80%] (ours)	0.7922	0.0691	0.0719	**0.0083**	0.1496
RDS+ [90%]	0.8015	**0.0880**	0.0813	0.0042	0.1553
Mid-PPL [90%]	0.8052	0.0796	0.0781	0.0000	0.1244
Random [90%]	0.7839	0.0865	0.1094	0.0042	0.1509
None	0.7722	0.0685	0.0813	0.0000	0.1199
W/o SFT	0.5958	0.0498	0.0781	0.0000	**0.2973**

Combined) against *Random*, *Mid-PPL* , and *RDS+* (Ivison et al., 2025), and report also the unpruned dataset and the base model for reference. All results use the unbiased pass@1 metric (Chen et al., 2021) with N=8 samples in a zero-shot chain-of-thought setting, following the LIMO protocol. Full training and evaluation details are provided in the Appendix.

Within-model IF pruning is effective. On Llama3-8B-Instruct (Table 1), our IF-selected subsets match or outperform *Random*/*Mid-PPL* baselines and *RDS+*. Results on AIME24 remain neglibigle across all methods, reflecting the benchmark's difficulty and indicating that SFT on small pruned subsets cannot recover performance. On GPQA, math SFT tends to degrade general scientific reasoning; our selections mitigate but do not eliminate this effect. Even when pruning more aggressively 50% of the data, IF-based selection improves upon or at least remains competitive with baselines (see Appendix, Table 3).

Cross-family transfer is non-conclusive. When fine-tuning Qwen2.5-Math-7B-Instruct with data subsets selected using Llama3-8B-Instruct (Appendix, Table 2), winners vary by task and no pruning strategy consistently outperforms the baselines. This suggests that improvements from subsets selected with one model do not straightforwardly carry over to a different model family.

5 Limitations and Future Work

While our study demonstrates the promise of influence-function–based pruning for reasoning, several limitations and opportunities for future work remain.

Experimental variance. Our results are based on a single training run per setting. Future work should include multiple seeds and confidence intervals to account for evaluation variance.

Dataset scope. We focus on LIMO, which is already pruned and curated for questions Qwen2.5-Math-7B-Instruct cannot solve. This may underestimate the potential of influence functions on larger, less filtered datasets such as Open-R1.

Comprehensive evaluation. We evaluate only a limited set of baselines and a single model size. We are planning a comprehensive study comparing influence-based pruning against reasoning heuristics and other methods such as InfDist, while spanning different datasets and model scales, to better understand when each approach is effective.

Computational cost. Influence-function estimation is expensive, and we do not account for the additional compute required for data selection. Future work should explore more efficient approximations, such as gradient datastores or training LLMs to predict influence scores directly.

Evolving data quality. Finally, we treat data quality as static, but in practice it may evolve as models improve. An open question is how influence scores shift with scale and training stage—for example, whether higher-quality examples track increasingly difficult problems as models become stronger.

References

Ankner, Z., Blakeney, C., Sreenivasan, K., Marion, M., Leavitt, M. L., and Paul, M. (2024). Perplexed by perplexity: Perplexity-based data pruning with small reference models.

Chen, M., Tworek, J., Jun, H., Yuan, Q., Pinto, H. P. D. O., Kaplan, J., Edwards, H., Burda, Y., Joseph, N., Brockman, G., et al. (2021). Evaluating large language models trained on code. *arXiv preprint arXiv:2107.03374*.

Chen, Z., Liu, T., Tian, M., Tong, Q., Luo, W., and Liu, Z. (2025). Advancing mathematical reasoning in language models: The impact of problem-solving data, data synthesis methods, and training stages.

Cobbe, K., Kosaraju, V., Bavarian, M., Chen, M., Jun, H., Kaiser, L., Plappert, M., Tworek, J., Hilton, J., Nakano, R., et al. (2021). Training verifiers to solve math word problems. *arXiv preprint arXiv:2110.14168*.

DeepSeek-AI and Others (2025). Deepseek-r1: Incentivizing reasoning capability in llms via reinforcement learning.

Grattafiori, A., Dubey, A., Jauhri, A., Pandey, A., Kadian, A., Al-Dahle, A., Letman, A., Mathur, A., Schelten, A., Vaughan, A., et al. (2024). The llama 3 herd of models. *arXiv preprint arXiv:2407.21783*.

Grosse, R., Bae, J., Anil, C., Elhage, N., Tamkin, A., Tajdini, A., Steiner, B., Li, D., Durmus, E., Perez, E., Hubinger, E., Lukošiūtė, K., Nguyen, K., Joseph, N., McCandlish, S., Kaplan, J., and Bowman, S. R. (2023). Studying large language model generalization with influence functions.

He, C., Luo, R., Bai, Y., Hu, S., Thai, Z. L., Shen, J., Hu, J., Han, X., Huang, Y., Zhang, Y., Liu, J., Qi, L., Liu, Z., and Sun, M. (2024). Olympiadbench: A challenging benchmark for promoting agi with olympiad-level bilingual multimodal scientific problems.

Hendrycks, D., Burns, C., Kadavath, S., Arora, A., Basart, S., Tang, E., Song, D., and Steinhardt, J. (2021). Measuring mathematical problem solving with the MATH dataset. In *Thirty-fifth Conference on Neural Information Processing Systems Datasets and Benchmarks Track (Round 2)*.

Ivison, H., Zhang, M., Brahman, F., Koh, P. W., and Dasigi, P. (2025). Large-scale data selection for instruction tuning.

Muennighoff, N., Yang, Z., Shi, W., Li, X. L., Fei-Fei, L., Hajishirzi, H., Zettlemoyer, L., Liang, P., Candès, E., and Hashimoto, T. (2025). s1: Simple test-time scaling.

Nikdan, M., Cohen-Addad, V., Alistarh, D., and Mirrokni, V. (2025). Efficient data selection at scale via influence distillation.

Pruthi, G., Liu, F., Sundararajan, M., and Kale, S. (2020). Estimating training data influence by tracing gradient descent.

Rein, D., Hou, B. L., Stickland, A. C., Petty, J., Pang, R. Y., Dirani, J., Michael, J., and Bowman, S. R. (2024). GPQA: A graduate-level google-proof q&a benchmark. In *First Conference on Language Modeling*.

Rodchenko, T., Noy, N., Scherrer, N., and Prendki, J. (2025). Not every ai problem is a data problem: We should be intentional about data scaling.

Ruis, L., Mozes, M., Bae, J., Kamalakara, S. R., Talupuru, D., Locatelli, A., Kirk, R., Rocktäschel, T., Grefenstette, E., and Bartolo, M. (2025). Procedural knowledge in pretraining drives reasoning in large language models.

Xia, M., Malladi, S., Gururangan, S., Arora, S., and Chen, D. (2024). Less: Selecting influential data for targeted instruction tuning.

Yang, A., Zhang, B., Hui, B., Gao, B., Yu, B., Li, C., Liu, D., Tu, J., Zhou, J., Lin, J., et al. (2024). Qwen2. 5-math technical report: Toward mathematical expert model via self-improvement. *arXiv preprint arXiv:2409.12122*.

Yang, C., Lin, X., Xu, C., Jiang, X., Wu, X., Liu, H., Xiong, H., and Guo, J. (2025). Select2reason: Efficient instruction-tuning data selection for long-cot reasoning.

Ye, Y., Huang, Z., Xiao, Y., Chern, E., Xia, S., and Liu, P. (2025). Limo: Less is more for reasoning.

Yin, J. O. and Rush, A. M. (2025). Compute-constrained data selection.

Yuan, W., Yu, J., Jiang, S., Padthe, K., Li, Y., Wang, D., Kulikov, I., Cho, K., Tian, Y., Weston, J. E., and Li, X. (2025). Naturalreasoning: Reasoning in the wild with 2.8m challenging questions.

A Training and Evaluation Details

Training. Our SFT adopts the same training setup as LIMO (Ye et al., 2025). All experiments are run on 8×H100 GPUs, with full-parameter fine-tuning for 10 epochs. We use a cosine learning rate schedule with peak learning rate 5×10^{-6}.

Evaluation. We follow the evaluation setup of LIMO (Ye et al., 2025), with the addition of the GSM8k test set manually added. We report the unbiased pass@1 metric (Chen et al., 2021), estimated from N=8 sampled generations. Sampling uses temperature 0.6, top-$p = 0.95$, top-$k = 1$, and a maximum generation length of $16{,}384$ tokens.

B Notes on influence functions computation

To compute influence function scores, we follow the derivation and approximations described in Grosse et al. (2023). Following Ruis et al. (2025),we perform the computation considering only MLP layers.

C Further details on our pruning strategies

In Figure 2, we report full details on the distribution of influence scores $s_C(d), s_I(d)$ and rank scores $r_C(d), r_I(d)$ over the over all data points $d \in \mathcal{D}$. We notice how the distributions display long tails where we expect pruning to be more effective.

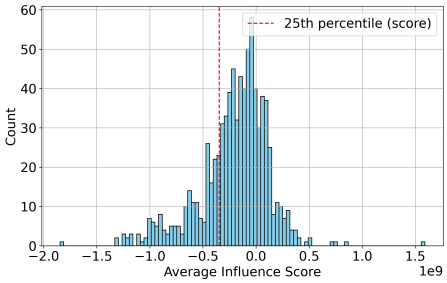
D Results on Qwen2.5-Math-7B-Instruct

We evaluate whether the dataset pruned using a specific model (LLaMA-3-8B-Instruct) would transfer its benefits when applied to a different model family. We report the full results on Qwen2.5-Math-7B-Instruct in Table 2.

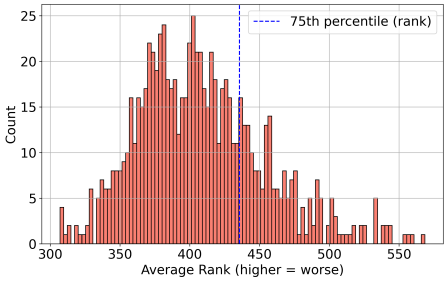
Table 2: **Results on Qwen2.5-Math-7B-Instruct.** Pass@1 with N=8 samples, zero-shot CoT. Same setup as Table 1, but we fine-tune Qwen2.5-Math-7B-Instruct on pruned subsets selected using LLaMA-3-8B-Instruct, testing their transfer across model families. Best in **bold**. Amount of samples kept in [brackets].

QWEN2.5-MATH-7B-INSTRUCT					
PRUNING	GSM8K	OLYMPIADBENCH	AMC23	AIME24	GPQA
Combined [90%] (ours)	0.9567	0.4635	0.5781	0.1917	0.3220
Incorrect [90%] (ours)	**0.9584**	0.4643	0.5719	0.1667	0.3232
Correct [80%] (ours)	0.9566	0.4637	0.6000	0.1458	0.3245
RDS+ [90%]	0.9566	0.4669	0.6094	0.1583	0.3201
Mid-PPL [90%]	0.9571	0.4619	**0.6125**	0.1583	0.3169
Random [90%]	0.9575	0.4596	0.5781	**0.2042**	0.3295
None	0.9576	**0.4687**	0.5469	0.1542	**0.3340**
W/o SFT	0.9538	0.3876	0.5813	0.1125	0.2929

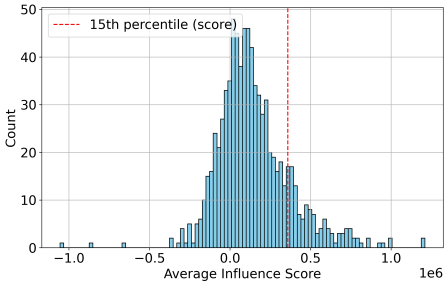
(a) *Correct* influence score $s_C(d)$.



(c) *Correct* rank score $r_C(d)$.



(b) *Inorrect* influence score $s_I(d)$.



(d) *Incorrect* rank score $r_I(d)$.

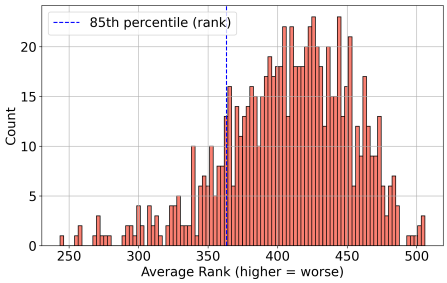


Figure 2: Histograms of the influence and rank scores measured across the training set $\mathcal{D}$.

E More aggressive pruning (50% of data)

E.1 Pruning strategy

We experiment with a more aggressive IF-based pruning strategy, removing 50% of the training data. Pruning is performed based on the following function:

$$A(s(v,d)) = \frac{\text{sign}(s_C(d))s_C(d)^2}{\sigma_C} - \frac{\text{sign}(s_I(d))s_I(d)^2}{\sigma_I},$$

taking the top 50% of the resulting ranking, as illustrated also in Figure 3.

E.2 Results

The results for our *Combined* strategy and all baselines confirm our main findings even under more aggressive pruning. As shown in Table 3, IF-based pruning remains competitive with strong baselines: *Combined (50%)* is best on GSM8k, AMC23, and MATH500, while *Random (50%)* is slightly ahead on OlympiadBench. On GPQA, performance improves compared to lighter pruning, confirming that reducing the amount of math-focused fine-tuning lessens the degradation of general scientific reasoning.

F Further plots

Figure 4 visualizes how our pruning strategy outperforms other baselines for Llama3-8B-Instruct, while results are inconclusive on cross-family transfer with Qwen2.5-Math-7B-Instruct.

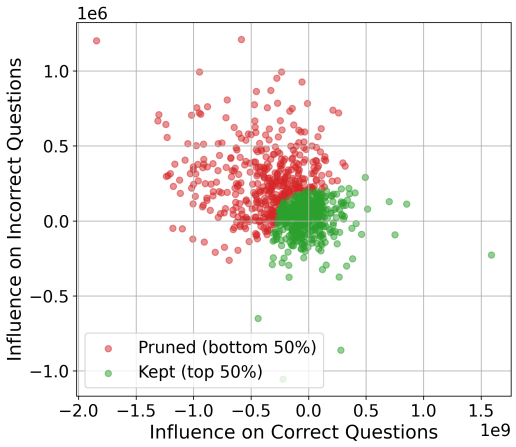
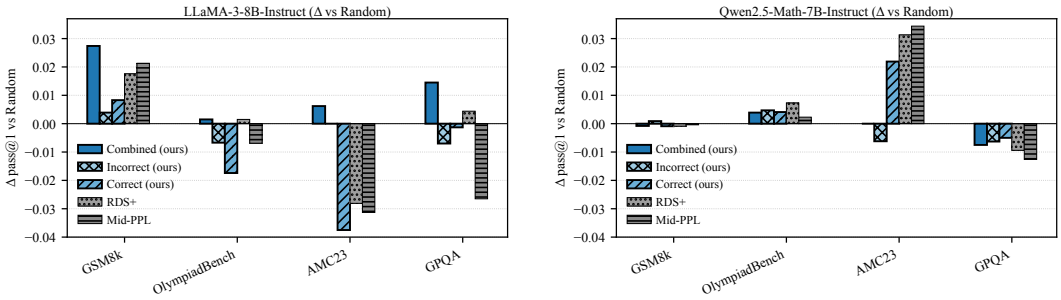


Figure 3: Influence scores $s_C(d), s_I(d)$ for training examples $d \in \mathcal{D}$ with aggressive 50% data pruning strategy (*Combined*).

Table 3: **Results on LLaMA-3-8B-Instruct, pruning 50% of the data.** Pass@1 with $N{=}8$ samples, zero-shot CoT. We fine-tune on math-pruned subsets selected by influence functions (*Correct*, *Incorrect*, *Combined*) and compare against baselines (*Random*, *Mid-PPL*, *RDS+*), as well as the unpruned dataset and base model. Evaluation covers math reasoning benchmarks (AIME24, AMC23, OlympiadBench, GSM8k) and GPQA for non-math transfer. Best in **bold**. Amount of samples not pruned in [brackets].

LLAMA3-8B-INSTRUCT

PRUNING	GSM8K	OLYMPIADBENCH	AMC23	GPQA
Combined [50%] (ours)	**0.7981**	0.0831	**0.0969**	0.1648
RDS+ [50%]	0.7962	0.0767	0.0813	0.1768
Mid-PPL [50%]	0.7797	0.0678	0.0938	0.1679
Random [50%]	0.7799	**0.0843**	0.0875	**0.2134**
None	0.7722	0.0685	0.0813	0.1199
W/o SFT	0.5958	0.0498	0.0781	0.2973



(a) LLaMA-3-8B-Instruct (b) Qwen2.5-Math-7B-Instruct

Figure 4: **Δpass@1 vs. Random pruning on benchmarks.** *(a)* For LLaMA-3-8B-Instruct, our IF-based Combined pruning achieves larger gains than baselines, particularly on GSM8k and OlympiadBench. On GPQA, *Combined* mitigates this degradation more effectively than other pruning strategies. *(b)* For Qwen2.5-Math-7B-Instruct (10% pruning), improvements do not transfer reliably: all methods remain close to *Random*, with AMC23 showing inconsistent deviations.