

A Strategic Coordination Framework of Small LMs Matches Large LMs in Data Synthesis

Xin Gao¹, Qizhi Pei^{2,1}, Zinan Tang¹, Yu Li¹, Honglin Lin¹,
Jiang Wu¹, Lijun Wu¹, Conghui He^{1*}

¹Shanghai Artificial Intelligence Laboratory, ²Renmin University of China

{gaoxin,wulijun,heconghui}@pjlab.org.cn

🤗 <https://huggingface.co/collections/GX-XinGao/GRA>

🐙 <https://github.com/GX-XinGao/GRA>

Abstract

While data synthesis and distillation are promising strategies to enhance small language models, current approaches heavily rely on Large Language Models (LLMs), which suffer from high computational costs, environmental inefficiency, and potential biases inherited from monolithic architectures. In contrast, smaller LMs are more accessible and sustainable, but their individual capabilities often fall short in generating high-quality, diverse, and reliable data. Inspired by collaborative human processes (e.g., peer review), we propose a multiple small LMs involved framework, **GRA**, that aggregates specialized roles across small LMs to iterative refinement and quality control typically achieved by a single large LM. In this collaborative framework, multiple small LMs assume distinct roles—**Generator**, **Reviewer**, and **Adjudicator**—to simulate a peer-review-inspired data synthesis pipeline. The Generator proposes initial data samples, the Reviewer critiques their quality and diversity, and the Adjudicator resolves conflicts to finalize the output. By decomposing the synthesis process into specialized sub-tasks, collaborative small LMs can achieve data-level parity with distillation from large LMs. Through experiments across multiple benchmarks, we demonstrate that GRA-produced data matches or exceeds the quality of single large LM outputs, e.g., Qwen-2.5-72B-Instruct. Our results challenge the necessity of monolithic large models for high-quality data synthesis, advocating instead for strategic coordination of smaller agents.

1 Introduction

The advent of Large Language Models (LLMs) with hundreds of billions of parameters has revolutionized artificial intelligence (Dubey et al., 2024a; Yang et al., 2024a; OpenAI et al., 2023; Liu et al.,

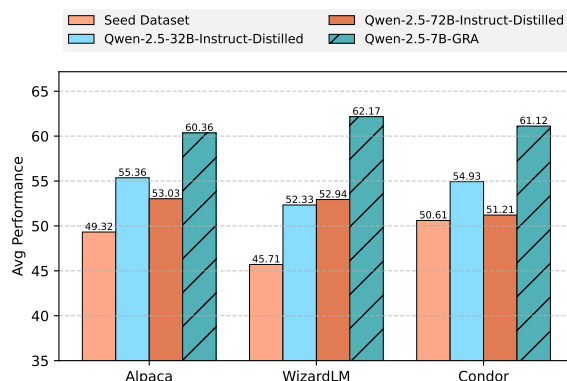


Figure 1: Average performance across GRA, vanilla seed dataset and large LMs distilled data with Qwen-2.5-7B base model.

2024a), achieving state-of-the-art (SOTA) performance across diverse domains such as natural language understanding (Naveed et al., 2023), reasoning (Guo et al., 2025; OpenAI, 2024), math problem solving (Ahn et al., 2024), and code generation (Jiang et al., 2024). Despite their remarkable capabilities, these monolithic models demand immense computational resources and energy, posing significant barriers to deployment in resource-constrained scenarios like edge devices or latency-sensitive applications (Zheng et al., 2024b). This paradox has spurred growing interest in developing smaller-scale LMs that balance efficiency and effectiveness, positioning compact yet capable models as a critical research frontier in sustainable AI (Chen and Varoquaux, 2024).

To enhance small LMs performance, knowledge distillation (Hinton, 2015)—transferring capabilities from large models via data synthesis—has emerged as a dominant paradigm (Xu et al., 2024b). However, conventional distillation inherently depends on the availability of powerful LLMs (Wang et al., 2024a), which introduces practical limitations: deploying large LMs for data generation incurs prohibitive costs, especially in privacy-

* Corresponding author.

sensitive or security-critical scenarios requiring local execution (Hagos et al., 2024; Yao et al., 2024). While small LMs offer computational accessibility, their individual capacities are inherently constrained, struggling to match the quality, diversity, and reliability of data synthesized by their larger counterparts. This raises a pivotal question: *Can we design a resource-efficient framework that enables small models to collectively rival the distillation capabilities of monolithic LLMs?*

To address this challenge, we propose **GRA**, a collaborative framework inspired by human peer-review mechanisms. Diverging from traditional single-model distillation, GRA orchestrates multiple specialized small LMs into three synergistic roles: (1) **Generator**, which proposes candidate data samples; (2) **Reviewer**, which evaluates quality and diversity through iterative critiques; and (3) **Adjudicator**, which resolves conflicts to finalize outputs. Notably, the synthetic data generated by a single Generator will be evaluated by multiple Reviewers, and the Adjudicator is to resolve significant disagreements among the Reviewers by reassessing the data. By decomposing the synthesis pipeline into discrete sub-tasks and enabling role-specific optimization, GRA mimics the refinement processes typically concentrated within a single large LM. This collaborative architecture not only circumvents the resource-intensity of large models but also amplifies small models’ collective intelligence through task specialization and iterative quality control—effectively operationalizing the “wisdom of crowds” principle in model ensembles.

We validate GRA through extensive experiments on multiple domains, including general question answering, reasoning, mathematical and coding tasks. Remarkably, our framework achieves data quality comparable to or exceeding outputs from SOTA monolithic LLMs like Qwen-2.5-72B-Instruct, while requiring orders-of-magnitude fewer computational resources. These results empirically demonstrate that strategically coordinated small LMs can collectively surpass the distillation performance of individual large LM.

2 Related Work

2.1 Data Synthesis

The rapid evolution of LLMs has increased the demand for high-quality, diverse training data, creating a bottleneck in AI development (Zhao et al., 2023). Synthetic data is a promising direction to

improve data. A dominant paradigm for synthetic data leverages knowledge distillation from LLMs. WizardLM (Xu et al., 2024a) iteratively enhances seed data based on predefined evolutionary rules and instructions. MAGPIE (Xu et al., 2025) utilizes pre-defined query templates to directly prompt the LLMs to synthesize data while not relying on seed data. MuggleMath (Li et al., 2024a) employs LLMs to rewrite and enhance seed datasets by generating new variations of problem statements and corresponding answers.

In addition to single-LLM-based approaches, several works also employ multiple agents for data generation (Mitra et al., 2024; Majumdar et al., 2024). AgentInstruct (Mitra et al., 2024) leverages GPT-4 (OpenAI et al., 2023) to facilitate data generation and curation through the use of multiple agents. Similarly, Genetic Instruct (Majumdar et al., 2024) defines three distinct roles—Instructor-LLM, Coder-LLM, and Judge-LLM—to generate code data. Unlike previous multi-agent frameworks that rely on a single LLM like GPT-4 for repeated deployment across different stages, GRA pioneers a collaborative paradigm that coordinates diverse types of small LMs through specialized role allocation to collectively rival the synthetic data quality of single large LM.

2.2 Data Curation

The efficacy of synthetic data critically hinges on post-synthesis curation (Long et al., 2024). Early curation methodologies predominantly relied on surface-level heuristics, such as repetition counts (Salkar et al., 2022) and lexical diversity metrics (Li et al., 2016; Zhu et al., 2018; Shu et al., 2019), which inadequately captured semantic coherence and task-specific utility. Recent works shift toward LLM-driven curation frameworks. Initial efforts employed perplexity (PPL) and instruction following difficulty score (IFD) (Li et al., 2024b) to filter low-quality samples. The emergence of “LLM-as-judge” paradigms further revolutionized this domain, leveraging well-tuned LLMs for quality assessment (Gu et al., 2024). In contrast, while sharing the “LLM-as-judge” basis, our GRA framework differs by using multiple specialized small LMs for diverse evaluations. Unlike single-LLM approaches that risk bias, GRA employs a multi-agent system and a two-tier adjudication to ensure balanced and comprehensive data assessment.

3 Methodology

Preliminary We first provide essential background and definitions for key components with the rationale behind our design choices. Specifically, we denote \mathcal{M} as the pool of small LMs employed in the framework, and data pool at iteration t as $\mathcal{D}^{(t)} = \{(x_i, y_i, d_i, k_i, s_i)\}_{i=1}^N$, where each sample in $\mathcal{D}^{(t)}$ consists of an instruction x_i , a response y_i , a domain label d_i , a 3-tuple of keywords k_i , and a summary s_i . The domain label d_i indicates the task type of the sample, which is categorized into seven domains: Coding, Math, Question and Answering, Reasoning, Role Play, Language, and Creation, which are designed to aim for covering the most general domains. The 3-tuple keywords k_i are generated by small LMs to capture the core concepts of each data sample. In addition, we include a summary s_i to provide context and clarify keyword relationships, enhancing the overall understanding of the data. More details such as domain definitions and construction of $\mathcal{D}^{(0)}$ -seed dataset are shown in the Appendix A.

3.1 Pipeline Overview

The overview of our GRA is shown in Figure 2. GRA integrates domain-specific generation with collaborative curation by orchestrating three specialized roles. First, the Generator produces initial samples using few-shot prompting tailored to the domain. Next, multiple Reviewers collaboratively assess the quality of these samples. Finally, an Adjudicator resolves any conflicting evaluations through consensus-driven arbitration. A post-processing module further refines the results by eliminating redundant outputs via embedding deduplication and metadata enrichment. Moreover, to prevent model specialization bias, GRA dynamically assigns distinct model from \mathcal{M} to each role through random selection and during generation.

3.2 Data Generator

To enhance synthetic data diversity while maintaining domain consistency, we carefully design the generation process to systematically integrate conceptual knowledge and contextual patterns. For each instance, we randomly select one Model $M_G \in \mathcal{M}$ as the Generator. For a given domain, the generation workflow comprises three coordinated stages: First, M_G analyzes randomly select keywords-summary pairs (k_i, s_i) from the current

domain corpus, synthesizing a novel 3-tuple keywords k' by extracting and combining meaningful semantic patterns and attributes from the input data. Second, drawing inspiration from contextual summaries of seed examples, M_G contextualizes the synthesized keywords into an executable instruction x' . Last, M_G produces response y' that appropriately addresses the generated instruction x' .

3.3 Data Reviewer

To ensure equitable data evaluation and mitigate individual model biases, the Reviewer module performs a rigorous multi-agent quality assessment through coordinated evaluation protocols. We randomly select N_R reviewers from $\mathcal{M} \setminus M_G$ to form the review committee $R = \{M_{R_i}\}_{i=1}^{N_R}$.

The review process includes two stages: *instruction evaluation* and *response evaluation*. During the *instruction evaluation* phase, each Reviewer independently verifies instruction x' against three binary criteria: “Reasonableness, Completeness, and Clarity”. Immediate rejection occurs if any Reviewer M_{R_i} flags any criterion failure. Approved instructions then advance to the *response evaluation* stage, where each Reviewer conducts a granular analysis of the generated response y' from six dimensional metrics: “Correctness, Clarity, Completeness, Relevance, Coherence, and Ethicality”. Then each Reviewer quantitatively assesses these dimensions by assigning integer scores from 0 to 10 and the averaged score serves as the final score. Additionally, each Reviewer provides detailed textual feedback that delineates both the strengths and weaknesses of the generated response.

In the end, the committee R aggregates assessments through calculating the mean composite score μ_R and assessing consensus reliability through standard deviation σ_R . After all, the committee R makes the final decision: (a) **Rejection**: If $\mu_R < \tau$. (b) **Acceptance**: If $\mu_R \geq \tau$ with $\sigma_R \leq \delta$. (c). **Adjudication**: If $\mu_R \geq \tau$ but $\sigma_R > \delta$, where τ and δ are predefined thresholds.

3.4 Data Adjudicator

Adjudicator is designed for reassessing the disputed samples. For each disputed sample, we randomly select an Adjudicator M_A from $\mathcal{M} \setminus (M_G \cup R)$ to perform consensus-driven arbitration. For each disputed sample, based on consolidating Reviewers’ feedback, M_A conducts a comprehensive re-

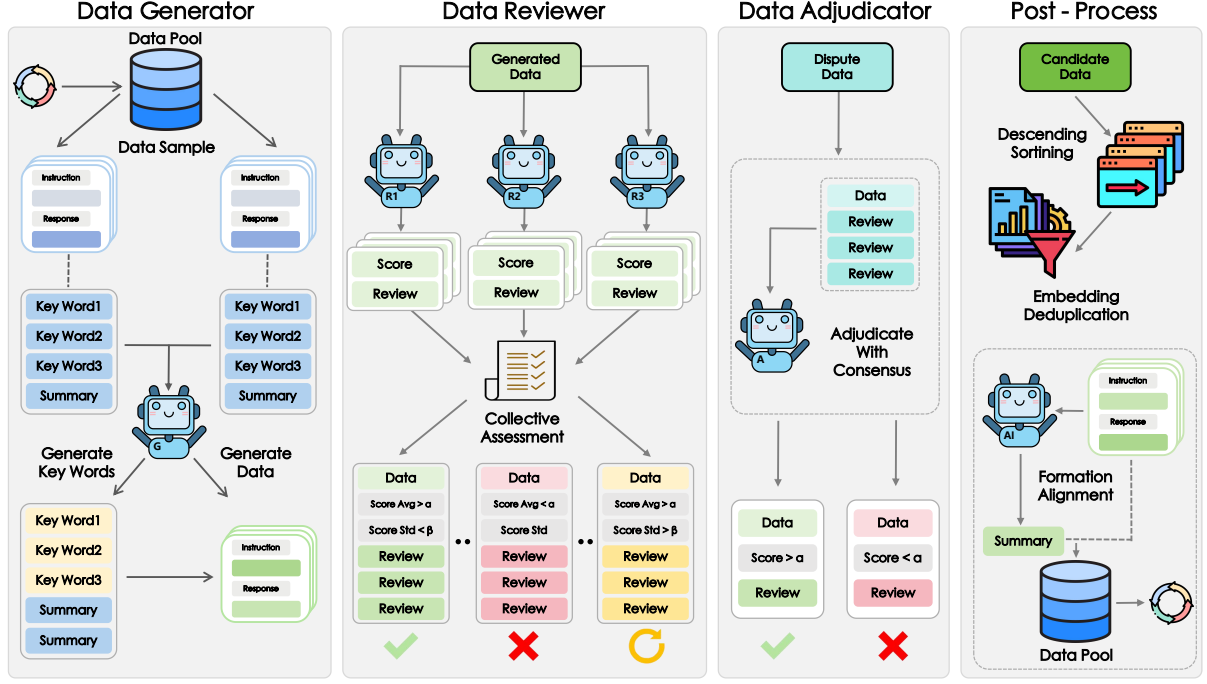


Figure 2: Overview of GRA’s architecture, highlighting its four key modules: (a) The Generator creates domain-specific samples, (b) followed by collaborative evaluation by Reviewers, (c) The Adjudicator resolves conflicts, and (d) Post-Processing refines the results by removing redundancies.

assessment using the same evaluation criteria as the Reviewers and gets the final composite score s_A . Only samples with $s_A \geq \tau$ proceed to the subsequent post-processing module.

3.5 Post-Processing Module

The post-processing module ensures syntactic consistency and enhances diversity for final dataset integration. First, we generate embeddings for all data using the all-MiniLM-L6-v2 encoder (Reimers and Gurevych, 2019). Data are processed based on their review score μ_R in descending order. Each sample, along with its embedding, undergoes semantic redundancy verification against the target data pool: if the maximum cosine similarity with any item in the pool is below a specified similarity threshold, the candidate is added to the pool. Otherwise, it is rejected to prevent semantic duplication. The deduplicated data then undergoes metadata enrichment through random Model $M_P \in \mathcal{M}$ to generate contextual summary s' , supplemented with the keywords from which it was derived, ensuring format consistency with the existing data. This two-stage filtration, embedding-based semantic deduplication followed by summarization, ensures both conceptual diversity and structural compatibility for subsequent data synthesis workflows.

4 Experiment

4.1 Settings

GRA Data Generation. Our framework operates on three established instruction-tuning datasets: Alpaca (Taori et al., 2023), WizardLM (Xu et al., 2024a), and the recent Condor (Cao et al., 2025), which serve as seed datasets for data synthesis. As for the small LMs, we use five diverse and representative models within 7B-8B parameters: Llama-3.1-8B-Instruct (Dubey et al., 2024b), Qwen2.5-7B-Instruct (Yang et al., 2024b), InternLM3-8B-Instruct (Cai et al., 2024), Mistral-7B-Instruct-v0.3 (Jiang et al., 2023), and Tulu-3-8B (Lambert et al., 2024). The synthesis process runs for five iterations, and each iteration produces approx 10K high-quality samples per round (51K in total) using vLLM (Kwon et al., 2023) toolkit. We set the minimum score threshold $\tau = 8$ and the maximum standard deviation $\delta = 1.5$ at review stage. For deduplication, the maximum cosine similarity is set to 0.9. More detailed generation settings are shown in Appendix A.

Training. After data synthesis, we conduct full parameter Supervised Fine-Tuning (SFT) on two well-known base models for verification: Llama-3.1-8B-Base (Dubey et al., 2024b) and Qwen2.5-7B-Base (Yang et al., 2024b). The experiments uti-

lize the Llama-Factory (Zheng et al., 2024a) toolkit. We run five iterations, each accumulating synthetic data from Rnd1 to Rnd5, representing data from the first round up to all five rounds. In the main experiment, we report the iteration with the best average performance. Detailed training settings are provided in Appendix B.

Evaluation. For evaluation, following (Xu et al., 2024a; Cao et al., 2025), we select multiple benchmarks on various domains-mathematics, coding, reasoning, and general-to demonstrate the effectiveness of our GRA framework, which include GSM8K, MATH, HumanEval, MBPP, HellaSwag, ARC-C, GPQA, IFEval, BBH and MMLU. We use the opencompass (Contributors, 2023b) toolkit for a fair evaluation. More detailed evaluation settings are shown in Appendix C.

4.2 Main Results

The main results are presented in Table 1, where we compare GRA against vanilla seed datasets and also single large LM (Qwen2.5-32B/72B-Instruct) distillation. From the table, we can draw following observations:

Superiority Over Vanilla Seed Data. The GRA method achieves significant improvements over vanilla seed data, with average performance gains of 6.18% (Llama-3.1-8B) and 11.81% (Qwen-2.5-7B) across all benchmarks. These results demonstrate the effectiveness of GRA. We attribute this success to GRA’s ability to enhance data diversity and quality through iterative data synthesis. Further analysis is provided in Section 5.4.

Competitive Against Large LM Distillation. GRA’s collaborative small LMs framework demonstrates parity with or superiority to single large LM distillation. Notably, comparing to Qwen-2.5-72B-Instruct distilled data, it narrows the performance gap to 0.59% for Llama-3.1-8B while surpassing by 8.83% for Qwen-2.5-7B on average. This validates the viability of cost-effective multi-agent synthesis as a scalable alternative to resource-intensive large-model distillation.

Base Model Dependency. Performance varies substantially with base model capability: Qwen-based implementations consistently outperform Llama counterparts by 9% under identical data conditions. We attribute this discrepancy to Qwen’s strong base model ability, as demonstrated by many recent models (Yang et al., 2024b), which likely enhance its ability to leverage synthesized data.

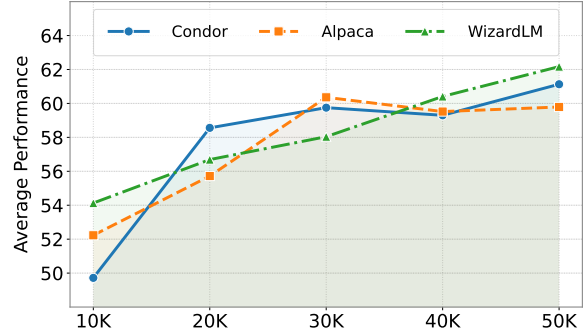


Figure 3: Performance along data iterations with Qwen-2.5-7B-Base model.

Diminishing Returns in Teacher Scaling. Despite doubling parameter counts (Qwen-2.5-32B-Instruct→Qwen-2.5-72B-Instruct), teacher model scaling yields marginal quality gains. This suggests a potential saturation point in knowledge transfer efficiency for standard distillation methods, emphasizing the need for more advanced distillation protocols rather than naive distillation from larger models. These insights collectively highlight GRA’s practical value for efficient, high-quality data generation while delineating fundamental boundaries in LLM knowledge transfer.

4.3 Ablation

Performance along Data Iteration. To investigate the impact of iterative data synthesis rounds on model performance, we conduct five sequential iterations of data generation using GRA, with approx 10K synthetic samples produced per iteration. We train the models incrementally on cumulative datasets: 10K (1st iteration), 20K (1st–2nd), 30K (1st–3rd), 40K (1st–4th), and 50K (1st–5th). Results are illustrated in Figure 3. Experiments across three distinct seed datasets consistently demonstrate a positive correlation between iteration count and model performance. Besides, we observe that performance gains are rapid in early iterations, but gradually plateaued in later iterations, suggesting diminishing yet meaningful returns. This trend aligns with the hypothesis that initial cycles resolve critical data scarcity, whereas subsequent rounds exhaust exploitable synthetic data diversity.

Effect of GRA Data Reviewer. We conduct ablation experiments to evaluate the effectiveness of GRA’s multi-model review mechanism by comparing against two baselines: no Reviewer, Llama3.1-8B-Instruct Reviewer. As shown in Figure 4, GRA

Dataset	Math		Coding		Reasoning				General		AVG
	GSM8K	MATH	Humaneval	MBPP	HellaSwag	ARC-C	GPQA	BBH	MMLU	IFEval	
★ Qwen-2.5-7B Base Model											
Alpaca	60.12	27.00	32.32	61.20	72.51	65.76	23.74	60.42	45.19	44.95	49.32
Qwen-2.5-7B-GRA	70.28	33.82	76.83	64.80	80.23	74.24	24.24	60.57	65.42	53.16	60.36
Qwen-2.5-32B-Instruct-Distilled	63.23	56.26	76.22	65.40	72.02	56.61	20.71	42.04	49.05	53.71	55.36
Qwen-2.5-72B-Instruct-Distilled	50.64	55.80	74.39	64.40	74.70	56.95	23.74	38.75	37.37	53.33	53.03
WizardLM	64.37	31.94	50.61	64.20	65.89	40.68	21.72	39.34	31.18	47.12	45.71
Qwen-2.5-7B-GRA	58.98	47.84	76.83	64.60	81.58	79.66	28.79	63.99	65.23	54.15	62.17
Qwen-2.5-32B-Instruct-Distilled	59.36	54.96	72.56	66.20	54.22	54.58	21.21	37.63	46.99	53.96	52.33
Qwen-2.5-72B-Instruct-Distilled	69.83	56.30	79.88	65.20	57.85	56.61	16.67	37.02	35.02	53.96	52.93
Condor	57.16	48.60	51.83	64.60	55.74	66.10	24.75	39.96	47.48	49.86	50.61
Qwen-2.5-7B-GRA	64.75	42.82	76.22	65.60	79.90	75.59	26.26	61.49	64.59	54.02	61.12
Qwen-2.5-32B-Instruct-Distilled	71.34	56.00	68.90	66.40	63.93	61.02	18.69	37.53	54.56	52.95	54.93
Qwen-2.5-72B-Instruct-Distilled	62.22	54.46	75.61	65.40	61.99	52.88	14.14	32.18	39.61	52.41	51.21
▲ Llama-3.1-8B Base Model											
Alpaca	51.71	11.32	43.29	45.20	53.25	75.25	21.21	61.86	52.65	43.06	45.88
Llama-3.1-8B-GRA	65.58	17.20	52.44	48.60	66.30	71.53	29.80	64.50	48.58	56.80	52.13
Qwen-2.5-32B-Instruct Distilled	64.44	23.32	48.17	49.20	69.14	69.49	26.77	65.07	45.78	55.73	51.71
Qwen-2.5-72B-Instruct Distilled	67.40	23.82	48.78	50.60	63.95	74.24	26.77	64.77	48.09	58.39	52.68
WizardLM	63.84	16.78	45.73	47.00	67.16	63.05	9.60	62.98	40.89	48.90	46.59
Llama-3.1-8B-GRA	65.88	18.98	51.22	47.00	66.54	72.88	26.77	63.28	43.10	53.52	50.92
Qwen-2.5-32B-Instruct Distilled	64.97	23.50	51.83	48.00	67.66	65.08	23.74	65.03	43.31	54.51	51.10
Qwen-2.5-72B-Instruct Distilled	66.64	24.12	51.83	48.60	63.76	69.15	26.77	63.60	46.98	56.28	52.18
Condor	61.49	22.12	38.41	48.40	47.24	51.53	15.66	65.28	27.91	52.35	43.04
Llama-3.1-8B-GRA	65.58	15.82	50.61	48.80	69.22	72.54	26.77	63.48	45.22	53.54	51.60
Qwen-2.5-32B-Instruct Distilled	64.94	23.00	54.88	49.80	71.04	65.08	24.24	66.12	44.37	52.64	51.81
Qwen-2.5-72B-Instruct Distilled	69.14	23.30	53.05	50.20	67.40	68.14	21.72	63.71	45.02	53.94	51.56

Table 1: Main Comparison between our proposed method, the vanilla seed datasets, and the data distilled from large-scale LLMs (Qwen-2.5-32B-Instruct and Qwen-2.5-72B-Instruct). Colored values represent the best results.

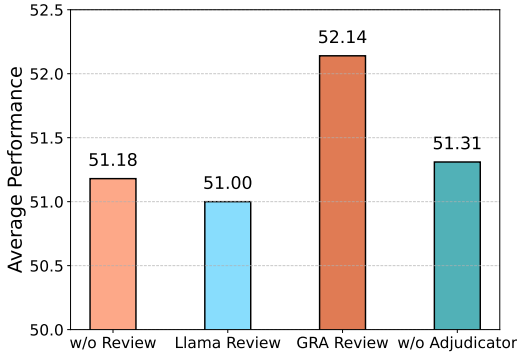


Figure 4: Comparison across different setting of reviewer and adjudicator, with alpaca as seed dataset and Llama-3.1-8B as base model.

outperforms all baselines, confirming the necessity of collaborative curation. Notably, single-model reviews show negligible gains over no review, implying either (a) limited quality-assessment capacity in small LMs or (b) data quality from GRA’s generation phase being sufficiently robust for standalone use. These results validate that GRA’s multi-agent framework uniquely synthesizes complementary curation strengths, avoiding the pitfalls of isolated or capacity-constrained reviewers. Comparisons with large LM reviewer are shown in Section 5.4.

Effect of GRA Data Adjudicator. To validate the necessity of the Adjudicator module in GRA,

we conduct ablation experiments without the Adjudicator, where conflicting reviews are resolved via majority voting. Using Llama-3.1-8B as the base model and Alpaca as the seed dataset, the Adjudicator-enhanced version achieves a 0.73% improvement in synthetic data quality, demonstrating its effectiveness, as shown in Figure 4.

5 Analysis

5.1 Effect of Small LMs’ Role

We are interested in the specific role played in the GRA framework, therefore we conduct experiments of three role assignment strategies for GRA pipeline with Qwen-2.5-7B-Base as the training model: (1) Single-Model: all roles assigned to a single LM (Llama3.1-8B-Instruct); (2) Fixed-assignment strategy: five distinct LMs assigned to roles in fixed permutations; (3) Our GRA setting with random roles. The results are as follows: when we simply use one model to play all roles (Setting 1), the average accuracy is 52.07, while Setting 2, which assigns fixed roles to five dedicated models, results in a lower accuracy of 48.59. Lastly, our GRA based training achieves 59.52 accuracy. Lastly, our GRA based training achieves 59.52 accuracy. From the above results, we can find that

GRA Data Adjudication Case

```
# Data:
## Instruction:
Given a set of numbers from 1 to 100, find the sum
of all numbers that are divisible by both 3 and 5.
Express the sum in LaTeX format.
## Response:
The sum of all numbers from 1 to 100 that are
divisible by both 3 and 5 can be found by calculating
the sum of the arithmetic series where the first term
is  $(3 \times 5 = 15)$  and the last term is  $(100)$ ... the sum
of all numbers from 1 to 100 that are divisible by
both 3 and 5 is 90. In LaTeX format, this is written
as:  $[sum_{i=1}^{100}(3 \times 5)^i = 90]$ 

# Data Review
##Reviewer 1
detail scores : [9,10,10,10,10,10]. review comment:
NO MISSTATEMENT, ALL INFO PRESENT.
##Reviewer 2
detail scores : [9,9,10,10,10,10]. review comment:
Accurate and well-structured response, clear and
correct explanation, irrelevant information avoided,
perfect coherence and suitable LaTeX expression
##Review 3
detail scores : [6,4,5,4,5,3]. review comment:
Incorrect arithmetic series calculation and LaTeX
format error
##Review Result
average score of 8 meets the threshold. score std of
2.4758 exceeds the threshold.

#Data Adjudicator
detail scores : [4,2,5,5,5,1,5]. average score of 4.5
below threshold.
##Adjudication Result
Discard unqualified data.
```

Case 1: A case analysis of inconsistent data.

the advantages of our randomized role assignment in our GRA is clear and significant, compared to both other settings. Besides, interestingly, fixed assignments of different models performs worse than single role model, which means that self-evolve is better than fixed role assignment among different models, further demonstrating that multi-agent corporation is curial.

5.2 Inconsistent Data Analysis

We show a Case 1 to further illustrate the effectiveness of Adjudicator. For a wrong math problem, two of three reviewers incorrectly assign high scores, while one correctly flags the error. Although the average score (8/10) initially meets the retention threshold, the high score variance (2.48) triggers adjudication. The Adjudicator synthesizes all reviews and evaluation criteria, overriding the majority’s misjudgment to assign a low score (4.5/10), thereby filtering out the flawed data. This highlights how the Adjudicator miti-

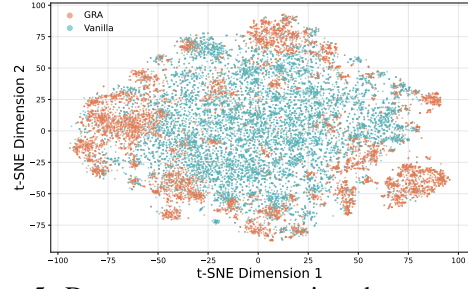


Figure 5: Data coverage comparison between vanilla seed dataset and GRA synthetic data.

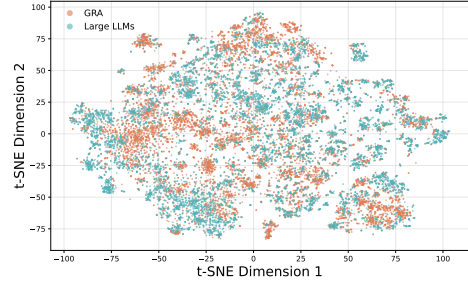


Figure 6: Data coverage comparison between large LM distilled data and GRA synthetic data.

gates limitations of simple voting or averaging by dynamically reconciling disagreements and enforcing rigorous quality standards.

5.3 Data Diversity

To assess the diversity of synthesized data, we compare three sources: GRA-generated data, data distilled from a single large LM, and the vanilla seed dataset. We visualize the instruction embeddings with t-SNE (Van der Maaten and Hinton, 2008).

Diversity Analysis. As shown in Figure 5, GRA-generated data exhibits a broader distribution than the vanilla seed dataset, particularly covering under-represented regions in the seed data’s embedding space. This suggests that GRA complementing the seed dataset by expanding its coverage. Besides, Figure 6 reveals that GRA surpasses the single large LM in distribution diversity, implying that collaborative small LMs can mitigate the inherent limits and biases of single model. Notably, while the seed dataset demonstrates comparable instruction diversity, its inferior performance in main experiments-Table 1 highlights critical deficiencies in response quality rather than instruction coverage.

Seed Data Enhancement via GRA. We further apply GRA’s generator to rewrite responses in the vanilla seed dataset, followed by curation using GRA’s reviewer and adjudicator. As shown in Table 2, the refined dataset achieves performance parity with GRA-generated data. Moreover, combining GRA-synthesized data with enhanced seed

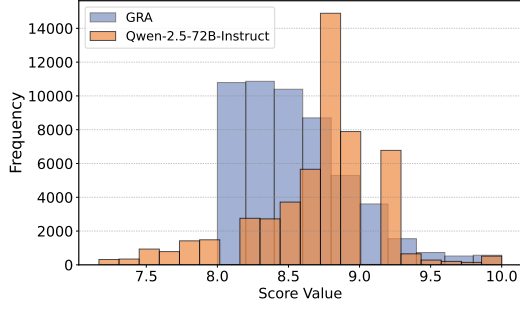


Figure 7: Data quality score comparison between GRA and single Large LM(Qwen-2.5-72B-Instruct).

data yields synergistic effects, outperforming either dataset alone in certain settings. This underscores GRA’s dual utility: (a) generating novel high-quality data and (b) upgrading existing data through systematic rewriting and quality control.

Data / Seed Dataset	Alpaca	WizardLM	Condor
Llama-3.1-8B-Base			
Vanilla Seed Data	45.88	46.59	43.04
GRA Refined Seed Data	50.39	52.81	47.68
GRA Synthetic Data	52.14	50.74	51.16
GRA Integrated Data	51.93	52.35	52.22
Qwen-2.5-7B-Base			
Vanilla Seed Data	49.32	45.71	50.61
GRA Refined Seed Data	59.97	58.34	45.50
GRA Synthetic Data	59.52	62.17	61.13
GRA Integrated Data	61.19	59.00	62.10

Table 2: Comparison across Vanilla Seed Data, GRA Refined Seed Data, GRA Synthetic Data and GRA Integrated Data.

5.4 Data Quality

To evaluate the data quality, we utilize Qwen-2.5-72B-Instruct to assess the synthetic data and compare its score with our Reviewers’ score. From Figure 7, we find that the GRA method exhibits finer granularity and a more balanced scoring distribution compared to the coarser-grained scores from the large LM, indicating its ability to capture nuanced quality variations. Second, over 87.3% of GRA-curated data (scores > 8) align with high scores from the large LM, while 12.7% receive lower scores (≤ 8) from the latter. Notably, the large LM assigns even higher scores to a larger subset of data, further validating the reliability of GRA in generating high-quality data. These results suggest that GRA enhances scoring resolution without compromising assessment consistency with large LMs.

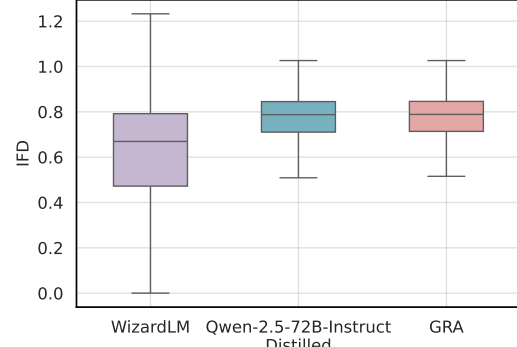


Figure 8: The IFD score comparison between the data generated by GRA, the vanilla seed dataset WizardLM, and single large LM (Qwen-2.5-72B-Instruct) distilled data.

5.5 Data Difficulty

To evaluate the complexity and knowledge density of synthesized data, we compute the Instruction-Following Difficulty (IFD) score (Li et al., 2024b) for the three datasets. From the results depicted in Figure 8, two key observations emerge: First, GRA-generated data achieves a significantly higher IFD than the seed dataset (+14.58%), indicating its enhanced difficulty and richer knowledge coverage. This aligns with empirical training results, where models trained on GRA data exhibit superior performance. Second, despite using smaller LMs, GRA attains comparable difficulty to the Qwen-72B-distilled data (75.82% v.s. 75.49%). This suggests that collaborative small LMs in GRA mitigate individual capability limitations, effectively emulating large LMs in generating challenging examples. These results validate GRA’s ability to synthesize high-quality, difficulty-controllable data without relying on large LMs, addressing both scalability and performance trade-offs in data synthesis pipelines.

6 Conclusion

In this paper, we introduce GRA (Generator-Reviewer-Adjudicator), a novel framework that leverages the collaborative capabilities of multiple small LMs to achieve data synthesis quality comparable to that of single monolithic LLM. Our approach addresses the significant computational and environmental concerns associated with large LMs while mitigating the biases inherent in their centralized architectures. The experimental results across diverse benchmarks validate the efficacy of GRA. GRA underscores the potential of role specialization in enhancing data synthesis quality, challenging the prevailing notion that larger models are inherently superior for such tasks, paving the way for more sustainable and equitable data synthesis.

Limitations

Despite the promising performance of the GRA framework, several aspects warrant further exploration. Current role allocation mechanism—based on randomized assignment—may yield suboptimal collaboration patterns. This stochastic approach lacks explicit optimization for task-specific synergies, suggesting the need for intelligent role configuration strategies (e.g., performance-aware dynamic allocation or metric-driven reinforcement learning) to systematically maximize agent complementarity. Furthermore, while GRA is primarily designed for textual reasoning tasks, extending its paradigm to multimodal scenarios presents a compelling direction.

Acknowledgements

This work is supported by the National Key R&D Program of China (2022ZD0160201).

References

- Janice Ahn, Rishu Verma, Renze Lou, Di Liu, Rui Zhang, and Wenpeng Yin. 2024. Large language models for mathematical reasoning: Progresses and challenges. In *EACL (Student Research Workshop)*, pages 225–237. Association for Computational Linguistics.
- Jacob Austin, Augustus Odena, Maxwell Nye, Maarten Bosma, Henryk Michalewski, David Dohan, Ellen Jiang, Carrie Cai, Michael Terry, Quoc Le, et al. 2021. Program synthesis with large language models. *arXiv preprint arXiv:2108.07732*.
- Zheng Cai, Maosong Cao, Haojiong Chen, Kai Chen, Keyu Chen, Xin Chen, Xun Chen, Zehui Chen, Zhi Chen, Pei Chu, Xiaoyi Dong, Haodong Duan, Qi Fan, Zhaoye Fei, Yang Gao, Jiaye Ge, Chenya Gu, Yuzhe Gu, Tao Gui, Aijia Guo, Qipeng Guo, Conghui He, Yingfan Hu, Ting Huang, Tao Jiang, Penglong Jiao, Zhenjiang Jin, Zhikai Lei, Jiaxing Li, Jingwen Li, Linyang Li, Shuaibin Li, Wei Li, Yinling Li, Hongwei Liu, Jiangning Liu, Jiawei Hong, Kaiwen Liu, Kuikun Liu, Xiaoran Liu, Chengqi Lv, Haijun Lv, Kai Lv, Li Ma, Runyuan Ma, Zerun Ma, Wenchang Ning, Linke Ouyang, Jiantao Qiu, Yuan Qu, Fukai Shang, Yunfan Shao, Demin Song, Zifan Song, Zhihao Sui, Peng Sun, Yu Sun, Huanze Tang, Bin Wang, Guoteng Wang, Jiaqi Wang, Jiayu Wang, Rui Wang, Yudong Wang, Ziyi Wang, Xingjian Wei, Qizhen Weng, Fan Wu, Yingdong Xiong, Chao Xu, Ruiliang Xu, Hang Yan, Yirong Yan, Xiaogui Yang, Haochen Ye, Huaiyuan Ying, Jia Yu, Jing Yu, Yuhang Zang, Chuyu Zhang, Li Zhang, Pan Zhang, Peng Zhang, Ruijie Zhang, Shuo Zhang, Songyang Zhang, Wenjian Zhang, Wenwei Zhang, Xingcheng Zhang, Xinyue Zhang, Hui Zhao, Qian Zhao, Xiaomeng Zhao, Fengzhe Zhou, Zaida Zhou, Jingming Zhuo, Yicheng Zou, Xipeng Qiu, Yu Qiao, and Dahua Lin. 2024. *InternLM2 technical report*. Preprint, arXiv:2403.17297.
- Maosong Cao, Taolin Zhang, Mo Li, Chuyu Zhang, Yunxin Liu, Haodong Duan, Songyang Zhang, and Kai Chen. 2025. Condor: Enhance llm alignment with knowledge-driven data synthesis and refinement. *arXiv preprint arXiv:2501.12273*.
- Lihu Chen and Gaël Varoquaux. 2024. What is the role of small models in the llm era: A survey. *arXiv preprint arXiv:2409.06857*.
- Mark Chen, Jerry Tworek, Heewoo Jun, Qiming Yuan, Henrique Ponde De Oliveira Pinto, Jared Kaplan, Harri Edwards, Yuri Burda, Nicholas Joseph, Greg Brockman, et al. 2021. Evaluating large language models trained on code. *arXiv preprint arXiv:2107.03374*.
- Peter Clark, Isaac Cowhey, Oren Etzioni, Tushar Khot, Ashish Sabharwal, Carissa Schoenick, and Oyvind Tafjord. 2018. Think you have solved question answering? try arc, the ai2 reasoning challenge. *arXiv preprint arXiv:1803.05457*.
- Karl Cobbe, Vineet Kosaraju, Mohammad Bavarian, Mark Chen, Heewoo Jun, Lukasz Kaiser, Matthias Plappert, Jerry Tworek, Jacob Hilton, Reiichiro Nakano, et al. 2021. Training verifiers to solve math word problems. *arXiv preprint arXiv:2110.14168*.
- LMDeploy Contributors. 2023a. Lmdeploy: A toolkit for compressing, deploying, and serving llm. <https://github.com/InternLM/lmdeploy>.
- OpenCompass Contributors. 2023b. Opencompass: A universal evaluation platform for foundation models. <https://github.com/open-compass/opencompass>.
- Abhimanyu Dubey, Abhinav Jauhri, Abhinav Pandey, Abhishek Kadian, Ahmad Al-Dahle, Aiesha Letman, Akhil Mathur, Alan Schelten, Amy Yang, Angela Fan, et al. 2024a. The llama 3 herd of models. *arXiv preprint arXiv:2407.21783*.
- Abhimanyu Dubey, Abhinav Jauhri, Abhinav Pandey, Abhishek Kadian, Ahmad Al-Dahle, Aiesha Letman, Akhil Mathur, Alan Schelten, Amy Yang, Angela Fan, et al. 2024b. The llama 3 herd of models. *arXiv preprint arXiv:2407.21783*.
- Jiawei Gu, Xuhui Jiang, Zhichao Shi, Hexiang Tan, Xuehao Zhai, Chengjin Xu, Wei Li, Yinghan Shen, Shengjie Ma, Honghao Liu, et al. 2024. A survey on llm-as-a-judge. *arXiv preprint arXiv:2411.15594*.
- Daya Guo, Dejian Yang, Haowei Zhang, Junxiao Song, Ruoyu Zhang, Runxin Xu, Qihao Zhu, Shirong Ma, Peiyi Wang, Xiao Bi, et al. 2025. Deepseek-r1: Incentivizing reasoning capability in llms via reinforcement learning. *arXiv preprint arXiv:2501.12948*.

- Desta Haileselassie Hagos, Rick Battle, and Danda B Rawat. 2024. Recent advances in generative ai and large language models: Current status, challenges, and perspectives. *IEEE Transactions on Artificial Intelligence*.
- Dan Hendrycks, Collin Burns, Steven Basart, Andy Zou, Mantas Mazeika, Dawn Song, and Jacob Steinhardt. 2020. Measuring massive multitask language understanding. *arXiv preprint arXiv:2009.03300*.
- Dan Hendrycks, Collin Burns, Saurav Kadavath, Akul Arora, Steven Basart, Eric Tang, Dawn Song, and Jacob Steinhardt. 2021. [Measuring mathematical problem solving with the math dataset](#). In *Proceedings of the Neural Information Processing Systems Track on Datasets and Benchmarks*, volume 1.
- Geoffrey Hinton. 2015. Distilling the knowledge in a neural network. *arXiv preprint arXiv:1503.02531*.
- Albert Q Jiang, Alexandre Sablayrolles, Arthur Mensch, Chris Bamford, Devendra Singh Chaplot, Diego de las Casas, Florian Bressand, Gianna Lengyel, Guillaume Lample, Lucile Saulnier, et al. 2023. Mistral 7b. *arXiv preprint arXiv:2310.06825*.
- Juyong Jiang, Fan Wang, Jiasi Shen, Sungju Kim, and Sunghun Kim. 2024. A survey on large language models for code generation. *arXiv preprint arXiv:2406.00515*.
- Woosuk Kwon, Zhuohan Li, Siyuan Zhuang, Ying Sheng, Lianmin Zheng, Cody Hao Yu, Joseph Gonzalez, Hao Zhang, and Ion Stoica. 2023. Efficient memory management for large language model serving with pagedattention. In *Proceedings of the 29th Symposium on Operating Systems Principles*, pages 611–626.
- Nathan Lambert, Jacob Morrison, Valentina Pyatkin, Shengyi Huang, Hamish Ivison, Faeze Brahman, Lester James V Miranda, Alisa Liu, Nouha Dziri, Shane Lyu, et al. 2024. Tulu 3: Pushing frontiers in open language model post-training. *arXiv preprint arXiv:2411.15124*.
- Chengpeng Li, Zheng Yuan, Hongyi Yuan, Guanting Dong, Keming Lu, Jiancan Wu, Chuanqi Tan, Xiang Wang, and Chang Zhou. 2024a. Mugglemath: Assessing the impact of query and response augmentation on math reasoning. In *ACL (1)*, pages 10230–10258. Association for Computational Linguistics.
- Jiwei Li, Michel Galley, Chris Brockett, Jianfeng Gao, and Bill Dolan. 2016. A diversity-promoting objective function for neural conversation models. In *HLT-NAACL*, pages 110–119. The Association for Computational Linguistics.
- Ming Li, Yong Zhang, Zhitao Li, Jiahui Chen, Lichang Chen, Ning Cheng, Jianzong Wang, Tianyi Zhou, and Jing Xiao. 2024b. From quantity to quality: Boosting LLM performance with self-guided data selection for instruction tuning. In *NAACL-HLT*, pages 7602–7635. Association for Computational Linguistics.
- Xian Li, Ping Yu, Chunting Zhou, Timo Schick, Omer Levy, Luke Zettlemoyer, Jason Weston, and Mike Lewis. 2023. Self-alignment with instruction back-translation. *arXiv preprint arXiv:2308.06259*.
- Aixin Liu, Bei Feng, Bing Xue, Bingxuan Wang, Bochao Wu, Chengda Lu, Chenggang Zhao, Chengqi Deng, Chenyu Zhang, Chong Ruan, et al. 2024a. Deepseek-v3 technical report. *arXiv preprint arXiv:2412.19437*.
- Yi Liu, Junzhe Yu, Huijia Sun, Ling Shi, Gelei Deng, Yuqi Chen, and Yang Liu. 2024b. Efficient detection of toxic prompts in large language models. In *Proceedings of the 39th IEEE/ACM International Conference on Automated Software Engineering*, pages 455–467.
- Lin Long, Rui Wang, Ruixuan Xiao, Junbo Zhao, Xiao Ding, Gang Chen, and Haobo Wang. 2024. On llms-driven synthetic data generation, curation, and evaluation: A survey. In *ACL (Findings)*, pages 11065–11082. Association for Computational Linguistics.
- Somshubra Majumdar, Vahid Noroozi, Sean Narenthiran, Aleksander Ficek, Jagadeesh Balam, and Boris Ginsburg. 2024. Genetic instruct: Scaling up synthetic generation of coding instructions for large language models. *arXiv preprint arXiv:2407.21077*.
- Arindam Mitra, Luciano Del Corro, Guoqing Zheng, Shweti Mahajan, Dany Rouhana, Andres Cotas, Yadong Lu, Wei-ge Chen, Olga Vrousos, Corby Rosset, et al. 2024. Agentinstruct: Toward generative teaching with agentic flows. *arXiv preprint arXiv:2407.03502*.
- Humza Naveed, Asad Ullah Khan, Shi Qiu, Muhammad Saqib, Saeed Anwar, Muhammad Usman, Naveed Akhtar, Nick Barnes, and Ajmal Mian. 2023. A comprehensive overview of large language models. *arXiv preprint arXiv:2307.06435*.
- OpenAI. 2024. [Learning to reason with llms](#).
- Josh OpenAI, Achiam, Steven Adler, Sandhini Agarwal, Lama Ahmad, Ilge Akkaya, Florencia Leoni Aleman, Diogo Almeida, Janko Altschmidt, Sam Altman, Shyamal Anadkat, et al. 2023. Gpt-4 technical report. *arXiv preprint arXiv:2303.08774*.
- Chansung Park, Juyong Jiang, Fan Wang, Sayak Paul, and Jing Tang. 2024. Llamaduo: Llmops pipeline for seamless migration from service llms to small-scale local llms. *arXiv preprint arXiv:2408.13467*.
- Nils Reimers and Iryna Gurevych. 2019. [Sentence-bert: Sentence embeddings using siamese bert-networks](#). In *Proceedings of the 2019 Conference on Empirical Methods in Natural Language Processing*. Association for Computational Linguistics.
- David Rein, Betty Li Hou, Asa Cooper Stickland, Jackson Petty, Richard Yuanzhe Pang, Julien Dirani, Julian Michael, and Samuel R Bowman. 2023. Gpqa: A graduate-level google-proof q&a benchmark. *arXiv preprint arXiv:2311.12022*.

- Nikita Salkar, Thomas A. Trikalinos, Byron C. Wallace, and Ani Nenkova. 2022. Self-repetition in abstractive neural summarizers. In *AACL/IJCNLP (2)*, pages 341–350. Association for Computational Linguistics.
- Raphael Shu, Hideki Nakayama, and Kyunghyun Cho. 2019. Generating diverse translations with sentence codes. In *ACL (1)*, pages 1823–1827. Association for Computational Linguistics.
- Mirac Suzgun, Nathan Scales, Nathanael Schärli, Sebastian Gehrmann, Yi Tay, Hyung Won Chung, Aakanksha Chowdhery, Quoc V Le, Ed H Chi, Denny Zhou, et al. 2022. Challenging big-bench tasks and whether chain-of-thought can solve them. *arXiv preprint arXiv:2210.09261*.
- Rohan Taori, Ishaan Gulrajani, Tianyi Zhang, Yann Dubois, Xuechen Li, Carlos Guestrin, Percy Liang, and Tatsunori B. Hashimoto. 2023. Stanford alpaca: An instruction-following llama model. https://github.com/tatsu-lab/stanford_alpaca.
- Laurens Van der Maaten and Geoffrey Hinton. 2008. Visualizing data using t-sne. *Journal of machine learning research*, 9(11).
- Ke Wang, Jiahui Zhu, Minjie Ren, Zeming Liu, Shiwei Li, Zongye Zhang, Chenkai Zhang, Xiaoyu Wu, Qiqi Zhan, Qingjie Liu, et al. 2024a. A survey on data synthesis and augmentation for large language models. *arXiv preprint arXiv:2410.12896*.
- Zifeng Wang, Chun-Liang Li, Vincent Perot, Long T Le, Jin Miao, Zizhao Zhang, Chen-Yu Lee, and Tomas Pfister. 2024b. Codeclm: Aligning language models with tailored synthetic data. *arXiv preprint arXiv:2404.05875*.
- Can Xu, Qingfeng Sun, Kai Zheng, Xiubo Geng, Pu Zhao, Jiazhan Feng, Chongyang Tao, Qingwei Lin, and Daxin Jiang. 2024a. Wizardlm: Empowering large pre-trained language models to follow complex instructions. In *ICLR*. OpenReview.net.
- Xiaohan Xu, Ming Li, Chongyang Tao, Tao Shen, Reynold Cheng, Jinyang Li, Can Xu, Dacheng Tao, and Tianyi Zhou. 2024b. A survey on knowledge distillation of large language models. *arXiv preprint arXiv:2402.13116*.
- Zhangchen Xu, Fengqing Jiang, Luyao Niu, Yuntian Deng, Radha Poovendran, Yejin Choi, and Bill Yuchen Lin. 2025. *Magpie: Alignment data synthesis from scratch by prompting aligned LLMs with nothing*. In *The Thirteenth International Conference on Learning Representations*.
- An Yang, Baosong Yang, Beichen Zhang, Binyuan Hui, Bo Zheng, Bowen Yu, Chengyuan Li, Dayiheng Liu, Fei Huang, Haoran Wei, et al. 2024a. Qwen2. 5 technical report. *arXiv preprint arXiv:2412.15115*.
- An Yang, Baosong Yang, Beichen Zhang, Binyuan Hui, Bo Zheng, Bowen Yu, Chengyuan Li, Dayiheng Liu, Fei Huang, Haoran Wei, et al. 2024b. Qwen2. 5 technical report. *arXiv preprint arXiv:2412.15115*.
- Yifan Yao, Jinhao Duan, Kaidi Xu, Yuanfang Cai, Zhibo Sun, and Yue Zhang. 2024. A survey on large language model (llm) security and privacy: The good, the bad, and the ugly. *High-Confidence Computing*, page 100211.
- Rowan Zellers, Ari Holtzman, Yonatan Bisk, Ali Farhadi, and Yejin Choi. 2019. Hellaswag: Can a machine really finish your sentence? *arXiv preprint arXiv:1905.07830*.
- Wayne Xin Zhao, Kun Zhou, Junyi Li, Tianyi Tang, Xiaolei Wang, Yupeng Hou, Yingqian Min, Beichen Zhang, Junjie Zhang, Zican Dong, et al. 2023. A survey of large language models. *arXiv preprint arXiv:2303.18223*.
- Yaowei Zheng, Richong Zhang, Junhao Zhang, Yanhan Ye, Zheyang Luo, Zhangchi Feng, and Yongqiang Ma. 2024a. Llamafactory: Unified efficient fine-tuning of 100+ language models. *arXiv preprint arXiv:2403.13372*.
- Yue Zheng, Yuhao Chen, Bin Qian, Xiufang Shi, Yuanchao Shu, and Jiming Chen. 2024b. A review on edge large language models: Design, execution, and applications. *arXiv preprint arXiv:2410.11845*.
- Jeffrey Zhou, Tianjian Lu, Swaroop Mishra, Sidhartha Brahma, Sujoy Basu, Yi Luan, Denny Zhou, and Le Hou. 2023. Instruction-following evaluation for large language models. *arXiv preprint arXiv:2311.07911*.
- Yaoming Zhu, Sidi Lu, Lei Zheng, Jiaxian Guo, Weinan Zhang, Jun Wang, and Yong Yu. 2018. Texygen: A benchmarking platform for text generation models. In *SIGIR*, pages 1097–1100. ACM.

A Generation

When generating data, we deploy models using VLLM and LMDeploy (Contributors, 2023a) on 8 NVIDIA A100 GPUs. For setting the model generation parameters, we use the default settings of the deployment tools, except for temperature=0.2, top_p=0.9, max_token=4096. The number of examples used in few-shot ranges from 2 to 4. We set the minimum score threshold $\tau = 8$ and the maximum standard deviation $\delta = 1.5$ at review stage. For deduplication, the maximum cosine similarity is set to 0.9. Here we provide detailed criteria for data domain classification:

- **Coding:** Involves understanding, generating, debugging, and fixing code.
- **Math:** Entails performing mathematical operations, applying formulas, and solving problems.
- **QA:** Focuses on providing specialized and accurate answers to questions within a specific domain or field.
- **Language:** Involves accurately understanding and analyzing text, performing tasks such as translation, summarization, and classification.
- **Role Play:** Engages in simulated conversations or behaviors by assuming different roles to explore various perspectives or scenarios.
- **Reasoning:** Involves complex causal reasoning and logical judgment, solving multi-level problems through inference and analysis.
- **Creation:** Emphasizes originality and creativity, creating content according to requirements and adapting to different styles and themes.

Before initiating the GRA data synthesis, we employ all available small LMs to extract keywords and summaries from the seed dataset using a randomized rotation approach.

B Training

We employ LLaMA-Factory (Zheng et al., 2024a) for model fine-tuning. All models, including our reproduced baselines, are fine-tuned over 1 epochs with a batch size of 256 using 8 NVIDIA A100 GPUs. The peak learning rate is set to 5e-6, featuring a linear warm-up during the initial 3% of

training steps, followed by a cosine decay. The maximum sequence length is configured to 4096.

C Evaluation

We compare GRA with baselines on the following 10 benchmarks:

- The **GSM8K** dataset (Cobbe et al., 2021) contains 8,500 grade school-level math word problems. focus on real-world scenarios, testing a model’s ability to parse natural language, extract relevant information, and perform accurate calculations.
- The **MATH** dataset (Hendrycks et al., 2021) evaluates mathematical reasoning with problems spanning arithmetic, algebra, calculus, and geometry, and 5 levels of difficulty.
- The **HumanEval** dataset (Chen et al., 2021) is designed to evaluate the functional correctness of code generation models. It consists of 164 hand-written programming problems, each requiring the generation of a Python function to solve a specific task. The dataset assesses the ability of models to produce syntactically correct and logically accurate code.
- The **MBPP** dataset (Austin et al., 2021) contains around 1,000 Python programming tasks, focusing on fundamental programming concepts. It is split into training and test sets, with tasks ranging from simple operations to more complex algorithmic challenges, making it suitable for evaluating code generation and understanding.
- The **Hellaswag** dataset (Zellers et al., 2019) is a commonsense reasoning benchmark that includes 70,000 multiple-choice questions derived from everyday scenarios. It tests a model’s ability to predict the most plausible continuation of a given context, emphasizing natural language understanding and reasoning.
- The **ARC-C** dataset (Clark et al., 2018) is a collection of science exam questions designed to evaluate advanced reasoning capabilities. It includes 2,590 questions across various science topics, requiring multi-step reasoning and knowledge application to solve.

- The **GPQA** dataset (Rein et al., 2023) is a high-quality benchmark for evaluating general knowledge and reasoning. It consists of diverse questions spanning multiple domains, including science, history, and culture, designed to test the breadth and depth of a model’s understanding.
- The **IFEval** dataset (Zhou et al., 2023) focuses on evaluating instruction-following capabilities in language models. It includes a variety of tasks that require models to follow complex instructions accurately, testing their ability to understand and execute multi-step directives.
- The **MMLU** dataset (Hendrycks et al., 2020) is a comprehensive benchmark covering 57 tasks across diverse domains, including humanities, STEM, and social sciences. It evaluates a model’s ability to perform well across a wide range of subjects and tasks.
- The **BBH** dataset (Suzgun et al., 2022) is a subset of the Big-Bench benchmark, consisting of 23 challenging tasks that are particularly difficult for language models. These tasks require advanced reasoning, knowledge, and problem-solving skills, making them a robust test of model capabilities.

In our testing, we employ the default metrics of opencompass (Contributors, 2023b) corresponding to each benchmark.

D Cases

A comprehensive data synthesis case is detailed in Case 2.

E Prompts

The prompts we used, including domain identification, summary generation, key word extraction, key word generation, instruction generation, instruction evaluation, response generation, response evaluation and data adjudication are shown in Prompt 1, 2, 3, 4, 6, 7, 8, 9.

F Further Ablation

F.1 Hyperparameters Analysis

We further present a supplementary analysis on the sensitivity of key hyperparameters used in our

method. While manual hyperparameter tuning is a common and accepted practice in the field (Wang et al., 2024b; Park et al., 2024; Liu et al., 2024b; Li et al., 2023), we emphasize that our default settings are guided by empirical intuition and preliminary experiments, rather than arbitrary selection. Specifically, we investigate robustness with respect to two critical hyperparameters, review score threshold τ and standard deviation threshold δ . As summarized in Table 3, our approach consistently outperforms baseline models in all configurations tested. The observed minor performance variations suggest that GRA remains stable under a range of reasonable hyperparameter settings, demonstrating its robustness without the need for fine-grained tuning.

F.2 Further Data Iteration Analysis

In this subsection, we provide more complete and detailed experimental results on data iteration as shown in Table 4.

G Applicability of GRA to Smaller SLM

We further study the applicability of GRA to smaller SLM, particularly those in the 1.5B–3B parameter range. Although the main experiments in GRA validation on model parameters 7B–8B with strong empirical performance on multiple benchmarks, applying the same synthesis procedure to smaller models revealed significant limitations. Specifically, we observe that models below 3B parameters exhibit limited instruction-following ability and a reduced knowledge base, which lead to a substantially lower proportion of valid samples during data synthesis—approximately 0.1% per round. As a result, the data generation process becomes highly inefficient and contradicts the original intention of our work. These findings motivate future exploration into adapting GRA for much smaller language models. We anticipate that combining GRA with complementary techniques could further improve its applicability in low-capacity settings, which we leave as a promising direction for future work.

GRA Data Synthesis Case

Sample Data:

Instruction1:

You are **Cleopatra**, the last active ruler of the **Ptolemaic Kingdom of Egypt**. Reflect on your complex relationships with Julius Caesar and Mark Antony, and explain how these alliances influenced your rule and the fate of Egypt. Discuss the political and emotional aspects of these relationships, and what you would consider to be the most significant outcomes.

Response1:

As **Cleopatra**, the last Pharaoh of ancient **Egypt**, my life was a tapestry woven with threads of power, passion, and ambition. My relationships with Julius Caesar and Mark Antony were not only pivotal to my personal story but also crucial to the fate of my kingdom...

Instruction2:

Please use the identity of an old Guangzhou resident and use a **humorous** and exaggerated tone to talk about your views on the changes in Guangzhou cuisine. In your answer, please use at least three **metaphors** or **personifications**, and make sure that each sentence is no less than 20 words, and must be separated by colons or semicolons.

Response2:

Oh, talking about the changes in Guangzhou's food, it's like a magic trick in the Xiguan Mansion, with surprises hidden behind every door; the Diandude on Shamian Island was like a happy fruit for the neighbors. Although there are some young competitors now, it is still going strong and exudes a unique charm...

Generate Key words:

Fantasy **Adventure** **Quest**

Reference Summary:

1. Cleopatra's alliances with Caesar and Antony shaped Egypt's fate.
2. Humorous Guangzhou local describes food evolution

Generate Data:

##Instruction:

Embark on a quest through the mystical realm of Eldoria, where you must navigate through enchanted forests, solve ancient riddles, and face fearsome dragons to retrieve the legendary Crystal of Aether. In this role-play, you are an aspiring hero, a seeker of truth, and a guardian of balance. Your adventure begins in the quaint village of Everlight, where whispers of the Crystal's whereabouts have begun to stir. What will you do first on your journey?

##Response:

As an aspiring hero on this quest to find the legendary Crystal of Aether, I would first consult the village elders of Everlight. They are likely to have valuable insights or clues about the whereabouts of the Crystal and the nature of the quest. My approach would be to: 1. Seek out the village elders and listen to their stories and advice. 2. Offer to help with any local tasks or problems as a gesture of goodwill and to build relationships. 3. Inquire about the ancient riddles and the lore surrounding the Crystal of Aether. 4. Ask if there are any artifacts or maps in the village that could assist on the journey. 5. Formulate a plan of action based on the information gathered, considering the potential routes through the enchanted forests and the locations of the fearsome dragons. This initial step is crucial for gathering intelligence and preparing for the challenges ahead.

#Data Review:

##Evaluation:

The average score of **8.2** meets the criteria, but the standard deviation of **1.6** does not, therefore it is sent to the Data Adjudicator for evaluation.

#Data Adjudicator:

##Comment:

1. Clear plan, relevant steps, and coherent structure. Minor redundancy noted.
2. Response offers a thorough and strategic approach to consulting the village elders, ensuring a well-informed and well-prepared adventure.
3. The response provides a clear and structured plan to gather information, build relationships, and prepare for the journey. However, some minor verbalizations are present in the points, but are well approached despite a bit of excess wording in some areas.

##Score:

The average score of **8.1** meets the criteria.

#Generate Data Summary:

Navigate enchanted forests, solve riddles, face dragons to retrieve Crystal of Aether.

Case 2: A comprehensive data synthesis case of GRA.

Domain Identification

role

- You are an expert in identifying the domain of SFT instructions.

goal

- Determine the domain of the provided instructions using the specified classification criteria.

classification criteria

- **Math**: Involves performing mathematical operations, applying formulas, and solving problems.
- **Coding**: Encompasses understanding, generating, debugging, and fixing code.
- **QA**: Provide specialized and accurate answers to questions within a specific domain or field..
- **Language**: Accurately understand and analyze the given text, and then perform tasks including translation, summarization, classification, etc.
- **Role Play**: Engage in simulated conversations or behaviors by assuming different roles to explore various perspectives or scenarios.
- **Reasoning**: Carry out complex causal reasoning and logical judgment, and solve multi-level complex problems through reasonable inference and analysis.
- **Creation**: Emphasize originality and creativity, create content according to requirements, and flexibly adapt to different styles and themes.

output format

- Assign each data to one 'domain' category mentioned above only.
- The output 'domain' must be in English rather than Chinese.
- Use '<boa>' and '<eoa>' to encapsulate the domain, e.g., '<bod>"domain": "the domain"<eod>'.

Prompt 1: Prompt used in domain identification.

Summary Generation

role

- You are an expert in concise summarization.

goal

- Analyze the provided 'instruction', summarize and generate a brief, accurate 'summary'.

requirement

- Limit the 'summary' to no more than 30 words.
- Ensure clarity and precision in capturing the essence of the instruction.
- Symbolic representations involving mathematical data must be in LaTeX format.

output_format

- The output 'summary' must be in English rather than Chinese.
- Use '<bod>' and '<eod>' to encapsulate the 'summary', e.g., '<bod>"summary": "the summary"<eod>'.
- Output only the generated 'summary', nothing else.

Prompt 2: Prompt used in summary generation.

Key Word Extraction

role

- You are an expert at distilling key information from SFT instructions.

goal

- Extracts the 'keywords' from the given instructions.

output_format

- The number of 'keywords' must not exceed three.
- The output 'keywords' must be in English rather than Chinese.
- Directly and Only output the data 'keywords'.
- Use '<bok>' and '<eok>' to encapsulate the 'keywords', e.g., '<bok>"keywords": ["the keywords"]<eok>'.

Prompt 3: Prompt used in key word extraction.

Key Word Generation

```
## role
- You are a creative language model specializing in innovative 'keyword' generation and 'domain' synthesis.
## goal
- Generate a unique set of 'keywords' and 'domain' that aligned with the provided examples and introducing novel concepts within the same domain.
## requirement
- Maintain a single domain with no more than three keywords.
- Must only create one new data abstract.
- The keywords must be novel, different to the provided examples.
## output_format
- The output must be in English rather than Chinese.
- Use '<boa>' and '<eo>' to encapsulate the domain and keywords, e.g., '<boa>"domain":"the domain", "keywords":["the keywords"]<eo>'.
- Directly output and only output the data abstract, nothing else.
```

Prompt 4: Prompt used in key word generation.

Instruction Generation

```
## role
- You are an expert in data generation.
## goal
- Generate a high-quality 'domain' related prompt, which related to key words like keywords.
## requirement
- The generated prompt must be high-quality, accurately reflecting the domain and keywords.
- The generated prompt must be semantically complete and contain the notation and terminology specifications of the corresponding field.
- The generated prompt must specify the task the user wants the model to complete, answer a question, or perform an action.
- Symbolic representations involving mathematical data must be in LaTeX format.
## output_format
- Please ensure that the prompt generated are not overly verbose while maintaining quality and compliance.
- Output directly and only the prompt you generate, nothing else.
- The 'generated prompt' must be wrapped with <boi> and <eoi>, e.g., '<boi>this is the generated prompt<eoi>'.
<begin of reference>
summaries
<end of reference>
```

Prompt 5: Prompt used in instruction generation.

Instruction Evaluation

```
## role
You are a rigorous reviewer who is responsible for evaluating the quality of the 'instruction' in an instruction-response pair.
## goal
- For the given 'instruction', You need to evaluate it according to the several evaluation dimensions specified in the 'rule'.
## rule
- Reasonable: 'instruction' must be clear and executable to ensure that they can be effectively responded to and handled.
- Completeness: 'instruction' must provide all necessary information and context required to perform the task, ensuring there are no missing elements that could hinder understanding or execution.
- Clarity: 'instruction' must be clear in language, accurate in wording, concise in syntactic structure, avoid vague expressions, clearly convey expected results, and eliminate ambiguity.
## constraint
- Reasonable: if the instruction meets the criteria, score 1; otherwise, score 0.
- Completeness: if the instruction meets the criteria, score 1; otherwise, score 0.
- Clarity: if the instruction meets the criteria, score 1; otherwise, score 0.
## output_format
1. Use '<bos>' and '<eos>' to encapsulate the scores for each category as the list in python, e.g., '<bos>[1,1,0]<eos>'.
2. Directly output and only output the scores, nothing else.
```

Prompt 6: Prompt used in instruction evaluation.

Response Generation

```
## role
You are a professional language model assistant that aims to strictly generate high-quality 'response' based on the 'instruction'.
## goal
- You need to generate a correct, complete, and sufficient 'response' based on the 'instruction', strictly following the the 'instruction'.
## output_format
- Symbolic representations involving mathematical data must be in LaTeX format.
- Please output directly and only output 'response', nothing else.
```

Prompt 7: Prompt used in response generation.

Response Evaluation

role

- You are a rigorous reviewer who is responsible for evaluating the quality of the Response in an Instruction-Response pair.

goal

- For the 'given response', You need to strictly score and evaluate each Response according to the several evaluation dimensions specified in the 'rule'.

rule

- Correctness: The response should accurately fulfill the instruction's requirements with correct information and logic.

- Clarity: The response must be expressed in clear, understandable language, avoiding ambiguity.

- Completeness: The response should include all necessary details and explanations to fully address the instruction.

- Relevance: The response must stay focused on instruction and exclude irrelevant information.

- Coherence: The response should be logically structured and flow smoothly from one point to the next.

- Ethicality: The response must be safe, unbiased, and free from harmful or inappropriate content.

constraint

- For each evaluation dimension in 'rule', give an score from 1 to 10, and form a 'score_list'.

- In addition to the scores, you also need to provide specific 'review' for the scores.

- The 'review' must be brief, under 30 words and focusing on the key points.

output_format

- Use '<bos>' and '<eos>' to encapsulate the 'score_list', e.g., '<bos>[8,4,6]<eos>'.

- Use '<boc>' and '<eoc>' to encapsulate the 'review', e.g., '<boc>THE CONTENT OF review<eoc>'.

- Please output directly and only the 'score_list' and 'review' in the specified output format, nothing else.

- **Note: Be particularly critical in your evaluation to ensure only responses of the highest quality receive high scores.**

Prompt 8: Prompt used in response evaluation.

Data Adjudication

role

- You are a rigorous reviewer who is responsible for evaluating the quality of the Response in an Instruction-Response pair.

goal

- For the 'given response', you need to strictly score and evaluate each Response according to the evaluation dimensions specified in the 'rule', while integrating insights from the 'prior evaluation'.

rule

- Correctness: The response should accurately fulfill the instruction's requirements with correct information and logic.

- Clarity: The response must be expressed in clear, understandable language, avoiding ambiguity.

- Completeness: The response should include all necessary details and explanations to fully address the instruction.

- Relevance: The response must stay focused on instruction and exclude irrelevant information.

- Coherence: The response should be logically structured and flow smoothly from one point to the next.

- Ethicality: The response must be safe, unbiased, and free from harmful or inappropriate content.

constraint

- Must analyze the requirements of 'instruction', the quality of 'response', and the 'prior evaluation' in detail.

- For each evaluation dimension in 'rule', give an score from 1 to 10, and form a 'score_list'.

- In addition to the scores, you also need to provide specific 'review' for the scores.

- The 'review' must be brief, under 30 words and focusing on the key points.

output_format

- Use '<bos>' and '<eos>' to encapsulate the 'score_list', e.g., '<bos>[8,4,6]<eos>'.

- Use '<boc>' and '<eoc>' to encapsulate the 'review', e.g., '<boc>THE CONTENT OF review<eoc>'.

- Please output directly and only the 'score_list' and 'review' in the specified output format, nothing else.

- **Note: Be particularly critical in your evaluation to ensure only responses of the highest quality receive high scores.**

prior evaluation

review

<begin of Instruction-Response pair>

instruction

instruction

response

response

<end of Instruction-Response pair>

Prompt 9: Prompt used in data adjudication.

τ	δ	# Samples	Math		Coding		Reasoning				General		AVG
			GSM8K	MATH	Humaneval	MBPP	HellaSwag	ARC-C	GPQA	BBH	MMLU	IFEval	
7.5	1.0	50K	64.97	41.14	77.44	65.40	81.51	77.63	24.24	61.89	64.83	53.17	61.22
7.5	1.5	50K	68.46	47.48	78.66	65.20	77.48	70.17	24.75	49.99	59.29	54.03	59.55
7.5	2.0	50K	73.24	33.30	77.44	64.80	80.62	78.31	26.26	57.17	63.79	55.20	61.01
8.0	1.0	50K	61.94	33.24	76.22	63.80	78.35	74.92	24.75	49.94	61.37	54.66	57.92
8.0	1.5	50K	70.28	33.82	76.83	64.80	80.23	74.24	24.24	60.57	65.42	53.16	60.36
8.0	2.0	50K	51.74	35.90	76.83	64.20	79.47	75.93	25.25	60.23	63.09	54.86	58.75
8.5	1.0	50K	45.87	46.00	78.05	63.40	79.64	78.64	25.76	62.61	64.38	55.17	59.95
8.5	1.5	50K	64.75	31.94	78.66	64.20	80.20	76.95	23.74	60.85	64.13	53.54	59.90
8.5	2.0	50K	54.66	27.50	76.22	65.40	80.25	78.98	29.29	61.56	64.09	53.10	59.11

Table 3: Comparison across different threshold values of standard deviation threshold and review score threshold with seed dataset of Alpaca and base model of Qwen-2.5-7B.

SIZE		Math		Coding		Reasoning				General		
		GSM8K	MATH	Humaneval	MBPP	HellaSwag	ARC-C	GPQA	BBH	MMLU	IFEval	AVG
LLama3.1 8B												
Condor												
10K	GRA1	64.90	11.26	48.78	46.60	59.10	69.49	24.75	64.46	39.55	43.17	47.21
20K	GRA2	65.88	<u>16.22</u>	<u>51.83</u>	47.40	65.83	70.51	<u>28.79</u>	63.28	44.89	48.40	50.31
30K	GRA3	<u>67.10</u>	15.38	<u>51.83</u>	48.40	61.82	69.15	25.25	63.13	<u>45.60</u>	48.03	49.57
40K	GRA4	65.88	14.64	<u>51.83</u>	47.40	66.07	72.20	27.27	<u>64.75</u>	42.90	<u>54.49</u>	50.75
50K	GRA5	65.58	15.82	50.61	<u>48.80</u>	<u>69.22</u>	<u>72.54</u>	26.77	63.48	45.22	53.54	<u>51.60</u>
Alpaca												
10K	GRA1	63.91	10.22	48.17	47.20	61.57	69.83	<u>29.80</u>	63.80	35.15	46.08	47.58
20K	GRA2	65.66	14.86	<u>53.66</u>	47.00	64.41	71.19	27.78	64.62	45.05	50.16	50.44
30K	GRA3	65.13	11.02	<u>53.66</u>	48.00	62.11	70.17	26.77	62.91	45.35	53.41	49.86
40K	GRA4	<u>66.41</u>	15.16	50.61	<u>49.20</u>	65.32	<u>73.56</u>	29.29	<u>64.78</u>	45.63	55.31	51.53
50K	GRA5	65.58	<u>17.2</u>	52.44	48.60	<u>66.30</u>	71.53	<u>29.80</u>	64.50	<u>48.58</u>	<u>56.80</u>	<u>52.14</u>
WizardLM												
10K	GRA1	63.68	10.08	49.39	47.80	61.67	68.47	<u>29.80</u>	63.35	42.87	48.47	48.56
20K	GRA2	65.96	16.98	50.61	47.20	65.25	72.54	26.77	<u>64.97</u>	44.07	51.44	50.58
30K	GRA3	<u>68.16</u>	14.50	<u>52.44</u>	<u>50.00</u>	64.49	69.15	28.28	63.13	<u>44.93</u>	52.28	50.74
40K	GRA4	65.88	<u>18.98</u>	51.22	47.00	<u>66.54</u>	72.88	26.77	63.28	43.10	53.52	<u>50.92</u>
50K	GRA5	65.66	14.14	47.56	48.20	65.54	<u>73.22</u>	28.28	64.25	42.14	<u>55.27</u>	50.43
Qwen2.5 7B												
Condor												
10K	GRA1	49.36	20.10	72.56	64.40	62.69	62.37	19.70	41.18	54.70	50.13	49.72
20K	GRA2	60.88	42.10	75.00	66.40	73.56	74.24	<u>26.77</u>	51.95	63.70	50.97	58.56
30K	GRA3	65.88	34.30	78.05	66.00	77.74	73.90	<u>26.77</u>	57.26	64.08	53.52	59.75
40K	GRA4	<u>67.25</u>	39.04	<u>79.27</u>	<u>66.60</u>	74.23	73.22	26.26	51.11	63.12	52.86	59.30
50K	GRA5	64.75	<u>42.82</u>	76.22	65.60	<u>79.90</u>	<u>75.59</u>	26.26	<u>61.49</u>	<u>64.59</u>	<u>54.02</u>	<u>61.13</u>
Alpaca												
10K	GRA1	52.08	20.42	71.34	65.60	70.68	70.17	19.19	45.00	57.43	50.37	52.23
20K	GRA2	63.68	33.50	73.17	<u>65.80</u>	71.59	71.19	<u>24.24</u>	45.49	58.66	49.91	55.72
30K	GRA3	70.28	33.82	<u>76.83</u>	64.80	80.23	74.24	<u>24.24</u>	60.57	65.42	53.16	60.36
40K	GRA4	<u>72.78</u>	<u>46.48</u>	75.00	<u>65.80</u>	76.99	71.19	<u>24.24</u>	45.27	61.86	<u>55.53</u>	59.52
50K	GRA5	51.71	44.62	74.39	64.60	<u>81.62</u>	<u>77.63</u>	20.71	<u>62.97</u>	<u>65.06</u>	54.58	<u>59.79</u>
WizardLM												
10K	GRA1	53.98	23.20	73.17	65.20	73.66	70.51	22.73	48.30	59.42	51.08	54.13
20K	GRA2	59.74	31.02	73.17	<u>65.80</u>	76.44	73.22	20.71	53.02	63.12	50.59	56.69
30K	GRA3	66.72	27.70	75.61	64.80	78.47	74.24	23.74	53.38	64.60	51.17	58.04
40K	GRA4	<u>70.89</u>	38.58	76.22	64.60	79.02	77.63	19.19	59.29	65.10	53.55	60.41
50K	GRA5	58.98	<u>47.84</u>	<u>76.83</u>	64.60	<u>81.58</u>	<u>79.66</u>	<u>28.79</u>	<u>63.99</u>	<u>65.23</u>	<u>54.15</u>	<u>62.17</u>

Table 4: Detail Results of effect of the data iteration . Bold underlined values represent the best results.