

Ranking Within Fixed Source Allocations: Controlled Mathematical Instruction Selection

Xin Huang*, Tianle Xu[†], Changjun Ren[‡], Yuhuang Chen[§],
Heng Zhan[¶], Jiawen Huang*, Shijie Jia*

*Dalian Jiaotong University, Dalian, China

[†]Fudan University, Shanghai, China

[‡]Shanghai University of Engineering Science, Shanghai, China

[§]Xi'an Jiaotong-Liverpool University, Suzhou, China

[¶]Harbin Institute of Technology, Harbin, China

20252340@djtu.edu.cn, xutianle@fudan.edu.cn, m325125218@sues.edu.cn

Yuhuang.Chen22@student.xjtlu.edu.cn, zhanheng288@gmail.com

wodenaози@gmail.com, jsj@djtu.edu.cn

Abstract—Generated instruction pools contain multiple records derived from the same source question. Global selection changes both source allocations and within-source ranking, obscuring their separate downstream value. We compare ranked records with random alternatives within exact source, type, and length cells while fixing source allocations. Experiments use 223,553 MetaMathQA records, Qwen2.5-1.5B, and matched input tokens, supervised tokens, and 1,024 training updates. In the retrospective dev256 comparison, within-source LESS exceeds matched random by a mean of 1.302 percentage points across nine construction–training pairs under a shared scorer. All three construction means are positive; the conditional 95% question-bootstrap interval is [0.000, 2.604]. Initialization-gradient gains vary across constructions, while real-source grouping increases coverage but scores below permuted grouping. On the mapping-selected secondary83 set, within-source LESS scores 1.606 percentage points below matched random; an AI-assisted audit flags reference defects. These contrasts distinguish ranking under fixed allocations from global policy differences and show why source coverage, realized training exposure, and evaluation validity should be recorded separately.

Index Terms—instruction tuning, data selection, large language models, source allocation, controlled evaluation

I. INTRODUCTION

Instruction-data selection is a resource-allocation decision: which problems should receive training effort, and which examples should represent each problem? Instruction-generation methods create synthetic instruction-tuning data [1]. In MetaMathQA, retained original-question fields provide source-grouping labels: one mathematical question can yield alternative phrasings, solutions, or task variants. Under a fixed budget, increasing one source’s allocation can reduce exposure to other sources. Dataset size, source coverage, and record value consequently describe different properties of the training collection.

Target-guided methods relate candidate records to an intended task. Influence functions and gradient tracing estimate record influence from parameter sensitivity or training-trajectory gradients [2], [3]. LESS uses candidate update directions and target loss gradients to score relevance [4].

Other work investigates representations, gradient subspaces, diversity, quality ratings, distribution matching, and selection cost [5]–[11]. A global comparison allows source allocation and within-source ranking to change together. Equal input-token ceilings still leave supervised-token exposure and update counts free to differ, which complicates the interpretation of an accuracy difference.

We examine a conditional question: how much downstream value does ranking retain when source identities and allocations are fixed? Our comparison matches source quotas, record types, exact input and supervised lengths, and training updates. Within each eligible cell, the ranked arm takes high-scoring records and the control samples uniformly. Their downstream difference measures record selection together with its resulting training trajectory under the prescribed allocation.

The study uses MetaMathQA [12] records mapped to GSM8K [13] source questions, Qwen2.5-1.5B [14], and a 2M input-token ceiling. Its contribution is a controlled set of contrasts. First, we compare ranking within eligible exact substitution cells under prescribed source allocations; selected manifests cover approximately 63% of pool sources. Second, the retrospective within-source LESS comparison has a positive mean dev contrast across the observed 3×3 grid, whereas initialization-gradient gains change sign across constructions and real-source grouping yields lower mean dev matching. Third, an audit of a mapping-selected evaluation complement demonstrates why source separation and reference validity require separate checks.

For teams curating instruction data, this factorized record makes selection decisions inspectable: source quotas specify the allocation, ranking specifies its contents, and exposure determines the training budget actually consumed. We evaluate these decisions in mathematical instruction tuning.

II. RELATED WORK

Target relevance. LESS combines LoRA warmup, candidate Adam update directions, projected cosine similarity to target

gradients, and checkpoint aggregation [4]. GIST uses target-gradient subspaces [6]; Influence Distillation approximates candidate–target relationships through embeddings [15]. We test relevance-guided ranking under explicit source allocations. An initialization-gradient diagnostic and a subsequent LESS adaptation are distinct scoring paths.

Coverage and redundancy. Representation and selection-algorithm choices can affect targeted selection [5]. G-DIG combines quality-related influence with gradient clustering [7], while normalized gradients and determinantal conditional marginals support diversity-oriented selection [8]. SemDeDup and D4 address semantic redundancy and diversification [16], [17], and duplicate removal changes what a fixed record budget represents [18]. DEITA couples quality and diversity when scoring instruction candidates [19]. Generated-to-original relationships can also be retained during paraphrastic augmentation [20]. We use declared, lexically verified original questions as grouping labels, making source allocations observable while permitting variation within each group.

Budget controls. Shared record budgets, multiple runs, and length diagnostics have precedents [5]. Compute-Constrained Data Selection studies token budgets and combined selection and training cost [9], and importance resampling matches a target distribution under a fixed budget [11]. LESS discusses response-length gradient bias and fixed-epoch versus aligned-step training [4]. Our additional control fixes source identities and per-source quotas, then matches input tokens, shifted supervised tokens, and updates within exact substitution cells. It addresses conditional ranking value rather than end-to-end efficiency.

III. CONTROLLED SELECTION PROTOCOL

A. Source Mapping and Evaluation Sets

We normalize MetaMathQA `original_question` fields using NFKC and whitespace normalization and retain unique lexical matches to GSM8K train questions. Of 395,000 MetaMathQA records, 240,000 map uniquely to 7,390 of the 7,473 GSM8K train questions. Removing records from the source-disjoint probe256 and dev256 partitions leaves 223,553 candidates from 6,878 sources. The original collection includes 8,438 probe-source and 8,009 dev-source derivatives; these are excluded from the pool. The other 155,000 records lack this unique mapping and are also excluded.

Probe256 supplies labeled target information for scoring. Dev256 contains 256 held-out GSM8K train questions. Secondary83 contains the 83 train questions outside the 7,390 uniquely mapped sources and is selected by mapping absence. The pool and all three evaluation-related sets have pairwise disjoint source IDs and normalized queries. Saved selection manifests, training orders, and update blocks show no probe, dev, or secondary source among the 51 downstream fits. Of these fits, 39 use probe-conditioned scores and 12 use score-independent random selection. The official GSM8K test split was not read. Early preparation accessed the full train table, so secondary83 is a coverage-shift set rather than an untouched benchmark.

Lexical source mapping does not authenticate upstream generation lineage or exclude semantic overlap and pretraining contamination. Source groups contain different MetaMathQA types, including answer augmentation, rephrasing, FOBAR, and self-verification variants; a common source label does not make every record semantically equivalent.

B. Exposure and Optimization

We format each prompt as an instruction description followed by the `### Instruction:` question field and the `### Response:` marker. Sequences are truncated to at most 1,024 tokens. Let c_i denote actual nonpadding input tokens and u_i supervised tokens after the causal shift. Selection satisfies

$$\sum_{i \in S} c_i \leq 2,000,000. \quad (1)$$

Records are indivisible. If a candidate unit does not fit, scanning continues. Within-source manifests nearly exhaust this ceiling; the grouping-policy comparison uses 1.63–1.66M tokens because its support differs.

Each fit starts from the same frozen Qwen2.5-1.5B base with fresh LoRA [21] and optimizer states. Downstream LoRA targets q/v projections with rank 8, alpha 16, and dropout 0. AdamW [22] uses learning rate 2×10^{-4} , betas (0.9, 0.999), epsilon 10^{-8} , and weight decay 0.01. Each selected record is traversed once. The seed-specific ordered manifest is partitioned into 1,024 nonempty contiguous blocks. One update per block divides summed answer-token cross-entropy by that block’s total supervised tokens. Matching total exposure and updates still allows record content, order, block membership, and optimizer trajectories to differ.

C. Ranking Within Fixed Allocations

An exact cell is (source ID, type, c_i, u_i). Sorted cells are shuffled by construction seed. For a cell containing m records of input cost c , with remaining budget R , both arms select

$$q = \min(\lfloor m/2 \rfloor, \lfloor R/c \rfloor). \quad (2)$$

The ranked arm takes the q highest-scoring records, breaking ties by row ID. The random arm samples q uniformly without replacement using a cell-specific seed independent of scores. The two arms can share records. Singleton cells receive zero quota. Thus, only cells with exact substitutes and sufficient remaining budget contribute selected records; the resulting comparison fixes source identities, source quotas, type composition, record count, and both token totals (Fig. 1).

Construction seeds 42, 43, and 44 yield source coverage of 63.36%, 62.69%, and 63.17%, respectively. Each construction is crossed with training seeds 42, 43, and 44. The primary estimand is the equal-weight mean over this observed 3×3 grid, conditional on the shared scorer, manifests, and fitted models.

Exact cell: same source, type, input and supervised lengths

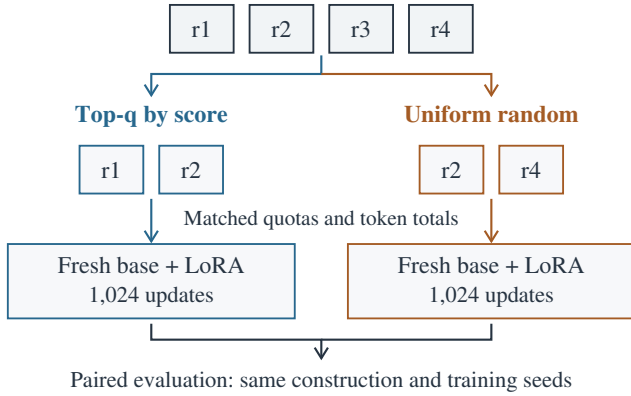


Fig. 1. Ranking within fixed allocations. One illustrative exact cell contains four records with quota $q = 2$; the arms can overlap. Selections across cells form matched manifests before separate training and paired evaluation. Scores come from LESS or the distinct initialization-gradient diagnostic.

D. Grouping and Global Comparisons

A separate experiment compares real source labels with permuted labels preserving the group-size distribution. Shared cells are (type, c_i, u_i) . Common cells are shuffled; each arm independently shuffles unused groups before allocating the same number. Each allocated group contributes its highest-scoring record in that cell, with row-ID tie-breaking, and is then permanently removed. Cell quotas are bounded by both arms’ available groups and remaining budget.

Real labels cover 6,574–6,589 true sources (95.58–95.80%), compared with 4,370–4,386 (63.54–63.77%) for permuted labels. Paired arms exactly match input tokens, supervised tokens, and 1,024 updates. Construction and permutation seeds are bound as $c42/p1042$, $c43/p1043$, and $c44/p1044$. Changing group–cell associations and concrete allocations makes this a joint grouping-policy comparison rather than an isolated causal effect of coverage.

Global LESS and global row random instead allow source allocations to vary. Each uses selection seed42 and three training seeds, the same input ceiling, and 1,024 updates. Their record counts and supervised-token totals differ, so they estimate overall policy differences.

IV. SCORING AND EVALUATION

A. Initialization Gradients and LESS Adaptation

The historical diagnostic computes 4,096-dimensional CountSketch features from response-token mean-loss gradients at q/v LoRA initialization. Ranking uses cosine similarity to the raw mean of probe features. It omits warmup and candidate Adam update directions and is reported separately from LESS.

The LESS adaptation follows author-code commit `8abf9628` [4]. A shared warmup uses 5% of the pool (11,177 records, seed42), four epochs, and one checkpoint per epoch. Warmup LoRA targets $q/k/v/o$ with rank 128, alpha 512, and dropout 0.1. Maximum length is 2,048; learning rate is 2×10^{-5} , weight decay 0, effective batch32,

clipping 1.0, a linear scheduler, and warmup ratio 0.03. There are 350 updates per epoch; the final nine-record batch is normalized by its actual size.

At each checkpoint, candidate gradients and saved Adam moments define update directions, while probe256 supplies ordinary loss gradients. TRAK 0.1.3 BasicProjector uses an 8,192-dimensional Rademacher projection, seed0, with FP16 gradients and projection matrices and FP32 saved features. Extraction uses evaluation mode with autograd enabled and dropout disabled. We normalize projected features, average candidate–probe cosine over probe questions, and aggregate checkpoints using each epoch’s actual mean learning rate. Projection batch16, the bound collector’s default, scores all candidates. It failed all four numerical-equivalence gates against historical batch8 on the checked shard; because gate thresholds and resulting top- q rank effects were not retained, this establishes a numerical-sensitivity boundary rather than equivalence evidence. The extension uses batch16 scores throughout. All LESS selections share this warmup and scoring realization; downstream fits start from the base model rather than the warmup adapter. The public-script scheduler and effective batch differ from the LESS paper appendix, and the target set and downstream recipe are task adaptations.

B. Execution Grid and Matching Metrics

The historical within-source and grouping experiments each contain 18 fits, totaling 36. Within-source LESS adds nine fits and reuses the nine exactly replayed historical random models. Global LESS and row random add six, giving 51 downstream models.

Historical evaluation uses batch1 greedy decoding on dev256, with at most 512 new tokens. The LESS extension uses batch8, left padding, attention masks, and the same greedy cap. All extension paired differences use batch8 predictions. Full-question token equivalence between batch1 and batch8 is unestablished, so their scores are interpreted separately.

Strict matching takes the last ##### or The answer is: marker and requires its entire suffix to match a signed decimal, allowing valid thousands separators, a dollar prefix, and a final period. Fractions, units, and extra prose are rejected; Decimal arithmetic compares values exactly. Trailing-number matching extracts the last signed decimal under frozen boundaries, requires no marker, and has no strict-parser fallback. Exact rules are retained in `public_selection/experiment.py` and `rescore.py`. Rejected parses and capped generations remain in the denominator; matching assesses numerical reference agreement rather than complete reasoning.

The extension initially contained 66 model-set evaluations. Nine matched-random dev evaluations were added after aggregate results were available; a plan froze these pairs before supplemental prediction, without retraining or hyperparameter changes. The resulting within-source dev comparison is retrospective, and the 75 evaluations reuse questions rather than forming independent repetitions.

TABLE I
SELECTION EXPOSURE. WITHIN-SOURCE ROWS APPLY TO BOTH LESS
AND RANDOM; EVERY FIT USES 1,024 UPDATES.

Manifest	Records	Input tokens	Supervised tokens
Within c42	9,718	1,999,998	1,233,666
Within c43	9,678	1,999,991	1,235,783
Within c44	9,672	1,999,926	1,236,156
Global LESS	10,196	1,999,953	1,203,674
Global random	7,872	1,999,967	1,321,452

C. Uncertainty

We take 10,000 paired question-bootstrap draws, seed42, with shared question indices across every model pair [23]. The 2.5th and 97.5th percentiles form conditional intervals. Equivalently, paired differences are first averaged per question and those question averages are resampled. These intervals omit scorer, construction, and retraining uncertainty and are unadjusted for multiple comparisons; paired designs should state these choices explicitly [24]. Sample SD with ddof = 1 describes fitted-pair variation rather than a seed confidence interval.

V. RESULTS AND ANALYSIS

A. Matched Exposure and Ranking Value

Table I shows that within-source LESS and random match both input and supervision. Global LESS instead uses more records and fewer supervised tokens than global random despite nearly identical input totals. Thus an equal input budget alone does not align all training exposure.

On the observed 3×3 dev256 grid, within-source LESS has a mean strict-matching difference of +1.302 pp over matched random (Table II). Figure 2 shows the nine fitted-pair differences: six are positive and three negative. Construction means are +2.474, +0.781, +0.651 pp; training-seed means are +2.474, +0.260, +1.172 pp. The conditional interval begins at zero within numerical precision.

Deleting one entire construction leaves mean differences of +0.716 to +1.628 pp; deleting one training seed gives +0.716 to +1.823 pp. Figure 3 shows these six single-omission means together with the corresponding secondary83 values. The sign persists within these finite-grid sensitivity checks. Each cell still contains one fit and all LESS selections share scoring, so row and column deletions do not quantify uncertainty for a new scorer or a complete rerun.

Trailing-number dev matching is 79.818% for LESS and 78.429% for random, a +1.389 pp difference with conditional interval [0.043, 2.734]. Two additional LESS numerical matches produce this increment. It is a parsing diagnostic on the same predictions.

B. Initialization Ranking and Source Coverage

Table III reports historical batch1 results. Initialization-gradient ranking has higher mean scoring cosine in every construction, yet downstream differences change sign:

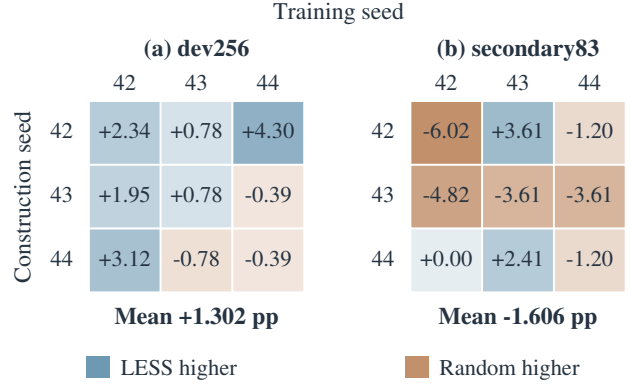


Fig. 2. Strict reference-match differences (LESS minus matched random, pp), inference batch8. Rows are construction seeds; columns are training seeds. Both panels use the same zero-centered color scale. Each cell is one fitted pair; these grids describe observed variation, not full-procedure uncertainty. The dev comparison was retrospective; secondary83 has reference-validity limitations.

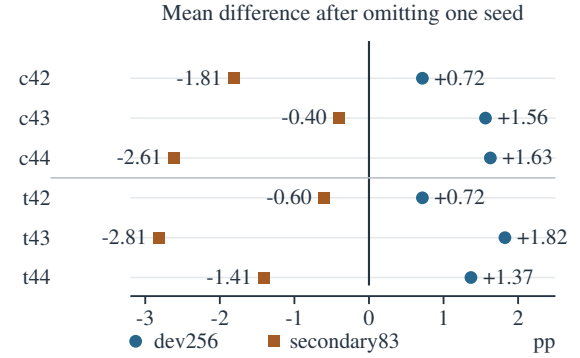


Fig. 3. Single-omission means, LESS minus matched random (pp). Each row deletes one construction seed or one training seed from the 3×3 grid; full-grid means are +1.302 pp on dev256 and -1.606 pp on secondary83. Both series retain their sign under every deletion. Colors follow Fig. 2: blue for positive and orange for negative means. These are descriptive finite-grid checks, not uncertainty intervals.

+2.734, -1.432, +0.391 pp across construction means. Individual correct-count differences range from -14 to +9. Higher scoring similarity therefore does not consistently translate into higher downstream matching under this diagnostic.

Real-source grouping increases coverage from approximately 64% to 96%, slightly increases mean cosine, and scores 0.998 pp below permuted grouping. Its three construction means are -1.432, -0.260, -1.302 pp. This result concerns the tested grouping policy, whose allocations and group-cell associations differ between arms. It establishes that broader coverage need not improve the measured outcome under matched exposure.

C. Coverage Shift and Reference Validity

Within-source LESS scores 1.606 pp below random on secondary83. Correct-count differences are -5, +3, -1, -4, -3, -3, 0, +2, -1. Deleting one construction or training seed retains a negative grid mean, with ranges

TABLE II
STRICT MATCHING IN THE LESS EXTENSION, INFERENCE BATCH8. DIFFERENCES ARE LESS MINUS RANDOM. SD DESCRIBES FITTED PAIRS; INTERVALS RESAMPLE SHARED QUESTIONS CONDITIONAL ON FITTED MODELS. THE LAST ROW WAS ADDED RETROSPECTIVELY.

Set	Scope	Pairs	LESS (%)	Random (%)	Δ (pp)	SD (pp)	Conditional 95% interval
secondary83	Within source	9	5.756	7.363	-1.606	3.244	[-3.481, 0.000]
secondary83	Global	3	8.434	8.032	+0.402	3.032	[-4.016, 4.819]
dev256	Global	3	78.385	79.818	-1.432	2.542	[-4.818, 1.823]
dev256	Within source (added)	9	79.731	78.429	+1.302	1.747	[0.000, 2.604]

TABLE III
HISTORICAL DEV256 STRICT MATCHING, INFERENCE BATCH1. EACH COMPARISON HAS NINE CONSTRUCTION-TRAINING PAIRS.

Comparison	Left (%)	Right (%)	Δ (pp)	SD (pp)
Init. gradient - random	79.123	78.559	+0.564	3.039
Real - permuted	79.427	80.425	-0.998	1.915

-2.610 to -0.402 and -2.811 to -0.602 pp, respectively (Fig. 3). Global differences are negative on dev and positive on secondary; both intervals cross zero. Because global allocations, content, and supervision differ, these results do not isolate ranking within sources.

Secondary means of 5.8-8.4% are much lower than dev means of 78.4-79.8%. Replaying frozen parsers on the 1,494 secondary predictions from the 18 within-source models finds 12 strict rejections, seven cap-reaching outputs, and no empty outputs. Relaxed parsing adds one match. Of the predictions passing strict parsing, 697 LESS and 687 random outputs disagree numerically with their reference. Formatting and truncation cannot explain the overall gap.

A retrospective qualitative sample exposed reference contradictions. In one case, the question specifies 20 trucks and five times as many tanks. Both models output 120, consistent with $20 + 5 \times 20$, but the reference’s final marker says 140 despite computing 120 in its body. Another case contains both a flawed reference and incorrect model reasoning. A reference defect therefore does not certify a model answer.

An AI-assisted census divides all 83 references into three disjoint shards and withholds model outputs and correctness from the auditors. A second AI-assisted review rechecks all 18 initial clear-error candidates; integrated categories are 16 clear reference errors, 26 ambiguous or underspecified cases, 40 reference-consistent cases, and one unresolved case. Auditors see saved references and know prior issues were found; ambiguous and consistent cases were not all independently rechecked, and there is no human adjudication. All queries, references, and IDs match the frozen local GSM8K train source, authenticating local transfer rather than the upstream distribution.

We retain original references, scores, and denominators and compute no corrected accuracy or revised ranking. The audit localizes some disagreements but does not explain the overall low score or estimate reference-error prevalence in GSM8K. Secondary83 cannot supply independent confirmation of a

mathematical-ability difference.

VI. DISCUSSION AND LIMITATIONS

Implications for data curation. Source allocations, record ranking, and training exposure should be recorded as separate decisions. Fixed allocations make relevance-guided substitution measurable; global policies remain useful comparisons but combine these factors. Increased coverage is an observable structural property whose utility must be assessed downstream. In shared instruction-data workflows, source IDs, quotas, token counts, and update counts provide a concrete record of those decisions.

Scope and uncertainty. The study covers one English mathematical pool, one 1.5B model, LoRA, and a roughly 2M-token budget. Exact matching restricts support to cells with alternatives and sufficient remaining budget; singleton cells contribute zero quota. The approximately 63% source-coverage figure describes selected manifests, not the fraction of candidate records supported or the strength of replacement within cells. Content, ordering, block membership, and AdamW trajectories remain coupled; grouping comparisons additionally bind construction and permutation seeds. Conditional bootstrap and deletion analyses do not establish full-procedure uncertainty.

Evaluation and provenance. Dev256 was repeatedly used, and its matched-random extension was retrospective. Secondary83 is mapping-selected, was accessible during early preparation, and has unresolved reference validity. Independent evaluation remains necessary for broader ability claims. Strict matching checks reference agreement rather than complete reasoning. Frozen data and model versions were acquired through public mirrors with incomplete independent verification against official hosts and licenses. Source/query separation does not establish semantic or pretraining cleanliness.

Cost and reproducibility. Warmup records 5,391.253 seconds for 1,400 updates, and 516 scoring workers sum to 620,701.535 seconds. Overlapping durations are neither elapsed time nor exclusive GPU cost; missing timings for aggregation, CPU selection, failures, and some historical processing prevent an end-to-end efficiency ranking. Saved predictions and original scoring functions support offline CPU rescoring of all 75 evaluations (10,377 outputs). The local package excludes weights, raw training data, the candidate pool, probe, and gradient matrices; it supports saved-result verification rather than member-level selection reconstruction, retraining, or regenerated inference.

VII. CONCLUSION

Fixing source identities and quotas makes ranking within an allocation observable separately from broader selection-policy changes. With input tokens, supervision, and updates matched, the retrospective finite-grid LESS comparison has an observed +1.302 pp dev mean, positive across construction averages but negative in three of nine fitted pairs. Initialization-gradient gains vary by construction, and broader source coverage under real grouping yields a lower mean score. The mapping-selected complement further shows that source separation and reference validity are distinct requirements. These conditional observations make allocation, exposure, and evaluation validity explicit components of instruction-data selection. Source quotas and realized training exposure should therefore accompany reported selection gains.

ACKNOWLEDGMENT

Ducc, a generative AI assistant, assisted manuscript restructuring, condensation, and wording throughout the abstract and Sections I–VII, and preparation of Figs. 1–3. AI-assisted reference annotation is described in Section V-C.

REFERENCES

- [1] Y. Wang, Y. Kordi, S. Mishra, A. Liu, N. A. Smith, D. Khashabi, and H. 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)*. Toronto, Canada: Association for Computational Linguistics, 2023, pp. 13 484–13 508.
- [2] P. W. Koh and P. Liang, “Understanding black-box predictions via influence functions,” in *Proceedings of the 34th International Conference on Machine Learning*, ser. Proceedings of Machine Learning Research, vol. 70. PMLR, 2017, pp. 1885–1894.
- [3] G. Pruthi, F. Liu, S. Kale, and M. Sundararajan, “Estimating training data influence by tracing gradient descent,” in *Advances in Neural Information Processing Systems*, vol. 33. Curran Associates, Inc., 2020, pp. 19 920–19 930.
- [4] M. Xia, S. Malladi, S. Gururangan, S. Arora, and D. Chen, “LESS: Selecting influential data for targeted instruction tuning,” in *Proceedings of the 41st International Conference on Machine Learning*, ser. Proceedings of Machine Learning Research, vol. 235. PMLR, 2024, pp. 54 104–54 132, method-reading version: arXiv:2402.04333v3, 13 June 2024.
- [5] N. V. Nayak, P. Rodriguez-Diaz, N. Hulkund, S. Beery, and D. Alvarez-Melis, “A critical look at targeted instruction selection: Disentangling what matters (and what doesn’t),” in *Proceedings of the 43rd International Conference on Machine Learning*, ser. Proceedings of Machine Learning Research, vol. 306. PMLR, 2026, pp. 92 008–92 043.
- [6] G. Min, T. Huang, K. Wan, and C. Chen, “GIST: Targeted data selection for instruction tuning via coupled optimization geometry,” in *Proceedings of the 43rd International Conference on Machine Learning*, ser. Proceedings of Machine Learning Research, vol. 306. PMLR, 2026, pp. 88 886–88 912.
- [7] X. Pan, L. Huang, L. Kang, Z. Liu, Y. Lu, and S. Cheng, “G-DIG: Towards gradient-based DIverse and hiGh-quality instruction data selection for machine translation,” in *Proceedings of the 62nd Annual Meeting of the Association for Computational Linguistics (Volume 1: Long Papers)*, L.-W. Ku, A. Martins, and V. Srikumar, Eds. Bangkok, Thailand: Association for Computational Linguistics, Aug. 2024, pp. 15 395–15 406.
- [8] P. Wang, Y. Shen, Z. Guo, M. Stallone, Y. Kim, P. Golland, and R. Panda, “Diversity measurement and subset selection for instruction tuning datasets,” *arXiv preprint arXiv:2402.02318v1*, 2024, version 1, 4 February 2024.
- [9] J. O. Yin and A. M. Rush, “Compute-constrained data selection,” in *International Conference on Learning Representations*, 2025, pp. 36 708–36 729.
- [10] A. Wettig, A. Gupta, S. Malik, and D. Chen, “QuRating: Selecting high-quality data for training language models,” in *Proceedings of the 41st International Conference on Machine Learning*, ser. Proceedings of Machine Learning Research, vol. 235. PMLR, 2024, pp. 52 915–52 971.
- [11] S. M. Xie, S. Santurkar, T. Ma, and P. S. Liang, “Data selection for language models via importance resampling,” in *Advances in Neural Information Processing Systems*, vol. 36. Curran Associates, Inc., 2023, pp. 34 201–34 227.
- [12] L. Yu, W. Jiang, H. Shi, J. Yu, Z. Liu, Y. Zhang, J. T. Kwok, Z. Li, A. Weller, and W. Liu, “MetaMath: Bootstrap your own mathematical questions for large language models,” in *International Conference on Learning Representations*, 2024.
- [13] K. Cobbe, V. Kosaraju, M. Bavarian, M. Chen, H. Jun, L. Kaiser, M. Plappert, J. Tworek, J. Hilton, R. Nakano, C. Hesse, and J. Schulman, “Training verifiers to solve math word problems,” *arXiv preprint arXiv:2110.14168v2*, 2021, version 2, 18 November 2021.
- [14] Qwen Team, “Qwen2.5 technical report,” *arXiv preprint arXiv:2412.15115v2*, 2025, version 2, 3 January 2025.
- [15] M. Nikdan, V. Cohen-Addad, D. Alistarh, and V. Mirrokni, “Efficient data selection at scale via influence distillation,” in *Advances in Neural Information Processing Systems*, vol. 38. Curran Associates, Inc., 2025, pp. 143 229–143 265.
- [16] A. Abbas, K. Tirumala, D. Simig, S. Ganguli, and A. S. Morcos, “SemDeDup: Data-efficient learning at web-scale through semantic deduplication,” *arXiv preprint arXiv:2303.09540v3*, 2023, version 3, 22 March 2023.
- [17] K. Tirumala, D. Simig, A. Aghajanyan, and A. S. Morcos, “D4: Improving LLM pretraining via document de-duplication and diversification,” in *Advances in Neural Information Processing Systems*, vol. 36. Curran Associates, Inc., 2023, pp. 53 983–53 995, datasets and Benchmarks Track. Appendix-reading version: arXiv:2308.12284v1.
- [18] K. Lee, D. Ippolito, A. Nystrom, C. Zhang, D. Eck, C. Callison-Burch, and N. Carlini, “Deduplicating training data makes language models better,” in *Proceedings of the 60th Annual Meeting of the Association for Computational Linguistics (Volume 1: Long Papers)*. Dublin, Ireland: Association for Computational Linguistics, 2022, pp. 8424–8445.
- [19] W. Liu, W. Zeng, K. He, Y. Jiang, and J. He, “What makes good data for alignment? A comprehensive study of automatic data selection in instruction tuning,” in *International Conference on Learning Representations*, 2024.
- [20] R. Culkin, J. E. Hu, E. Stengel-Eskin, G. Qin, and B. Van Durme, “Iterative paraphrastic augmentation with discriminative span alignment,” *Transactions of the Association for Computational Linguistics*, vol. 9, pp. 494–509, 2021.
- [21] E. J. Hu, Y. Shen, P. Wallis, Z. Allen-Zhu, Y. Li, S. Wang, L. Wang, and W. Chen, “LoRA: Low-rank adaptation of large language models,” in *International Conference on Learning Representations*, 2022.
- [22] I. Loshchilov and F. Hutter, “Decoupled weight decay regularization,” in *International Conference on Learning Representations*, 2019.
- [23] B. Efron and R. J. Tibshirani, *An Introduction to the Bootstrap*, ser. Monographs on Statistics and Applied Probability. Boca Raton, FL: Chapman & Hall/CRC, 1993, no. 57.
- [24] R. Dror, G. Baumer, S. Shlomov, and R. Reichart, “The Hitchhiker’s guide to testing statistical significance in natural language processing,” in *Proceedings of the 56th Annual Meeting of the Association for Computational Linguistics (Volume 1: Long Papers)*. Melbourne, Australia: Association for Computational Linguistics, 2018, pp. 1383–1392.