Large-Scale Diverse Synthesis for Mid-Training

Xuemiao Zhang[1,3*], **Chengying Tu**[2,3*], **Can Ren**[1,3*],
Rongxiang Weng[3†], **Hongfei Yan**[2,4†], **Jingang Wang**[3], **Xunliang Cai**[3]
[1] Peking University [2] School of Computer Science, Peking University [3] Meituan
[4] State Key Laboratory of Multimedia Information Processing, Peking University
{zhangxuemiao, yanhf}@pku.edu.cn {tuchengying, 2401210098}@stu.pku.edu.cn
wengrongxiang@gmail.com {wangjingang02, caixunliang}@meituan.com

Abstract

Mid-training has become critical for enhancing the knowledge and reasoning ability of large language models (LLMs), especially through the utilization of large-scale synthetic data. However, existing data synthesis methods often generate simplistic and homogeneous QA pairs, with limited scale and diversity. To address this, we propose BoostQA, a novel framework designed to synthesize large-scale, diverse, and high-quality QA data for mid-training. BoostQA introduces model probes during mid-training for the first time and implements STEM-focused multi-grade synthesis to boost data diversity as well as high-difficulty synthesis to alleviate difficulty degradation, followed by answer refinement to further improve quality. Extensive experiments by mid-training Llama-3 8B demonstrate that using only 20B-token BoostQA data achieves a significant average improvement of **12.74%** on MMLU and CMMLU over the pre-training baseline. After mid-training on 500B tokens, including 100B-token BoostQA data, our model achieves SOTA average results across benchmarks among mainstream models of comparable size. BoostQA also demonstrates robust scalability, with performance consistently improving as model size, data volume, and initial FLOPs scale.[1]

1 Introduction

Mid-training has emerged as a pivotal stage in the development of LLMs, strategically positioned between pre-training and post-training (Wang et al., 2025; Tu et al., 2025). During this intermediate stage, large-scale synthetic data is widely employed to enhance LLMs with improved token efficiency, especially in domains related to knowledge and reasoning (Dubey et al., 2024; OLMo et al., 2025; LongCat-Team et al., 2025a,b; GLM-5-Team et al., 2026). Previous studies have demonstrated that incorporating QA data into training corpora significantly improves model performance (Parmar et al., 2024; Maini et al., 2024; Chen et al., 2025; Wang et al., 2025), as such data is well-structured and knowledge-intensive. However, naturally collected QA data remains insufficient to meet training demands, making synthesis methods crucial and effective for QA data generation (Cheng et al., 2024; Jiang et al., 2025; Akter et al., 2025; Zhou et al., 2025; Su et al., 2025; Zhang et al., 2026). These methods leverage LLMs to rephrase or extract QA pairs from existing corpora, focusing primarily on the data format transformation, which makes synthetic QA data often limited in scale and diversity.

In this paper, we introduce BoostQA, a novel framework designed to synthesize *large-scale*, *diverse*, and *high-quality* QA data. The framework aims to solve three major challenges:

What types of data should be synthesized. Existing synthesis methods often generate simplistic and undifferentiated data that the model has already mastered. To address this, we design probe experiments to identify data types that are underrepresented in model development, which turn out to be STEM domains and high-difficulty tasks.

How to achieve data diversity while scaling up the volume. Current synthetic QA data tends to be homogeneous in content relative to the seed corpora and remains limited in scale. Drawing inspiration from the multi-grade teacher roles in education, we design multi-grade synthesis that expands each seed into multiple distinct QA samples. The seeds are reinterpreted by several virtual teacher roles, each offering unique perspectives and formulations. This one-to-many expansion greatly enhances diversity among QA samples derived from the same seed while scaling up the overall volume.

[*] Equal contribution.
[†] Corresponding author.
[1] The core code and dataset are available at https://github.com/ChasyT/BoostQA.

Findings of the Association for Computational Linguistics: ACL 2026, pages 16517–16539

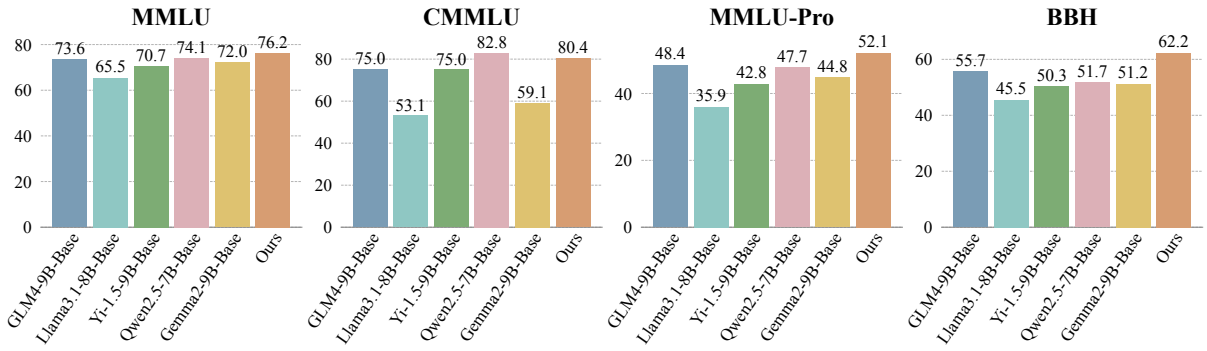


Figure 1: Comparison with mainstream models of comparable size across benchmarks.

How to overcome the issue of reduced and insufficient difficulty levels during synthesis. Our analysis reveals a reduction in data difficulty during multi-grade synthesis. To address this, we employ targeted upsampling of high-difficulty seeds and introduce a difficulty booster, which significantly enhances the proportion of high-difficulty samples.

By overcoming these challenges, BoostQA effectively generates large-scale, high-quality QA data with both diversity and enhanced difficulty. In summary, our work makes the following contributions: (1) We introduce model probes during mid-training *for the first time*, enabling systematic diagnosis of model weaknesses across knowledge domains and difficulty levels and the synthesis of underrepresented data. (2) We propose a novel BoostQA framework that synthesizes QA data and effectively boosts model performance during mid-training. (3) Extensive experiments by mid-training Llama-3 8B demonstrate that only 20B-token BoostQA greatly improves upon the pre-training baseline, with an average improvement of **12.74%** on MMLU and CMMLU. As shown in Figure 1, our model achieves SOTA average results across benchmarks among mainstream models of comparable size after being pre-trained on 10T tokens and mid-trained on 500B tokens—including 100B-token BoostQA.

2 Method

We propose BoostQA, a novel framework to synthesize a large-scale QA dataset that boosts model performance during mid-training. The framework is illustrated in Figure 2. Given a seed QA dataset, we (i) conduct a diagnostic probe experiment on pre-trained checkpoints using a held-out split to identify underperforming knowledge categories via per-label accuracy, (ii) design novel expansion-based multi-grade and high-difficulty synthesis, followed by answer refinement, to generate large-scale QA data with both diversity and enhanced difficulty, and (iii) mid-train the model on a controlled mixture of targeted synthetic data and high-quality pre-training corpora to boost overall capability.

2.1 Problem Statement

Let $\mathcal{M}_{\theta_0}$ denote a pre-trained LLM parameterized by θ_0. Due to inherent biases in the distribution of the pre-training corpora, $\mathcal{M}_{\theta_0}$ often exhibits knowledge deficiencies in specific categories. During mid-training, we assume the existence of an underlying ideal data distribution P_{oracle} that contains the necessary knowledge to address these deficiencies.

Given a seed dataset $\mathcal{S}$ and a predefined taxonomy $\mathcal{T}$, our objective is to synthesize a QA dataset $\mathcal{D}_{\text{syn}}$ that approximates the ideal distribution P_{oracle}, with particular emphasis on underperforming categories identified in $\mathcal{M}_{\theta_0}$. To inject specific knowledge while preserving general capabilities, we augment the original pre-training dataset by incrementally incorporating the synthetic QA dataset: $\mathcal{D}_{\text{train}} = \mathcal{D}_{\text{train}}^{\text{old}} \cup \mathcal{D}_{\text{syn}}$, where $\mathcal{D}_{\text{train}}^{\text{old}}$ denotes the original pre-training data and $\mathcal{D}_{\text{syn}}$ represents the synthetic QA dataset.

2.2 Model Probes

We curate the seed dataset $\mathcal{S}$ from heterogeneous sources to ensure content diversity, including QA pairs (Li et al., 2016; Amini et al., 2019), books (Gao et al., 2021), and high-quality web pages (Chang et al., 2024; Zhou et al., 2025). Details on the composition of the seed sources and the resulting synthetic QA data distribution are provided in Appendix A.1. From $\mathcal{S}$, we extract a QA subset to form a probe dataset $\mathcal{D}_{\text{probe}}$ following SliCK settings (Gekhman et al., 2024), which is used to identify underrepresented data categories through annotation and evalua-

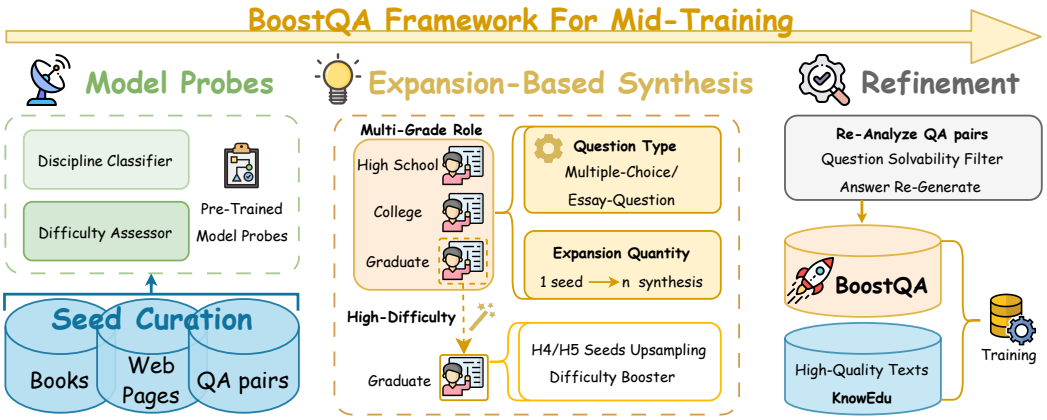
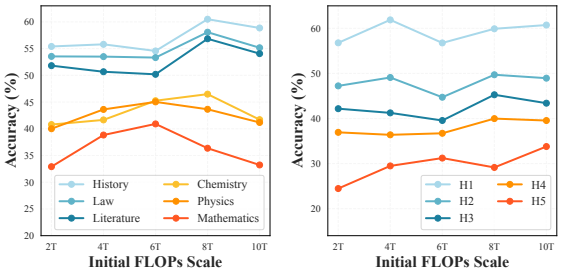


Figure 2: The BoostQA synthesis framework.

Pass Rate	Difficulty Level
(80%, 100%]	Basic (H1)
(50%, 80%]	Standard (H2)
(30%, 50%]	Advanced (H3)
(10%, 30%]	Challenge (H4)
[0%, 10%]	Extreme (H5)

Table 1: Definition of difficulty levels.



(a) Discipline probe results. (b) Difficulty probe results.

Figure 3: Discipline and difficulty probe experimental results, with six representative disciplines shown.

tion. Specifically, we evaluate model performance over 10 repeated trials, utilizing a distinct 5-shot prompt in greedy generation mode (temperature=0, top_p=1.0, max_tokens=48). Since greedy decoding is deterministic, variation across trials arises solely from the different few-shot examples, thereby isolating the effect of prompt selection on accuracy. The aggregate accuracy over all 10 trials is used to measure model mastery of the data.

To annotate each question $q \in \mathcal{D}_{\text{probe}}$, we design the taxonomy $\mathcal{T}$ for discipline classification and difficulty assessment. The discipline classifier $\mathcal{T}_d : \mathcal{Q} \to \mathcal{C}$ maps each question to one of 62 standardized discipline categories $\mathcal{C} = \{c_i\}_{i=1}^{62}$[2]. The difficulty assessor $\mathcal{T}_h : \mathcal{Q} \to \mathcal{H}$ assigns each question a difficulty level from $\mathcal{H} = \{h_i\}_{i=1}^{5}$. The difficulty levels are defined via the pass rate $P(q)$:

$$P(q) = \frac{\left|u \in \mathcal{U} : \text{Pass}(u, q)\right|}{|\mathcal{U}|}, \qquad (1)$$

where $\text{Pass}(u, q)$ indicates that student $u \in \mathcal{U}$ from QS Top 100 universities solves q correctly within one hour. The pass rate $P(q)$ is then discretized into five levels using thresholds specified in Table 1.

[2]GB/T 13745-2008 taxonomy, detailed in Appendix E.

The model is guided to simulate the pass rate $P(q)$ in order to annotate the difficulty level (see Appendix E). Thus, the taxonomy $\mathcal{T}$ assigns for each question q a pair of labels: $\mathcal{T}(q) = (\mathcal{T}_d(q), \mathcal{T}_h(q))$.

Building upon the annotated probe dataset $\mathcal{D}_{\text{probe}}$, which is further sampled approximately 1,000 items per difficulty level and 1,000 items per discipline, with the internal distribution within each stratum kept consistent, we conduct systematic probe experiments to evaluate the mastery of pre-trained models across different disciplines and difficulty levels. We measure answer accuracy on $\mathcal{D}_{\text{probe}}$ of various pre-training checkpoints, ranging from 2T tokens to 10T tokens. The results shown in Figure 3 reveal two key findings: (i) *STEM generally achieves lower accuracy than Humanities*; (ii) *Accuracy monotonically decreases as difficulty increases*. To steer the model toward more balanced and robust capabilities, we are motivated to synthesize data with an enhanced proportion of STEM and high-difficulty (H4/H5) samples.

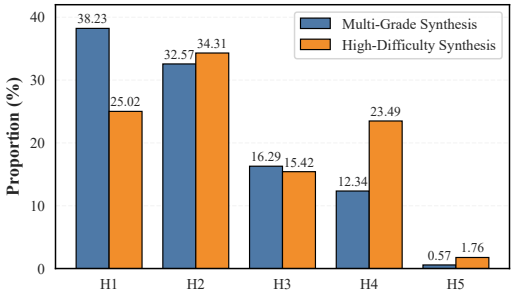


Figure 4: Difficulty distribution of multi-grade synthetic data and high-difficulty synthetic data.

2.3 Multi-Grade Synthesis

Current methods for synthesizing QA datasets often struggle to produce data with sufficient diversity and scale. Recent studies have explored multi-role prompting (Ge et al., 2025; Akter et al., 2025), but these are typically limited to one-to-one data creation or format transformation and lack large-scale synthesis attempts. Building on these insights, we design a *expansion-based* multi-grade synthesis module that leverages *educational-role* paradigms to systematically control the linguistic complexity and conceptual depth of generated questions. The prompt template $T_{\text{MG}}(r, t, n)$ is based on the educational role $r \in \{\text{high school, college, graduate}\}$, generative question type $t \in \{\text{multiple-choice, essay-question}\}$, and expansion quantity n (see Appendix F). The synthesis generator G_ϕ, parameterized by ϕ, then takes the seed sample s with STEM upsampling and the constructed prompt $T_{\text{MG}}(r, t, n)$ to synthesize a set of n diverse QA pairs that align with the specific role and question type. Formally,

$$\{(q_i, a_i)\}_{i=1}^{n} = G_\phi(s, T_{\text{MG}}(r, t, n)) \quad (2)$$

It supports one-to-many generation, allowing each seed to yield multiple question variants across different grade levels and enabling scalable targeted data expansion for underrepresented disciplines. As detailed in Appendix D, this one-to-many strategy significantly enhances diversity among QA samples generated from the same seed: compared with one-to-one generation approach, it achieves 53% higher intra-seed diversity and covers 42% more unique knowledge points.

2.4 High-Difficulty Synthesis

Synthetic questions often exhibit a systematic downward shift in difficulty, where the generated content corresponds to a lower level than the target educational stage (see Appendix B.1). To quantify this, we apply our trained difficulty assessor $\mathcal{T}_h$ to a split of multi-grade synthetic questions. The resulting difficulty distribution, shown in Figure 4, reveals a significant skew towards lower difficulty levels, indicating inherent limitations of standard role-driven synthesis in generating difficult content.

To address this, we integrate a *difficulty booster* B_d into the multi-grade synthesis, which *conditions the generator on high-difficulty constraints*, with role restricted to graduate ($r = r_{\text{grad}}$) and high-difficulty (H4/H5) seed upsampling. The prompt for high-difficulty synthesis is constructed as $T_{\text{HD}} = \text{Concat}(T_{\text{MG}}, B_{\text{d}})$, where $\text{Concat}(\cdot)$ denotes the string concatenation operator that appends the difficulty booster instruction B_{d} to the multi-grade prompt T_{MG} (see Appendix F). The synthesis process then follows:

$$\{(q_i, a_i)\}_{i=1}^{n} = G_\phi(s, T_{\text{HD}}) \quad (3)$$

This targeted intervention substantially alters the difficulty profile of the synthetic dataset. As quantified in Figure 4, the proportion of high-difficulty (H4/H5) questions increases by $1.96\times$, rising from 12.91% to 25.25% compared to multi-grade synthesis. This shift demonstrates the effectiveness of our framework in increasing the proportion of high-difficulty (H4/H5) synthetic data.

2.5 Answer Refinement

To ensure the quality of the raw QA dataset $\mathcal{D}_{\text{syn}}^{\text{raw}}$ generated by the synthesis modules, we employ a refinement model R_θ that performs both solvability verification and answer refinement. For each QA pair $(q, a) \in \mathcal{D}_{\text{syn}}^{\text{raw}}$, R_θ is prompted to first assess whether q is a well-posed, solvable question (see Appendix F). If deemed unsolvable, the pair is discarded. Otherwise, R_θ generates a refined answer a' through stepwise reasoning, correcting errors and improving completeness relative to the original answer a. Formally,

$$R_\theta(q, a) = \begin{cases} (q, a') & \text{if Solvable}(q) = \text{True} \\ \emptyset & \text{otherwise} \end{cases} \quad (4)$$

The final QA dataset, namely BoostQA, is obtained by applying R_θ to all raw pairs, with approximately 5% of generated questions identified as unsolvable and discarded:

$$\mathcal{D}_{\text{syn}} = \{R_\theta(q, a) \mid (q, a) \in \mathcal{D}_{\text{syn}}^{\text{raw}}\} \setminus \{\emptyset\}. \quad (5)$$

We further perform decontamination by employing exact 10-gram matching and embedding-based similarity filtering with a threshold of 0.6 to mitigate contamination from benchmark contents (Shao et al., 2024). Approximately 1% of the data in the synthetic dataset is filtered out during this process. The removed samples primarily consist of common entities (e.g., persons or events) that also appear in the benchmark examples, suggesting that the actual contamination risk is minimal and that most filtered cases are likely false positives captured by our intentionally conservative thresholds.

2.6 Training Data Construction

To ensure distributional alignment with pre-training corpora and mitigate catastrophic forgetting while maintaining data quality (Parmar et al., 2024), we construct *KnowEdu* via a dual-criteria selection, as a high-quality general dataset $\mathcal{D}_{\text{gen}}$ for constructing $\mathcal{D}_{\text{train}}$. The pre-training dataset undergoes rigorous quality stratification using the QuRater model (Wettig et al., 2024) to quantify knowledge density (Duan et al., 2025) and the educational classifier from FineWeb-Edu (Penedo et al., 2024) to assess educational utility. Only texts that score in the top 5% on both criteria simultaneously are retained.

3 Experiments

3.1 Experimental Setup

Training Details Based on the synthesis model analysis in Appendix C, we adopt DeepSeek-R1 (DeepSeek-AI et al., 2025) for synthesis. For answer refinement, we use DeepSeek-V3 (DeepSeek-AI et al., 2024).

We conduct mid-training experiments on Llama-3 8B (Dubey et al., 2024), which is initially pre-trained on 10T tokens from a dataset mirroring the composition of Matrix (Zhang et al., 2024). For the main experiments, we mid-train the 2T-token checkpoint on 40B tokens, comprising 20B-token KnowEdu (detailed in § 2.6) and 20B-token BoostQA in the naive format *{question}\n{answer}*. We implement mid-training using the Megatron framework (Narayanan et al., 2021) and utilize the Adam optimizer (Kingma and Ba, 2017) with a linearly decaying learning rate schedule initialized at 1.9×10^{-4} and terminating at 1.9×10^{-5} (see Appendix A.2). To further investigate the performance upper bound compared to mainstream models of comparable size, we conduct mid-training on the 10T-token checkpoint and scale the dataset up to 500B tokens, which includes 100B tokens from BoostQA.

We rigorously evaluate the scalability of BoostQA during mid-training from three perspectives: *model size*, *data volume*, and *initial FLOPs*. For model size, we assess the performance of Llama-3 with 1.7B, 8B, and 16B parameters (pre-trained on 4T, 10T, and 10T tokens, respectively) using identical 40B-token settings. For data volume, we scale the mid-training data up to 190B tokens based on the main experimental setting. For initial FLOPs, we additionally mid-train the 10T-token checkpoint with the same settings as the main experiments. All experiments preserve the learning rate decay relative to their pre-training checkpoints.

Evaluation We leverage 12 benchmarks to assess model capabilities across multiple dimensions. Knowledge-intensive evaluation includes MMLU (Hendrycks et al., 2021a), CMMLU (Li et al., 2024a), C-Eval (Huang et al., 2023), MMLU-Pro (Wang et al., 2024), and MMLU-STEM. Mathematical reasoning evaluation includes GSM8K (Cobbe et al., 2021) and MATH (Hendrycks et al., 2021b). Commonsense reasoning evaluation includes WinoGrande (Sakaguchi et al., 2021), HellaSwag (Zellers et al., 2019), and ARC-C (Clark et al., 2018). Complex reasoning evaluation includes BBH (Suzgun et al., 2023) and DROP (Dua et al., 2019).

Baselines We categorize the baselines into two paradigms, **each using the same 40B tokens for mid-training**. *The first paradigm evaluates general corpora*, including the pre-training dataset, used for subsequent mid-training; FineWeb-Edu (FW-Edu) (Penedo et al., 2024), a collection of high-quality educational texts curated from web crawling; KnowEdu, our curated high-quality knowledge-rich and educational data; and Nemotron-CC (N-CC) (Su et al., 2025), which has a 9:1 ratio of documents to QA pairs. *The second paradigm assesses the QA blend* **using the same 1:1 mixing ratio between KnowEdu and the experimental QA datasets** (detailed in Appendix A.3). For general-domain QA baselines, we include N-CC QA, a QA subset of N-CC, and YulanQA, a QA subset collected from the continual pre-training dataset of Chen et al. (2025). For mathematics QA baselines, we consider Nemotron-MIND (N-MIND) (Akter et al., 2025), a synthetic dataset of math dialogues; MegaMathQA (MMQA), a QA subset from MegaMath-

Dataset	MMLU	CMMLU	C-Eval	M-Pro	STEM	GSM8K	MATH	W.G.	H.S.	ARC-C	BBH	DROP	AVG.
Pre-train	55.08	52.23	57.11	24.32	45.17	33.95	6.50	51.50	43.00	70.50	35.79	42.31	43.12
FW-Edu	56.23	58.88	56.80	25.46	47.78	31.49	2.50	53.50	35.00	69.60	34.38	39.44	42.59
KnowEdu	58.17	62.99	61.98	25.64	49.16	32.56	8.00	54.50	36.00	71.50	35.12	41.07	44.72
N-CC	57.74	54.92	54.63	26.93	48.49	27.79	3.00	57.50	42.50	76.00	35.86	46.13	44.29
N-CC QA	58.62	62.02	59.02	27.68	49.25	29.33	7.00	55.00	41.50	73.00	34.86	41.63	44.91
YulanQA	56.15	58.95	57.53	24.89	47.09	30.48	9.00	53.00	43.00	73.00	35.75	42.18	44.25
N-MIND	58.48	61.11	60.98	30.32	51.55	47.42	13.50	52.00	40.50	73.00	37.69	46.33	47.74
MMQA	55.98	59.04	58.91	26.86	48.59	44.65	6.50	55.50	41.50	68.50	35.64	43.31	45.42
JZ	56.55	60.30	59.52	27.43	48.68	**56.27**	**23.00**	55.00	36.50	71.50	36.33	45.05	48.01
BoostQA	**64.78**	**68.00**	**66.65**	**30.61**	**57.71**	47.73	14.00	**60.50**	**44.00**	**80.00**	**38.52**	**47.41**	**51.66**

Table 2: Comparison across 12 benchmarks. The best and second best are in **bold** and underlined, respectively. Abbreviations: M-Pro = MMLU-Pro, W.G. = WinoGrande, H.S. = HellaSwag.

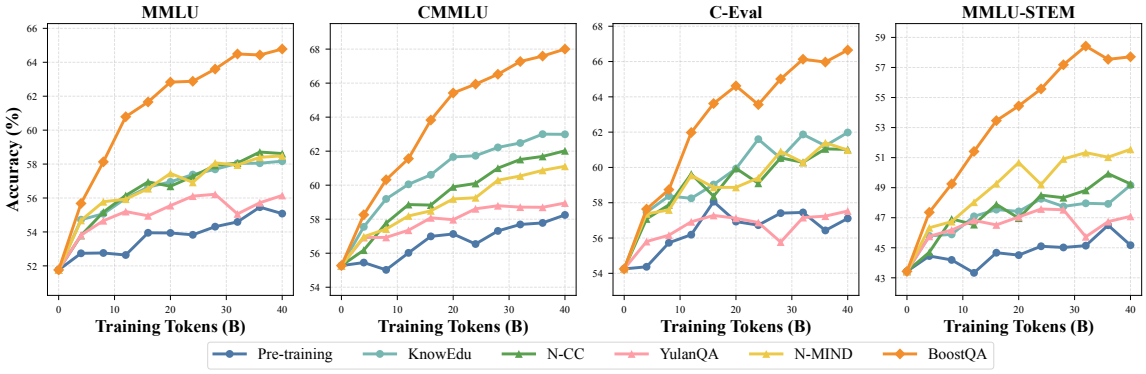


Figure 5: Performance dynamics of BoostQA and baselines as training progresses.

Synthetic (Zhou et al., 2025); and JiuZhang3.0 (JZ) (Zhou et al., 2024), which contains structured math QA with chain-of-thought (CoT).

3.2 Main Results

The main results are presented in Table 2, with results on MMLU subsets detailed in Appendix A.5.

BoostQA can significantly boost the performance of LLMs. Compared to the pre-training baseline, BoostQA demonstrates average improvements of **12.74%** on MMLU and CMMLU, as well as an average gain of 8.54% across 12 benchmarks. Even against high-quality corpora, BoostQA offers average improvements of 9.07% over FW-Edu and 6.94% over KnowEdu. Furthermore, as illustrated in Figure 1, our model, mid-trained on 500B tokens including 100B-token BoostQA, achieves SOTA average results across benchmarks among mainstream models of comparable size.

BoostQA outperforms QA counterparts. On MMLU and CMMLU, BoostQA achieves an average improvement of 6.07% over the strongest QA baseline, N-CC QA, and an improvement of 3.65% over JZ across 12 diverse benchmarks. In addition, on GSM8K and MATH, BoostQA$_{\text{MC}}$ variant with CoT performs better, detailed in § 3.4. These results demonstrate that BoostQA amplifies the advantages of the QA format through diverse and difficulty-enhanced synthesis.

BoostQA exhibits strong scalability and stability. As illustrated in Figure 5, BoostQA consistently maintains leading performance across benchmarks as training progresses. Notably, its upward trajectory continues even at the 40B-token threshold, indicating substantial unexploited potential. This training stability and continued improvement contrast sharply with other baselines, which exhibit convergence. This highlights BoostQA's effectiveness for sustainable capability growth through large-scale data expansion.

3.3 Scalability Analysis

Scalability results are shown in Figure 6.

At various model sizes, the performance gap between BoostQA and the baseline progressively widens as training advances. This suggests that BoostQA provides complementary learning signals

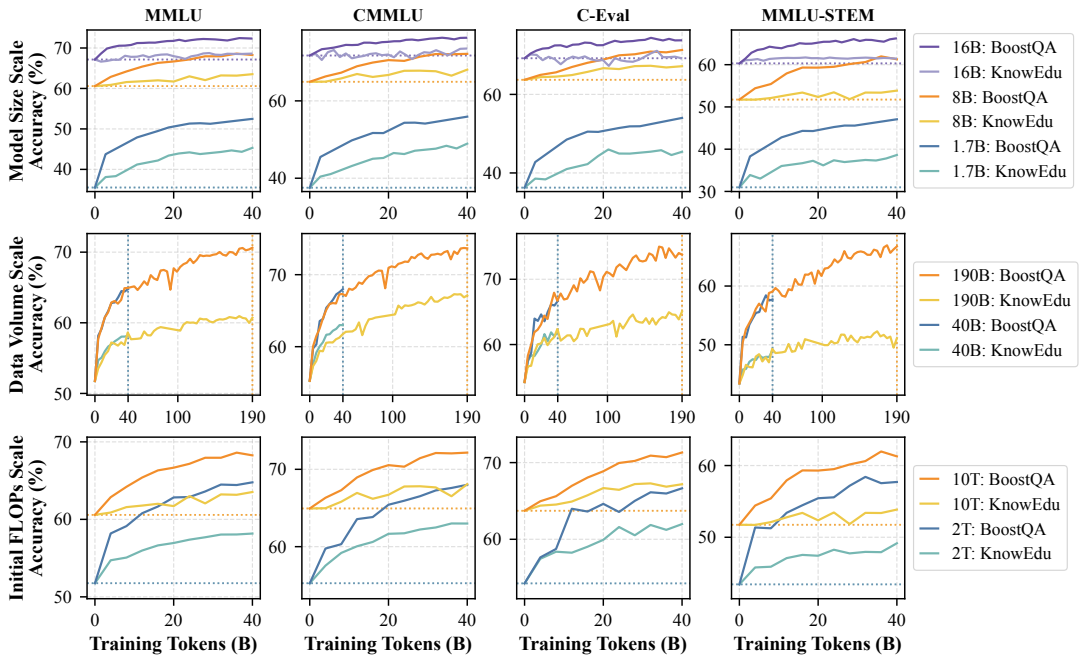


Figure 6: Multi-dimensional scalability of BoostQA, with values detailed in Appendix A.4.

that accumulate over time, rather than offering only temporary benefits. BoostQA demonstrates measurable performance improvements over KnowEdu for all model sizes. These enhancements are consistent across all evaluated benchmarks, with the magnitude of improvement positively correlating with increased exposure to training tokens.

At different data volumes, BoostQA maintains significant effectiveness when scaling from 40B to 190B tokens. The model's performance continues to rise at both the 40B and 190B tokens, demonstrating the strong scalability of BoostQA. Upon completing full-scale training, the model trained on 190B tokens of BoostQA surpasses the one trained on 40B tokens, achieving an average improvement of 5.76% on MMLU and CMMLU.

At different initial FLOPs, BoostQA can effectively improve performance. The initial FLOPs reflect the varying model capabilities at the start of mid-training. Introducing BoostQA at either the earlier 2T-token or the later 10T-token checkpoint yields stable performance gains compared to KnowEdu. Notably, on MMLU, BoostQA from the 2T-token checkpoint ultimately surpasses KnowEdu from the 10T-token checkpoint.

3.4 Mathematical Results

We conduct experiments in mathematics and compare the performance of the CoT variant, as shown in Table 3. We design three variants of BoostQA: BoostQA$_{\text{MM}}$, synthesized from high-quality web pages MegaMath-Web-Pro (MMWP) (Zhou et al., 2025); BoostQA$_{\text{Math}}$, synthesized from mathematical seeds; and BoostQA$_{\text{MC}}$, augmented from BoostQA$_{\text{Math}}$ with CoT reasoning in the *{question}\n{CoT}\n{answer}* format.

Dataset	CoT	G.	M.	Elem.	High.	Coll.	AVG.
Pre-train	-	33.95	6.50	36.50	30.50	25.00	26.49
FW-Edu	-	31.49	2.50	38.00	31.00	30.00	26.60
KnowEdu	-	32.56	8.00	37.50	27.50	31.00	27.31
MMWP	-	42.35	13.50	51.00	38.00	28.00	34.57
N-MIND	✔	47.42	13.50	43.50	29.50	37.00	34.18
MMQA	✔	44.65	6.50	47.00	31.50	35.00	32.93
JZ	✔	56.27	**23.00**	42.50	30.50	31.00	36.65
BoostQA	✗	47.73	14.00	53.00	44.50	38.00	39.45
BoostQA$_{\text{MM}}$	✗	42.26	13.50	**60.50**	**49.50**	**48.00**	42.75
BoostQA$_{\text{Math}}$	✗	42.96	13.50	57.00	46.50	44.00	40.79
BoostQA$_{\text{MC}}$	✔	**65.67**	22.50	54.50	46.00	43.00	**46.33**

Table 3: Comparison of mathematical performance. The best and second best are in **bold** and underlined, respectively. Abbreviations: G. = GSM8K, M. = MATH, (MMLU mathematical subsets) Elem. = elementary, High. = high school, Coll. = college.

BoostQA$_{\text{MC}}$ with CoT achieves SOTA average performance across mathematical benchmarks. BoostQA$_{\text{MC}}$ delivers peak performance on GSM8K while achieving near-parity to JZ with CoT on

	Dataset	MMLU	DROP	AVG.
Base	BoostQA	64.78	47.41	51.66
Refine	- w/o Refine	63.83	45.89	49.76
Question Type	- Multiple-Choice	64.20	46.32	51.26
	- Essay-Question	63.28	49.33	50.38
Synthesis Type	- Multi-Grade	65.15	46.42	50.38
	- High-Difficulty	63.37	47.61	49.76
General Corpora	- w/ N-CC Doc	61.81	43.08	48.06
	- w/ FW-Edu	63.68	45.91	49.91

Table 4: Ablation results showing the performance on MMLU, DROP, and the average across 12 benchmarks.

MATH. BoostQA$_{MM}$ without CoT attains optimal accuracy across all mathematical subsets of MMLU. These findings highlight the quality of seeds and the domain adaptability of BoostQA.

The comparative advantage of BoostQA$_{MC}$ underscores the essential role of CoT. BoostQA$_{MC}$ outperforms other BoostQA variants without CoT on GSM8K and MATH. When compared to baselines with CoT, BoostQA$_{MC}$ not only maintains comparable performance on GSM8K and MATH, but also exhibits an advantage on the mathematical subsets of MMLU.

QA serves as a superior knowledge carrier, and BoostQA amplifies its utility effectively. MMWP, serving as the data source for MMQA and BoostQA$_{MM}$, demonstrates inferior performance in mathematical tasks, underscoring the superior effectiveness of QA over seed web pages. Furthermore, the enhanced performance of BoostQA$_{MM}$ compared to MMQA indicates that the BoostQA framework is more effective for synthesizing high-quality QA data. BoostQA extends beyond mere distillation by effectively co-occurring diverse knowledge densely in QA format, yielding data distribution of high quality.

3.5 Ablations

We conduct four ablation studies, the results of which are shown in Table 4.

Refinement enhances answer accuracy and improves model performance. Implementing refinement results in an average improvement of 1.90%. Our sampling indicates a 16.18% rate of inconsistent answers before and after refinement, highlighting its importance for data quality. We utilize DeepSeek-V3 for refinement, and model comparisons are available in Appendix B.2. All subsequent experiments use the refined outputs.

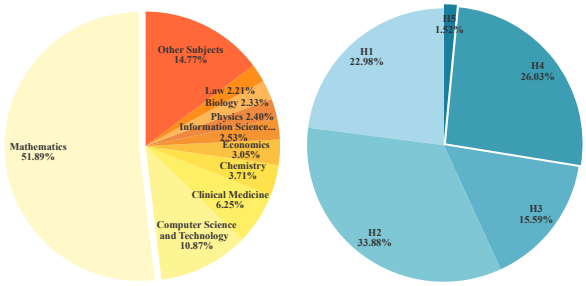
Integrating both multiple-choice and essay-question formats achieves optimal performance. The multiple-choice format offers advantages on benchmarks such as MMLU, possibly due to format alignment, while essay questions are vital for complex understanding tasks such as DROP. The combination of both formats seems to maximize effectiveness, with multiple-choice potentially facilitating efficient knowledge verification and essay-question enabling deep comprehension, which may lead to synergistic improvements.

The integration of multi-grade and high-difficulty synthesis yields strong performance. Multi-grade synthesis appears to optimize breadth and performs well on benchmarks that require extensive knowledge, such as MMLU, whereas high-difficulty synthesis seems to enhance depth for complex reasoning tasks, such as DROP. Their integration may help overcome individual limitations through complementary knowledge foundations and increased levels of difficulty.

BoostQA maintains robustness when blended with various general corpora. The base configuration of BoostQA, blended with KnowEdu, delivers top performance. As shown in Table 2 and Table 4, when BoostQA is blended with alternative corpora, BoostQA with N-CC Document outperforms the original N-CC by an average of 3.77%, and BoostQA with FineWeb-Edu surpasses the original FineWeb-Edu by 7.32%. This corpora-agnostic enhancement effect appears to support the view that BoostQA can be effective in overcoming structural and knowledge limitations across diverse corpora, potentially serving as a complementary data stream.

3.6 Analysis for BoostQA

We sample instances from BoostQA to investigate its distinct discipline and difficulty stratification patterns, as shown in Figure 7, with difficulty distribution comparison with N-CC QA detailed in Appendix B.3. Discipline sampling reveals a pronounced concentration in STEM domains, with mathematics constituting 51.89% of the total content. Computer Science and Technology, and Clinical Medicine follow at 10.87% and 6.25% respectively, collectively establishing STEM dominance. This discipline skew demonstrates intentional syn-



(a) Discipline distribution. (b) Difficulty distribution.

Figure 7: The distribution of BoostQA.

thesis prioritization rather than random distribution, strategically addressing capability gaps in complex analytical domains where traditional pre-training corpora exhibit insufficient coverage. Difficulty analysis demonstrates hierarchical stratification across the H1-H5 difficulty spectrum. Basic-level H1/H2 predominates at 56.86% prevalence, establishing foundational knowledge scaffolding essential for progressive learning trajectories. Intermediate tier H3 occupies 15.59%, forming a bridge between basic and extreme levels. The H4/H5 proportion increases to 27.55%, providing quantitative evidence of our high-difficulty synthesis module's capacity to amplify the high-difficulty proportion.

To evaluate the data quality of BoostQA, we randomly sample 100 QA pairs from each difficulty level and engage a professional annotator to assess the solvability of each question and the accuracy of its answer. A QA pair is labeled as correct only if the question is solvable and the provided answer is accurate. As shown in Table 5, the manual quality review indicates that most synthetic QA pairs are correct, with a strong correlation to difficulty levels.

Difficulty Level	Correct (%)
H1/H2	98
H3	94
H4/H5	87

Table 5: Data quality review of different difficulty levels for BoostQA.

4 Related Work

Synthetic data has been an important direction for improving the coverage and quality of LLM training datasets. At the pre-training stage, Li et al. (2023) created "textbook-like" corpora from annotated topics. Maini et al. (2024) leveraged an off-the-shelf instructed LLM to paraphrase pre-training documents into different styles, such as Wikipedia and QA formats. Nguyen et al. (2025) synthesized high-quality documents based on existing low-quality documents and CoT prompting. At the supervised fine-tuning (SFT) stage, Wang et al. (2023) bootstrapped new queries via few-shot prompting and distilled answers from proprietary LLMs. Li et al. (2024b) used existing web documents as answers and synthesized corresponding queries through back-translation. Xu et al. (2025) started from existing queries and evolved them into complex ones using various strategies. These approaches rely heavily on their initial seeds, limiting the difficulty level of the synthetic data.

Mid-training has recently emerged as an essential stage in the development of LLMs, distinguished from pre-training and SFT by its targeted approach to injecting specialized knowledge and enhancing critical, often underexposed abilities such as reasoning, mathematics, and coding. For instance, Yang et al. (2025); Ma et al. (2025) constructed knowledge graphs and sampled entity relations from them to synthesize relevant documents, though scalability was limited by the reliance on high-quality seed documents. Other approaches, such as Akter et al. (2025), generated mathematical dialogue datasets from math corpora using role-specific prompting. Similarly, Su et al. (2025); Zhou et al. (2025) extracted QA pairs from high-quality web pages and documents. The main shortcoming of these works is their lack of control over the knowledge diversity and the difficulty level of generated questions—a factor we demonstrate to be crucial for effective data synthesis.

5 Conclusion

In this paper, we propose BoostQA, a novel framework designed to synthesize large-scale, diverse, and high-quality QA data. We introduce model probes during mid-training for the first time. BoostQA incorporates diverse seed curation, STEM-focused multi-grade synthesis to boost data diversity, and high-difficulty synthesis to mitigate difficulty degradation, followed by answer refinement. Extensive experiments by mid-training Llama-3 8B validate BoostQA's 12.74% average improvement on MMLU and CMMLU. After being pre-trained on 10T tokens and mid-trained on 500B tokens, including 100B-token BoostQA, our model achieves SOTA average results among mainstream models of comparable size.

Limitations

This study designs probe experiments to identify data that are underrepresented in the current state of the model, thereby guiding subsequent data synthesis. As training progresses, the model may exhibit new states. The effectiveness of dynamic, multi-stage data synthesis remains to be explored. Furthermore, the BoostQA synthesis framework primarily focuses on two dimensions: discipline and difficulty level. In reality, more fine-grained dimensions, such as knowledge points, could be incorporated to provide more precise control over the distribution of synthetic data.

References

Syeda Nahida Akter, Shrimai Prabhumoye, John Kamalu, Sanjeev Satheesh, Eric Nyberg, Mostofa Patwary, Mohammad Shoeybi, and Bryan Catanzaro. 2025. MIND: Math informed synthetic dialogues for pretraining LLMs. In *The Thirteenth International Conference on Learning Representations*.

Aida Amini, Saadia Gabriel, Shanchuan Lin, Rik Koncel-Kedziorski, Yejin Choi, and Hannaneh Hajishirzi. 2019. MathQA: Towards interpretable math word problem solving with operation-based formalisms. In *Proceedings of the 2019 Conference of the North American Chapter of the Association for Computational Linguistics: Human Language Technologies, Volume 1 (Long and Short Papers)*, pages 2357–2367.

Yaoyao Chang, Lei Cui, Li Dong, Shaohan Huang, Yangyu Huang, Yupan Huang, Scarlett Li, Tengchao Lv, Shuming Ma, Qinzheng Sun, Wenhui Wang, Furu Wei, Ying Xin, Mao Yang, Qiufeng Yin, and Xingxing Zhang. 2024. Redstone: Curating general, code, math, and QA data for large language models. *CoRR*, abs/2412.03398.

Jie Chen, Zhipeng Chen, Jiapeng Wang, Kun Zhou, Yutao Zhu, Jinhao Jiang, Yingqian Min, Xin Zhao, Zhicheng Dou, Jiaxin Mao, Yankai Lin, Ruihua Song, Jun Xu, Xu Chen, Rui Yan, Zhewei Wei, Di Hu, Wenbing Huang, and Ji-Rong Wen. 2025. Towards effective and efficient continual pre-training of large language models. In *Proceedings of the 63rd Annual Meeting of the Association for Computational Linguistics (Volume 1: Long Papers)*, pages 5779–5795.

Daixuan Cheng, Yuxian Gu, Shaohan Huang, Junyu Bi, Minlie Huang, and Furu Wei. 2024. Instruction pre-training: Language models are supervised multitask learners. In *Proceedings of the 2024 Conference on Empirical Methods in Natural Language Processing*, pages 2529–2550.

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. *CoRR*, abs/1803.05457.

Karl Cobbe, Vineet Kosaraju, Mohammad Bavarian, Mark Chen, Heewoo Jun, Lukasz Kaiser, Matthias Plappert, Jerry Tworek, Jacob Hilton, Reiichiro Nakano, Christopher Hesse, and John Schulman. 2021. Training verifiers to solve math word problems. *CoRR*, abs/2110.14168.

DeepSeek-AI, Daya Guo, Dejian Yang, Haowei Zhang, Junxiao Song, Ruoyu Zhang, Runxin Xu, Qihao Zhu, Shirong Ma, Peiyi Wang, Xiao Bi, Xiaokang Zhang, Xingkai Yu, Yu Wu, Z. F. Wu, Zhibin Gou, Zhihong Shao, Zhuoshu Li, Ziyi Gao, and 81 others. 2025. Deepseek-r1: Incentivizing reasoning capability in llms via reinforcement learning. *CoRR*, abs/2501.12948.

DeepSeek-AI, Aixin Liu, Bei Feng, Bing Xue, Bingxuan Wang, Bochao Wu, Chengda Lu, Chenggang Zhao, Chengqi Deng, Chenyu Zhang, Chong Ruan, Damai Dai, Daya Guo, Dejian Yang, Deli Chen, Dongjie Ji, Erhang Li, Fangyun Lin, Fucong Dai, and 81 others. 2024. Deepseek-v3 technical report. *CoRR*, abs/2412.19437.

Dheeru Dua, Yizhong Wang, Pradeep Dasigi, Gabriel Stanovsky, Sameer Singh, and Matt Gardner. 2019. DROP: A reading comprehension benchmark requiring discrete reasoning over paragraphs. In *Proceedings of the 2019 Conference of the North American Chapter of the Association for Computational Linguistics: Human Language Technologies, Volume 1 (Long and Short Papers)*, pages 2368–2378.

Feiyu Duan, Xuemiao Zhang, Sirui Wang, Haoran Que, Yuqi Liu, Wenge Rong, and Xunliang Cai. 2025. Enhancing llms via high-knowledge data selection. In *Proceedings of the AAAI Conference on Artificial Intelligence*, volume 39, pages 23832–23840.

Abhimanyu Dubey, Abhinav Jauhri, Abhinav Pandey, Abhishek Kadian, Ahmad Al-Dahle, Aiesha Letman, Akhil Mathur, Alan Schelten, Amy Yang, Angela Fan, Anirudh Goyal, Anthony Hartshorn, Aobo Yang, Archi Mitra, Archie Sravankumar, Artem Korenev, Arthur Hinsvark, Arun Rao, Aston Zhang, and 82 others. 2024. The llama 3 herd of models. *CoRR*, abs/2407.21783.

Leo Gao, Stella Biderman, Sid Black, Laurence Golding, Travis Hoppe, Charles Foster, Jason Phang, Horace He, Anish Thite, Noa Nabeshima, Shawn Presser, and Connor Leahy. 2021. The pile: An 800gb dataset of diverse text for language modeling. *CoRR*, abs/2101.00027.

Tao Ge, Xin Chan, Xiaoyang Wang, Dian Yu, Haitao Mi, and Dong Yu. 2025. Scaling synthetic data creation with 1,000,000,000 personas. *Preprint*, arXiv:2406.20094.

Zorik Gekhman, Gal Yona, Roee Aharoni, Matan Eyal, Amir Feder, Roi Reichart, and Jonathan Herzig. 2024.

Does fine-tuning LLMs on new knowledge encourage hallucinations? In *Proceedings of the 2024 Conference on Empirical Methods in Natural Language Processing*, pages 7765–7784.

GLM-5-Team, :, Aohan Zeng, Xin Lv, Zhenyu Hou, Zhengxiao Du, Qinkai Zheng, and 1 others. 2026. Glm-5: from vibe coding to agentic engineering. *Preprint*, arXiv:2602.15763.

Dan Hendrycks, Collin Burns, Steven Basart, Andy Zou, Mantas Mazeika, Dawn Song, and Jacob Steinhardt. 2021a. Measuring massive multitask language understanding. In *International Conference on Learning Representations*.

Dan Hendrycks, Collin Burns, Saurav Kadavath, Akul Arora, Steven Basart, Eric Tang, Dawn Song, and Jacob Steinhardt. 2021b. Measuring mathematical problem solving with the MATH dataset. In *Thirty-fifth Conference on Neural Information Processing Systems Datasets and Benchmarks Track (Round 2)*.

Yuzhen Huang, Yuzhuo Bai, Zhihao Zhu, Junlei Zhang, Jinghan Zhang, Tangjun Su, Junteng Liu, Chuancheng Lv, Yikai Zhang, jiayi lei, Yao Fu, Maosong Sun, and Junxian He. 2023. C-eval: A multi-level multi-discipline chinese evaluation suite for foundation models. In *Thirty-seventh Conference on Neural Information Processing Systems Datasets and Benchmarks Track*.

Jinhao Jiang, Junyi Li, Xin Zhao, Yang Song, Tao Zhang, and Ji-Rong Wen. 2025. Mix-CPT: A domain adaptation framework via decoupling knowledge learning and format alignment. In *The Thirteenth International Conference on Learning Representations*.

Diederik P. Kingma and Jimmy Ba. 2017. Adam: A method for stochastic optimization. *Preprint*, arXiv:1412.6980.

Haonan Li, Yixuan Zhang, Fajri Koto, Yifei Yang, Hai Zhao, Yeyun Gong, Nan Duan, and Timothy Baldwin. 2024a. CMMLU: Measuring massive multitask language understanding in Chinese. In *Findings of the Association for Computational Linguistics: ACL 2024*, pages 11260–11285.

Peng Li, Wei Li, Zhengyan He, Xuguang Wang, Ying Cao, Jie Zhou, and Wei Xu. 2016. Dataset and neural recurrent sequence labeling model for open-domain factoid question answering. *Preprint*, arXiv:1607.06275.

Xian Li, Ping Yu, Chunting Zhou, Timo Schick, Omer Levy, Luke Zettlemoyer, Jason Weston, and Mike Lewis. 2024b. Self-alignment with instruction backtranslation. *Preprint*, arXiv:2308.06259.

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

LongCat-Team, Bayan, Bei Li, Bingye Lei, Bo Wang, and 1 others. 2025a. Longcat-flash technical report. *Preprint*, arXiv:2509.01322.

LongCat-Team, Anchun Gui, Bei Li, Bingyang Tao, Bole Zhou, Borun Chen, Chao Zhang, Chao Zhang, Chengcheng Han, Chenhui Yang, Chi Zhang, and 1 others. 2025b. Introducing longcat-flash-thinking: A technical report. *Preprint*, arXiv:2509.18883.

Shengjie Ma, Xuhui Jiang, Chengjin Xu, Cehao Yang, Liyu Zhang, and Jian Guo. 2025. Synthesize-on-graph: Knowledgeable synthetic data generation for continue pre-training of large language models. *Preprint*, arXiv:2505.00979.

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

Deepak Narayanan, Mohammad Shoeybi, Jared Casper, Patrick LeGresley, Mostofa Patwary, Vijay Korthikanti, Dmitri Vainbrand, Prethvi Kashinkunti, Julie Bernauer, Bryan Catanzaro, Amar Phanishayee, and Matei Zaharia. 2021. Efficient large-scale language model training on gpu clusters using megatron-lm. In *Proceedings of the International Conference for High Performance Computing, Networking, Storage and Analysis*, SC '21, New York, NY, USA. Association for Computing Machinery.

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

Jianmo Ni, Gustavo Hernandez Abrego, Noah Constant, Ji Ma, Keith Hall, Daniel Cer, and Yinfei Yang. 2022. Sentence-t5: Scalable sentence encoders from pretrained text-to-text models. In *Findings of the Association for Computational Linguistics: ACL 2022*, pages 1864–1874. Association for Computational Linguistics.

Team OLMo, Pete Walsh, Luca Soldaini, Dirk Groeneveld, Kyle Lo, Shane Arora, Akshita Bhagia, Yuling Gu, Shengyi Huang, Matt Jordan, Nathan Lambert, Dustin Schwenk, Oyvind Tafjord, Taira Anderson, David Atkinson, Faeze Brahman, Christopher Clark, Pradeep Dasigi, Nouha Dziri, and 21 others. 2025. 2 olmo 2 furious. *Preprint*, arXiv:2501.00656.

Jupinder Parmar, Sanjev Satheesh, Mostofa Patwary, Mohammad Shoeybi, and Bryan Catanzaro. 2024. Reuse, don't retrain: A recipe for continued pretraining of language models. *Preprint*, arXiv:2407.07263.

Guilherme Penedo, Hynek Kydlíček, Loubna Ben allal, Anton Lozhkov, Margaret Mitchell, Colin Raffel, Leandro Von Werra, and Thomas Wolf. 2024. The fineweb datasets: Decanting the web for the finest

text data at scale. In *The Thirty-eight Conference on Neural Information Processing Systems Datasets and Benchmarks Track*.

Keisuke Sakaguchi, Ronan Le Bras, Chandra Bhagavatula, and Yejin Choi. 2021. Winogrande: an adversarial winograd schema challenge at scale. *Commun. ACM*, 64(9):99–106.

Zhihong Shao, Peiyi Wang, Qihao Zhu, Runxin Xu, Junxiao Song, Xiao Bi, Haowei Zhang, Mingchuan Zhang, Y. K. Li, Y. Wu, and Daya Guo. 2024. Deepseekmath: Pushing the limits of mathematical reasoning in open language models. *Preprint*, arXiv:2402.03300.

Dan Su, Kezhi Kong, Ying Lin, Joseph Jennings, Brandon Norick, Markus Kliegl, Mostofa Patwary, Mohammad Shoeybi, and Bryan Catanzaro. 2025. Nemotron-CC: Transforming Common Crawl into a refined long-horizon pretraining dataset. In *Proceedings of the 63rd Annual Meeting of the Association for Computational Linguistics (Volume 1: Long Papers)*, pages 2459–2475, Vienna, Austria. Association for Computational Linguistics.

Mirac Suzgun, Nathan Scales, Nathanael Schärli, Sebastian Gehrmann, Yi Tay, Hyung Won Chung, Aakanksha Chowdhery, Quoc Le, Ed Chi, Denny Zhou, and Jason Wei. 2023. Challenging BIG-bench tasks and whether chain-of-thought can solve them. In *Findings of the Association for Computational Linguistics: ACL 2023*, pages 13003–13051.

Chengying Tu, Xuemiao Zhang, Rongxiang Weng, Rumei Li, Chen Zhang, Yang Bai, Hongfei Yan, Jingang Wang, and Xunliang Cai. 2025. A survey on llm mid-training. *Preprint*, arXiv:2510.23081.

Yizhong Wang, Yeganeh Kordi, Swaroop Mishra, Alisa Liu, Noah A. Smith, Daniel Khashabi, and Hannaneh Hajishirzi. 2023. Self-instruct: Aligning language models with self-generated instructions. *Preprint*, arXiv:2212.10560.

Yubo Wang, Xueguang Ma, Ge Zhang, Yuansheng Ni, Abhranil Chandra, Shiguang Guo, Weiming Ren, Aaran Arulraj, Xuan He, Ziyan Jiang, Tianle Li, Max Ku, Kai Wang, Alex Zhuang, Rongqi Fan, Xiang Yue, and Wenhu Chen. 2024. MMLU-pro: A more robust and challenging multi-task language understanding benchmark. In *The Thirty-eight Conference on Neural Information Processing Systems Datasets and Benchmarks Track*.

Zengzhi Wang, Fan Zhou, Xuefeng Li, and Pengfei Liu. 2025. Octothinker: Mid-training incentivizes reinforcement learning scaling. *Preprint*, arXiv:2506.20512.

Alexander Wettig, Aatmik Gupta, Saumya Malik, and Danqi Chen. 2024. Qurating: Selecting high-quality data for training language models. In *International Conference on Machine Learning*, pages 52915–52971.

Can Xu, Qingfeng Sun, Kai Zheng, Xiubo Geng, Pu Zhao, Jiazhan Feng, Chongyang Tao, Qingwei Lin, and Daxin Jiang. 2025. Wizardlm: Empowering large pre-trained language models to follow complex instructions. *Preprint*, arXiv:2304.12244.

Zitong Yang, Neil Band, Shuangping Li, Emmanuel Candes, and Tatsunori Hashimoto. 2025. Synthetic continued pretraining. In *The Thirteenth International Conference on Learning Representations*.

Rowan Zellers, Ari Holtzman, Yonatan Bisk, Ali Farhadi, and Yejin Choi. 2019. HellaSwag: Can a machine really finish your sentence? In *Proceedings of the 57th Annual Meeting of the Association for Computational Linguistics*, pages 4791–4800.

Ge Zhang, Scott Qu, Jiaheng Liu, Chenchen Zhang, Chenghua Lin, Chou Leuang Yu, Danny Pan, Esther Cheng, Jie Liu, Qunshu Lin, Raven Yuan, Tuney Zheng, Wei Pang, Xinrun Du, Yiming Liang, Yinghao Ma, Yizhi Li, Ziyang Ma, Bill Lin, and 26 others. 2024. Map-neo: Highly capable and transparent bilingual large language model series. *Preprint*, arXiv:2405.19327.

Xuemiao Zhang, Can Ren, Chengying Tu, Rongxiang Weng, Shuo Wang, Hongfei Yan, Jingang Wang, and Xunliang Cai. 2026. Expanding reasoning potential in foundation model by learning diverse chains of thought patterns. In *The Fourteenth International Conference on Learning Representations*.

Fan Zhou, Zengzhi Wang, Nikhil Ranjan, Zhoujun Cheng, Liping Tang, Guowei He, Zhengzhong Liu, and Eric P. Xing. 2025. Megamath: Pushing the limits of open math corpora. *Preprint*, arXiv:2504.02807.

Kun Zhou, Beichen Zhang, jiapeng wang, Zhipeng Chen, Xin Zhao, Jing Sha, Zhichao Sheng, Shijin Wang, and Ji-Rong Wen. 2024. Jiuzhang3.0: Efficiently improving mathematical reasoning by training small data synthesis models. In *The Thirty-eighth Annual Conference on Neural Information Processing Systems*.

A Experimental Details

A.1 Seed Details

We construct the seed dataset from three complementary sources to ensure broad content coverage: (1) approximately 350 million existing QA pairs, which provide structured knowledge across a wide range of disciplines; (2) about 500,000 books, offering in-depth, long-form domain knowledge—each book is segmented into passages, and 1-to-N synthesis is applied to each passage independently; and (3) around 10 million high-quality web pages, selected via our dual-criteria KnowEdu pipeline that

independently scores each document for knowledge density (using QuRater) and educational utility (using FineWeb-Edu), retaining only those in the top-5% on both dimensions.

Seed Source	Scale	Synthetic QA Proportion
QA Pairs	$\sim$ 350M	$\sim$ 80%
HQ Web Pages	$\sim$ 10M	$\sim$ 15%
Books	$\sim$ 500K	$\sim$ 5%

Table 6: Details of seed source.

The resulting synthetic QA pairs are distributed across seed sources as shown in Table 6. Each seed generates n=10 QA pairs per synthesis call. To upsample STEM content, we perform multiple synthesis rounds on seeds from underrepresented STEM disciplines (using a temperature of 0.6 to encourage output variation) while reducing synthesis frequency for non-STEM categories. Consequently, the final data distribution does not simply mirror the seed distribution—STEM-related content is intentionally concentrated.

A.2 Training Details

We adopt DeepSeek-R1 as the synthesis model and DeepSeek-V3 as the answer refinement model, configured to a temperature of 0.6, a top-p value of 0.95, and a top-k value of -1. All computations are executed on a dedicated cluster of 300 H20-141G GPUs. We use 256 Ascend 910B NPUs to mid-train Llama-3 8B from the 2T-token checkpoint, using 40B tokens, with each model training for over 22 hours. We implement it via the Megatron framework, optimized by the Adam algorithm with standard parameters $\beta_1 = 0.9$ and $\beta_2 = 0.95$. The training employs a global batch size of 960 and a linearly decaying learning rate schedule initialized at 1.9×10^{-4} and terminating at 1.9×10^{-5}.

For the model size scalability experiments, we evaluate models with 1.7B, 8B, and 16B parameters using 40B tokens. For the 1.7B model, we use 80 NPUs for training, with each model trained for over 38 hours. For the 16B model, we use 480 NPUs for training, with each model trained for over 21 hours. Table 7 presents the model architectures for the 1.7B, 8B, and 16B models.

We further analyze the computational cost and data scale required to construct 1M QA samples. Specifically, generating 1M QA pairs using DeepSeek-R1 requires 514.07 GPU hours on H20 GPUs, while answer refinement with DeepSeek-V3 incurs an additional 318.72 GPU hours. For reference, 1M pure QA samples correspond to 0.093B tokens, whereas 1M CoT-augmented QA samples correspond to 0.437B tokens.

Hyperparameter	1.7B	8B	16B
Precision	bfloat16	bfloat16	bfloat16
Layers	24	32	40
Hidden Size	2048	4096	5120
Attention Heads	32	32	64
Head Type	GQA	GQA	GQA
Intermediate Size	8192	14336	18432
Vocab Size	131072	131072	163840
Sequence Length	8192	8192	8192
Activation	SiLU	SiLU	SiLU
Position Embedding	RoPE	RoPE	RoPE

Table 7: Model architecture for the 1.7B, 8B, and 16B models.

A.3 Datasets

We compare BoostQA with various large-scale QA datasets of different types and from different sources, as detailed in Table 8. For N-CC, it originally consists of a 9:1 ratio of documents to QA pairs. We conduct two sets of experiments on N-CC: one using the original 40B N-CC dataset, and another using a 40B dataset formed by a 1:1 mixture of KnowEdu and the N-CC QA subset.

Dataset	Synthesis	CoT	Type	Domain	Tokens
N-CC	✔	✗	Doc. + QA	General	499.5B
N-CC QA	✔	✗	QA	General	51B
YulanQA	✗	✔	QA	General	4.92B
N-MIND	✔	✔	Conv.	Math	138B
MMQA	✔	✔	QA	Math	7.0B
JZ	✔	✔	QA	Math	4.6B
BoostQA	✔	✗	QA	General	100B
$\text{BoostQA}_{\text{CoT}}$	✔	✔	QA	General	140B

Table 8: Comparison with large-scale QA datasets. $\text{BoostQA}_{\text{CoT}}$ extends BoostQA by incorporating supplementary math CoT data $\text{BoostQA}_{\text{MC}}$. In practice, the complete CoT dataset may contain more tokens.

A.4 Scaling Details

The specific accuracy values of the final checkpoint in the scaling experiments are presented in Table 9.

A.5 Performance on MMLU Subsets

The MMLU subset results shown in Table 10 further substantiate BoostQA's cross-domain superiority, outperforming general corpora and other synthetic QA baselines across all categories: MMLU-

Scale	Settings	MMLU	CMMLU	C-Eval	STEM
Model Size	1.7B: KnowEdu	45.32	48.95	45.39	38.63
	1.7B: BoostQA	52.49	55.92	54.01	47.08
	8B: KnowEdu	63.53	68.08	67.18	53.86
	8B: BoostQA	68.26	72.16	71.32	61.28
	16B: KnowEdu	68.61	72.53	69.23	61.44
	16B: BoostQA	72.34	76.32	73.77	66.21
Data Volume	40B: KnowEdu	58.17	62.99	61.98	49.16
	40B: BoostQA	64.78	68.00	66.65	57.71
	190B: KnowEdu	60.72	67.12	65.07	51.11
	190B: BoostQA	70.63	73.66	73.76	66.74
Initial FLOPs	2T: KnowEdu	58.17	62.99	61.98	49.16
	2T: BoostQA	64.78	68.00	66.65	57.71
	10T: KnowEdu	63.53	68.08	67.18	53.86
	10T: BoostQA	68.26	72.16	71.32	61.28

Table 9: Accuracy of the final checkpoint in the scaling experiments. Abbreviation: STEM = MMLU-STEM.

Dataset	Social	Humanity	Other
Pre-train	63.85	59.44	58.26
FW-Edu	63.50	60.48	58.67
KnowEdu	67.20	<u>62.44</u>	60.32
N-CC	66.92	60.98	60.74
N-CC QA	67.31	61.84	<u>61.88</u>
YulanQA	65.65	59.28	58.84
N-MIND	<u>67.40</u>	61.23	59.60
MMQA	63.02	58.39	59.18
JZ	65.19	59.80	58.37
BoostQA	**72.83**	**66.26**	**66.40**

Table 10: Comparison on MMLU subsets. The best and second best are in **bold** and <u>underlined</u>, respectively.

social, MMLU-humanity, and MMLU-other. This consistent advantage persists despite structural divergence in knowledge representation, confirming that our synthesis framework fundamentally enhances conceptual generalization rather than optimizing domain-specific heuristics. Crucially, BoostQA generates transferable reasoning signals through its STEM-focused educational diversified progression and difficulty booster. When integrated with high-quality educational corpora, it establishes robust cross-disciplinary cognitive scaffolding unavailable in the baselines. The observed performance patterns validate BoostQA's unique capacity to transcend discipline boundaries while maintaining task-agnostic robustness.

B Findings

B.1 Educational Stage Alignment

We randomly sample 10,000 instances from each educational stage (high school, college, and graduate) within the multi-grade synthetic dataset and employ DeepSeek-V3 to reassess their stage alignment. The distribution results, shown in Table 11, reveal an automatic decline in the educational stage alignment of synthetic data: high school-targeted requests predominantly yield junior high school content, college-targeted requests generate high school-level output, and graduate-level requests produce college-level data. Consequently, achieving target-level question generation necessitates instructing the model to synthesize higher-tier content. For example, specifying graduate-level prompts to obtain university-level questions.

Targeted Stage	Match Acc.	Actual Grade Distribution					
		prim.	junior.	high.	coll.	grad.	other
high.	30.31	2.31	**55.60**	0.03	<u>38.62</u>	0.10	3.34
coll.	42.87	1.40	<u>36.71</u>	**56.41**	0.16	0.59	4.73
grad.	7.86	0.63	18.70	<u>24.93</u>	**52.38**	0.01	3.35

Table 11: Stage alignment testing for multi-grade synthesis. The largest and second largest proportions are in bold and underlined, respectively. Abbreviations: prim. = primary school, junior. = junior high school, high. = high school, coll. = college, grad. = graduate.

B.2 Comparison of Refinement Models

We test the performance of synthetic datasets refined by DeepSeek-V3 and DeepSeek-R1. The results in Table 12 show that the average effects of both models are similar.

Metric	DeepSeek-V3	DeepSeek-R1
MMLU	64.78	64.77
CMMLU	68.00	67.78
GSM8K	47.73	48.79
MATH	14.00	13.50
HellaSwag	60.50	62.00
WinoGrande	44.00	45.00
ARC-C	80.00	78.00
AVG.	54.14	54.26

Table 12: Comparison of different refinement models.

B.3 Difficulty Distribution Comparison

We compare the difficulty distribution of BoostQA with N-CC QA, another general synthetic QA dataset. As illustrated in Figure 8, BoostQA demonstrates a significant increase in the proportion of high-difficulty (H4/H5) questions.

C Synthesis Model Analysis

To systematically investigate the impact of different synthesis models, we conduct comparative exper-

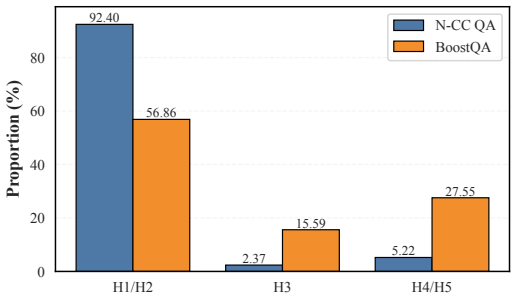


Figure 8: Difficulty distribution comparison between N-CC QA and BoostQA.

iments on strong open-source models. The open-source nature of these models enables large-scale deployment for synthesis, such as DeepSeek-R1 and Qwen2.5-72B-Instruct. For comparison analysis, we uniformly sample 1,000 seed instances as $\mathcal{S}_{\text{test}}$ across all difficulty levels and use both models to synthesize data for analysis.

DeepSeek-R1 achieves higher semantic diversity and greater expansion from seed data. We first fix both models to synthesize $n = 10$ instances for each seed $s \in \mathcal{S}_{\text{test}}$ (*i.e.*, the same seed set and the same expansion quantity). Using Sentence-T5(Ni et al., 2022), we embed the synthesized instances and compute the mean pairwise cosine distance over the entire synthesized set:

$$D_{\text{pairwise}} = \frac{1}{\binom{N}{2}} \sum_{1 \le i < j \le N} \left(1 - \cos(e_i, e_j)\right), \quad (6)$$

where $N = |\mathcal{S}_{\text{test}}| \cdot n$ is the total number of synthesized instances for each model, and e_i denotes the embedding of the i-th synthesized instance. DeepSeek-R1 achieves $D_{\text{pairwise}} = 0.1732$, which is **1.24**$\times$ higher than Qwen's 0.1402.

Furthermore, we measure how far each model diffuses from an individual seed by the intra-seed spread. For each seed $s \in \mathcal{S}_{\text{test}}$, let $\{g_{s,i}\}_{i=1}^{n}$ be the embeddings of its n synthesized instances. We compute:

$$S_{\text{intra}}^{(s)} = \frac{1}{\binom{n}{2}} \sum_{1 \le i < j \le n} \left(1 - \cos(g_{s,i}, g_{s,j})\right), \quad (7)$$

and report the average over seeds, $S_{\text{intra}} = \frac{1}{S} \sum_{s=1}^{S} S_{\text{intra}}^{(s)}$. As shown in Figure 9, DeepSeek-R1 achieves an average intra-seed spread of 0.1175, which is **2.81**$\times$ higher than Qwen's 0.0418. Moreover, DeepSeek-R1 exhibits higher intra-seed spread than Qwen on **92.6%** of all seeds.

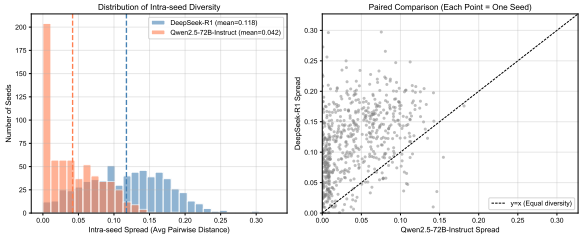


Figure 9: Embedding visualization of synthesized data. DeepSeek-R1 demonstrates greater semantic spread both globally and within individual seeds.

DeepSeek-R1 provides better coverage of seed knowledge points while introducing more novel ones. We annotate knowledge points (KPs) for both synthesized and seed data using a unified KP labeling pipeline. Compared to the seed KP set, DeepSeek-R1 covers **23.4%** of seed KPs and introduces **842.2%** new KPs. By contrast, Qwen covers only 17.6

DeepSeek-R1 maintains diversity as the expansion quantity n increases, while Qwen degrades. By varying the per-seed expansion number $n \in \{5, 10, 15, 20\}$, we measure intra-seed diversity (average pairwise cosine distance within each seed's outputs) and global diversity (average pairwise cosine distance across all outputs). As shown in Figure 10, DeepSeek-R1 maintains stable diversity metrics across different values of n, while Qwen's intra-seed diversity decreases by 10% (from 0.043 to 0.039) as n increases. Furthermore, the diversity gap between the two models widens with larger n, with the R1/Qwen ratio increasing from 2.68$\times$ to 3.10$\times$ for intra-seed diversity. These results indicate that DeepSeek-R1 is more suitable for large-scale data synthesis without diversity degradation.

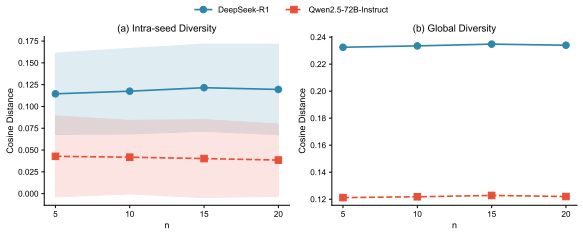


Figure 10: Comparison of diversity between DeepSeek-R1 and Qwen2.5-72B-Instruct across different expansion quantity n. (a) Intra-seed diversity with standard deviation bands. (b) Global diversity. DeepSeek-R1 remains stable while Qwen's intra-seed diversity degrades as n increases.

In summary, under controlled conditions, DeepSeek-R1 demonstrates clear advantages over Qwen2.5-72B-Instruct in both synthesis diversity and knowledge coverage. Consequently, we select DeepSeek-R1 as the synthesis model for BoostQA.

D Expansion Analysis

In addition to comparing the impact of the synthetic model on data diversity, we further compared the effects of one-to-many and one-to-one generation strategies. We sampled 100 seed questions (20 per difficulty level from H1 to H5) and generated questions using the same model (DeepSeek-V3) with n=10 and temperature 0.6. For the 1-to-N condition, each seed produces 10 questions in a single call. For the N×(1-to-1) condition, each seed is used in 10 independent calls, each producing one question. We then computed the same embedding-based diversity metrics from Appendix C.

Metric	1-to-N	N×(1-to-1)	Ratio
Global pairwise cosine distance (Eq.6)	0.8125	0.8040	1.01×
Intra-seed spread (Eq.7)	0.3962	0.2597	1.53×
Unique knowledge points	2340	1653	1.42×

Table 13: Comparison between 1-to-N and N×(1-to-1) strategies.

As shown in Table 13, the 1-to-N strategy yields 53% higher intra-seed diversity and covers 42% more unique knowledge points than the N×(1-to-1) strategy. This is because when generating multiple questions in a single call, the model is explicitly conditioned to produce differentiated outputs, actively avoiding repetition. In contrast, independent single-question calls lack this inter-question awareness and tend to converge on similar patterns. Moreover, the N×(1-to-1) approach wastes N-1 copies of the input context without leveraging it for diversification.

E Annotation System

The discipline classifier categorizes content into 62 disciplines[3]. To ensure labeling efficiency while maintaining quality, we implement a two-stage annotation pipeline. First, we use DeepSeek-R1 with the discipline-constrained prompt and generate preliminary labels on 20M seed samples. Next, we perform uniform stratified sampling across all 62 disciplines to curate a balanced subset of 500K high-confidence samples. This subset is used to finetune Qwen2.5-7B-Instruct, resulting in a specialized discipline classifier. Empirical validation shows an 82.18% label consistency between our specialized classifier and DeepSeek-R1, which confirms reliable knowledge distillation. The prompt used for data annotation with discipline constraints and for training the corresponding labeler is shown below.

Prompt for Discipline Classifier

Act as an educational taxonomist. Classify the input question into our standardized discipline hierarchy using sequential reasoning, then output strictly in JSON format:
1. Primary Discipline Identification
   Select exactly one primary discipline from:
   {Discipline List}
   – Use "cross–discipline" only for explicit multi–domain integration
   – Assign "Other" only if no discipline matches >=60% relevance
2. Secondary Discipline Assignment
   – Identify the most specific applicable sub–discipline
   – Null if the primary discipline has no sub–domains
   – Use "General" for non–specialized content
3. Validation Rules
   – Reject non–educational content –> Output "Invalid"
   – Correct spelling/terminology variations before classification

Output Schema:
```
{
  "primary_discipline": "",
  "secondary_discipline": "",
  "confidence": 0.0–1.0,
  "rejection_reason": null
}
```
Input: {Data}

The list of 62 disciplines is as follows.

[3]GB/T 13745-2008 taxonomy

Discipline List (62)

['Mathematics', 'Computer Science and Technology', 'Clinical Medicine', ' Chemistry', 'Economics', 'Information Science and Systems Science', 'Physics', ' Biology', 'Law', 'Philosophy', 'Sociology', ' Literature', 'Psychology', 'Statistics', ' History', 'Power and Electrical Engineering', 'Earth Science', ' Management Science', 'Electronics and Communication Technology', 'Linguistics', 'Preventive Medicine and Public Health', ' Political Science', 'Education Science', ' Aerospace Science and Technology', ' Astronomy', 'Materials Science', ' Mechanics', 'Sports Science', 'Ethnology and Cultural Studies', 'Basic Medicine', ' Environmental Science and Resource Science', 'Journalism and Communication', 'Religious Studies', 'Engineering and Technology Related to Information and Systems Science', 'Food Science and Technology', 'Engineering and Technology ', 'Art Studies', 'Mechanical Engineering', ' Traditional Chinese Medicine and Chinese Materia Medica', 'Pharmacy', 'Civil and Architectural Engineering', 'Chemical Engineering', 'Nuclear Science and Technology', 'Marxism', 'Agronomy', ' Energy Science and Technology', ' Transportation Engineering', 'Military Science', 'Safety Science and Technology', 'Animal Husbandry and Veterinary Science', 'Archaeology', 'Engineering and Technology Related to Product Applications', 'Library, Information and Documentation Science', 'Geomatics Science and Technology', 'Aquaculture Science', 'Metallurgical Engineering Technology', 'Hydraulic Engineering', ' Military Medicine and Special Medicine', ' Textile Science and Technology', 'Mining Engineering Technology', 'Forestry', ' Engineering and Technology Related to Natural Sciences']

In parallel, the difficulty assessor utilizes human performance metrics, defined as pass rates under standardized one-hour testing conditions with QS Top 100 university students majoring in relevant disciplines, to calibrate five difficulty tiers (H1-H5). We employ a multi-stage annotation pipeline. Initially, DeepSeek-R1 generates difficulty labels for 500K QA pairs using a structured prompt that simulates human problem-solving behaviors. These preliminary annotations are then used to fine-tune Qwen2.5-14B-Instruct, effectively distilling the knowledge. Expert assessment by five PhD evaluators (Krippendorff's α = 0.85) demonstrates a strong correlation (Pearson's r = 0.92) between model predictions and actual human performance metrics. The prompt used for difficulty annotation is presented below.

Prompt for Difficulty Assessor

Act as an educational assessment expert, analyze the provided question through sequential reasoning and output strictly in JSON format:
1. Knowledge Analysis
  – Core concepts (<=3): [comma–separated list]
  – Integration type: {single–concept | cross–chapter | cross–discipline}
2. Cognitive Tier (Bloom's Taxonomy)
  {memory | understanding | application | analysis | synthesis | evaluation}
3. Difficulty Assessment
  – Estimated pass rate (P) for QS Top 100 university majors: [0–100%]
  – Tier:
    – extreme: P < 10%
    – challenge: 10% <= P < 30%
    – improvement: 30% <= P < 50%
    – standard: 50% <= P < 80%
    – basic: P >= 80%
    – other: invalid inputs
4. Exception Handling
  – Mark "other" for non–questions/ unanswerable items
  – Correct minor errors (e.g., missing correct options) before assessment
  – Ignore provided solutions/answers

Output Schema:
{
  "difficulty_tier": "basic | standard | improvement | challenge | extreme | other",
  "rationale": [

"Involves {N} core knowledge points
",
"Cognitive level: {Bloom's tier}",
"Estimated pass rate: approximately {
XX}% for target cohort"
]
}
Input: {Data}

F Synthesis Prompts

The prompts and specific rules used to synthesize diverse QA pairs are as follows.

Prompt for Multi-Grade Synthesis

Act as a {Educational Role} educator, analyze the knowledge points assessed by the provided seed data. Generate { Expansion Quantity} novel questions adhering to these requirements:

1. Questions must demonstrate substantial differentiation while testing the application of identified knowledge points.
2. Difficulty must align with {Educational Role} standards through:
   a) Down−scaling overqualified knowledge points to prerequisite concepts;
   b) Up−scaling underqualified points to advanced applications.
3. Linguistic consistency must be maintained with the input question.
{Format−specific Constraints}

Output Schema: {Format−specified JSON}

Input: {Seed Data}

{Educational Role} r can be assigned as *high school*, *college*, or *graduate*. In our experiment, {Expansion Quantity} n is set to 10. {Format-specific Constraints} and {Format-specific JSON} are controlled by the generative question type t and follows the specific rules below:

Rules for Generative Question Type

Multiple−Choice:
{Format−specific Constraints}
4. The generated question type is multiple −choice. For each question, four alternative options must be generated, and among the four options, there must be one correct answer.
{Format−specified JSON}
[{"question": "", "options": [], " answer_index": 0−3}, ...]

Essay−question:
{Format−specific Constraints}
4. The generated question type is essay− question. For each question, the solution steps and the final correct answer are provided. The generated questions cannot be open−ended questions (such as those of the solution type, thinking type, information listing type, etc.), but must be self−contained with a final answer that can be determined as correct.
{Format−specified JSON}
[{"question": "", "solution": "", "answer": ""}, ...]

The following prompt for the high-difficulty synthesis is based on the multi-grade one but specifies the role as graduate and adds a difficulty booster part.

Prompt for High-Difficulty Synthesis

Act as a graduate educator, analyze the knowledge points assessed by the provided seed data. Generate {Expansion Quantity} novel questions adhering to these requirements:

1. Questions must demonstrate substantial differentiation while testing the application of identified knowledge points
2. Difficulty must align with graduate standards through:
   a) Down−scaling overqualified knowledge points to prerequisite concepts
   b) Up−scaling underqualified points to

advanced applications
3. Linguistic consistency must be maintained with the input question
{Format−specific Constraints}

[Difficulty Booster]
1. Knowledge Analysis:
- Core concepts (<=3)
- Integration type: {single | cross−chapter | cross−discipline}
2. Cognitive Tier (Bloom's Taxonomy):
{memory | understanding | application | analysis | synthesis | evaluation}
3. Difficulty Validation:
- Estimate pass rate 0 <= P <= 100%
- Tier:
  - extreme: P < 10%
  - challenge: 10% <= P < 30%
  - improvement: 30% <= P < 50%
  - standard: 50% <= P < 80%
  - basic: P >= 80%
- REJECT if not challenge/extreme (P >= 30%)

Output Schema: {Format−specified JSON}

Input: {Seed data}

The following is the prompt for answer refinement.

Prompt for Answer Refinement

Please strictly follow the requirements below to analyze the given question and answer:
Answer Requirements
1. Perform step−by−step reasoning and present the complete thought process, which must include:
- Extraction of key information from the question
- Application of relevant formulas or theorems
- Reminders of common error types
- Display of logical reasoning chains
2. If the problem is unsolvable, then output an empty answer. Otherwise, provide a refined answer that fits the format requirements.
3. Notes
- If the question already includes solution steps and answers, please ignore them and do not be influenced by them, as they may be incorrect or suboptimal.
- For multiple−choice questions, if the correct answer is missing:
  - Add a fifth option: "(E) [Correct Answer]"
  - Set answer_index=4
  - Keep the original options unchanged

Output Schema: {Format−specified JSON}

Input: {Data}

G Case Study

Table 14: Case of different seed and different synthesis.

Seed Data 1 (High-quality Web Page)	**Seed Data 2 (QA)**
To divide a 4-digit number by a single-digit number, follow these steps: 1. Divide the first digit of the 4-digit number by the single-digit number. If the result is less than the single-digit number, consider the first two digits. 2. Multiply the result by the single-digit number and subtract it from the considered digits. 3. Bring down the next digit and repeat the process until all digits are considered. 4. The final result will be the quotient. Examples: ...	How does Nuclear Magnetic Resonance Imaging (MRI) work, and what factors affect the signal produced by the nuclei in the magnetic field? Answer: Nuclear Magnetic Resonance Imaging (MRI) is a non-invasive medical imaging technique that uses magnetic fields and radio waves to generate detailed images of internal body structures. It relies on the principles of nuclear magnetic resonance (NMR), primarily targeting hydrogen nuclei (protons) in water and fat molecules due to their abundance in biological tissues. Below is a step-by-step explanation of how MRI works, followed by key factors affecting the signal.
Synthetic QA pairs (multi-grade, multiple-choice, n = 10)	**Synthetic QA pairs (multi-grade, multiple-choice, n = 10)**
Question 1: When dividing a 4-digit number by a single-digit number, what should you do if the first digit is smaller than the divisor? A. Consider only the first digit B. Consider the first two digits C. Skip to the next digit immediately D. Add the divisor to the first digit	**Question 1**: In MRI, what is the primary function of the gradient magnetic field during slice selection? A. To generate T1-weighted contrast B. To spatially encode the position of nuclei along the z-axis C. To amplify the RF excitation pulse D. To reduce T2* decay artifacts
Answer 1: B. Consider the first two digits	**Answer 1**: B. To spatially encode the position of nuclei along the z-axis
Question 2: What must always be true about the remainder in division? A. Remainder equals the divisor B. Remainder is greater than the divisor C. Remainder is less than the divisor D. Remainder must be zero	**Question 2**: Which tissue property directly determines proton density-weighted image contrast? A. T1 relaxation time B. T2 relaxation time C. Concentration of hydrogen nuclei D. Magnetic susceptibility
Answer 2: C. Remainder is less than the divisor	**Answer 3**: C. Concentration of hydrogen nuclei
Question 3: Which equation correctly verifies division results? A. Divisor × Quotient = Dividend B. Divisor × Remainder = Dividend C. Divisor × Quotient + Remainder = Dividend D. Quotient × Remainder = Dividend	**Question 3**: Why does cerebrospinal fluid (CSF) appear bright in T2-weighted MRI images? A. Short T1 relaxation time B. High proton density C. Long T2 relaxation time D. Low diffusion coefficient
Answer 3: C. Divisor × Quotient + Remainder = Dividend	**Answer 3**: C. Long T2 relaxation time
Question 4: In the division process, after multiplying and subtracting, what do you bring down next? A. The previous remainder B. The divisor C. The next digit of the dividend D. The quotient digit	**Question 4**: What is the primary source of T2* decay in gradient-echo sequences? A. Spin-spin interactions B. Magnetic field inhomogeneities C. RF pulse imperfections D. Gradient coil heating
Answer 4: C. The next digit of the dividend	**Answer 4**: B. Magnetic field inhomogeneities
Question 5: What is the first step when calculating 5,274 ÷ 6? A. Divide 5 by 6 B. Divide 52 by 6 C. Divide 527 by 6 D. Divide 5274 by 6	**Question 5**: If an MRI image exhibits 'zipper artifacts' along the frequency-encoding direction, what is the most likely cause? A. Insufficient TR time B. RF interference from external sources C. Gradient amplifier failure D. Patient motion during scan
Answer 5: B. Divide 52 by 6	**Answer 5**: B. RF interference from external sources
...	...
Synthetic QA pairs (high-difficulty, multiple-choice, n = 10)	**Synthetic QA pairs (high-difficulty, multiple-choice, n = 10)**
Question 1: In a modified division algorithm where remainders are allowed to be negative, when dividing 4107 by 6, what would be the smallest absolute value remainder possible? A. 3 B. -3 C. 2 D. -2	**Question 1**: In MRI imaging, when comparing T1-weighted and T2-weighted images, which parameter combination primarily determines the contrast difference? A. TR=500ms, TE=20ms B. TR=3000ms, TE=100ms C. TR=2000ms, TE=20ms D. TR=500ms, TE=80ms
Answer 1: B. -3	**Answer 1**: B. TR=3000ms, TE=100ms

Table 14: (continued)

Question 2: Given the division 7608 ÷ 8 = 951, if a single-digit error occurs in the quotient's tens place during calculation, what maximum absolute difference could this error create in the final result? A. 80 B. 40 C. 800 D. 8	**Question 2**: For a spin-echo sequence, what happens to the MR signal if both TR and TE are doubled while maintaining other parameters? A. T1 weighting increases B. Signal-to-noise ratio improves C. T2 weighting dominates D. Specific absorption rate decreases
Answer 2: A. 80	**Answer 2**: C. T2 weighting dominates
Question 3: Using the division principle from the examples, what is the minimum number of multiplication/verification steps required to divide any 4-digit number by a single-digit divisor? A. 4 steps B. 5 steps C. Depends on carries D. Always 3 steps	**Question 3**: Which factor has the MOST significant impact on the Larmor frequency in clinical MRI systems? A. Patient body temperature B. Main magnetic field strength C. Radiofrequency pulse amplitude D. Gradient coil performance
Answer 3: A. 4 steps	**Answer 3**: B. Main magnetic field strength
Question 4: For the division 6585 ÷ 5 = 1317, if the dividend is increased by k while keeping the same quotient, what is the maximum k such that the remainder doesn't change? A. 4 B. 5 C. 10 D. 15	**Question 4**: When implementing fat suppression techniques, which physical property difference is primarily exploited? A. T1 relaxation time B. Proton density C. Chemical shift D. Magnetic susceptibility
Answer 4: A. 4	**Answer 4**: C. Chemical shift
Question 5: For a cryptographic application using modulo 9 arithmetic, what is 3408 mod 3 equivalent to, given the original division example? A. 0 B. 1 C. 2 D. 3	**Question 5**: What is the primary reason for using phase encoding gradients in MRI? A. Slice selection B. Frequency determination C. Spatial localization in one dimension D. Magnetic field homogenization
Answer 5: A. 0	**Answer 5**: C. Spatial localization in one dimension
...	...

Table 15: Samples for synthetic QA pairs of different disciplines.

Mathematics
Sample 1: Question: A machine fills 120 bottles in 2.5 hours. What is the average time to fill one bottle? A. 1.25 minutes B. 1.5 minutes C. 1.75 minutes D. 2.0 minutes
Answer: A. 1.25 minutes
Sample 2: Question: In a modified deck with 13 cards per suit (52 total), what is the expected number of draws until three consecutive cards of the same suit appear? Round to the nearest tenth.
Answer: 401.4
Computer Science and Technology
Sample 1: Question: Which BEST describes the method resolution order (MRO) impact when using mixins in Python? A. Mixins always take precedence over base classes B. MRO follows C3 linearization rules C. Depth-first left-right search order D. Mixins are ignored in super() calls
Answer: B. MRO follows C3 linearization rules
Sample 2: Question: Calculate a doubly linked list function that supports fuzzy deletion, which requires randomly deleting nodes based on a given probability distribution while maintaining the integrity of the remaining linked list structure. Please describe the selection and implementation methods of the probability model.
Answer: Implementation steps: 1) Select the probability model and assign node probabilities. 2) Randomly select nodes based on cumulative distribution. 3) Perform the standard doubly linked list deletion operation
Clinical Medicine
Sample 1: Question: Which pathogen is classified as a virus in the given scenario? A. Giardia B. Trichinella C. Hepatitis A D. Salmonella enterica
Answer: C. Hepatitis A
Sample 2: Question: Which three serum biomarkers would best differentiate pre-eclampsia-related pulmonary edema from septic ARDS in a postpartum patient?
Answer: sFlt-1, PlGF, and Procalcitonin.
Chemistry

Table 15: (continued)

Sample 1: Question: A linear polymer is synthesized via condensation reactions using 20 monomers. How many water molecules are released? A. 19 water molecules B. 20 water molecules C. 18 water molecules D. 21 water molecules

Answer: A. 19 water molecules

Sample 2: Question: Design a theoretical calculation scheme to determine whether O_2 adsorption on the catalyst surface is physical or chemical adsorption through quantum chemical methods, and list three key calculation parameters.

Answer: The key calculation parameters are: adsorption energy (E_ads), charge density difference (CDD), and O_2 bond length variation.

Economics

Sample 1: Question: If a depositor holds $300,000 in a single checking account at an FDIC-insured bank, how much of this amount is covered by FDIC insurance? A. $100,000 B. $250,000 C. $300,000 D. $500,000

Answer: B. $250,000

Sample 2: Question: Based on the mechanism of real estate mortgage loans, analyze the fundamental legal obstacles that make it difficult for developing countries to popularize mortgage loans and their second-order impact on the structure of the financial market.

Answer: The fundamental legal obstacles to the popularization of mortgage loans in developing countries mainly include the imperfect property rights registration system, the low efficiency of law enforcement and the difficulty in disposing of collateral. The second-order impact of these obstacles on the structure of the financial market is manifested as insufficient development of long-term financing tools, increased reliance on informal financial channels to increase systemic risks, and restrictions on the depth and breadth of the financial market.

Information Science and Systems Science

Sample 1: Question: What is the primary technological factor that diminished RIM's corporate market advantage? A. Superior hardware design by competitors B. Improved security features in iOS devices C. Development of 5G network technology D. Reduction in smartphone production costs

Answer: B. Improved security features in iOS devices

Sample 2: Question: In the detection of abnormal behaviors in social networks, how to construct an abnormal scoring function for user behavior sequences based on the hidden Markov model?

Answer: The anomaly scoring function for the user behavior sequence constructed based on the hidden Markov model is $S(X) = -\log P(X|\lambda)$, where $P(X|\lambda)$ is calculated through the forward algorithm. Parameter estimation is iteratively optimized using the Baum-Welch algorithm.

Physics

Sample 1: Question: A train 200 m long crosses a man walking at 5 km/h in the same direction in 15 seconds. What is the train's speed? A. 45 km/h B. 50 km/h C. 53 km/h D. 58 km/h

Answer: C. 53 km/h

Sample 2: Question: The object is moving at an initial velocity of 20m/s and decelerates uniformly. The displacement in the last second before it stops is 2m. Find the magnitude of the acceleration.

Answer: 4

Biology

Sample 1: Question: Which symbiotic relationship is characterized by one organism benefiting while the host is harmed but rarely killed? A. Mutualism B. Commensalism C. Parasitism D. Amensalism

Answer: C. Parasitism

Sample 2: Question: Two pairs of alleles (X/x, Y/y) are inherited independently, with X being dominant and Y being dominant (the presence of Y masks the X phenotype). The homozygous parents are XXYY (Y dominant) and xxyy (non-dominant), and F_1 self-pollinates to F_2. Calculate the proportion of individuals with phenotypes different from those of their parents in F_2.

Answer: 3/16

Law

Sample 1: Question: A clinic uses an unencrypted email service to send lab results to patients. Which safeguard is most critically missing? A. Multi-factor authentication B. Business Associate Agreement C. Technical protection for ePHI in transit D. Annual HIPAA training

Answer: C. Technical protection for ePHI in transit

Sample 2: Question: County Party Secretary A promised to change the land use for developer B, but demanded that B donate the bribe of 2 million yuan to the charity foundation designated by A. How to determine the nature of Party A's actions?

Table 15: (continued)

Answer: Party A's actions constitute the crime of accepting bribes, with the amount of the bribe being 2 million yuan.

Table 16: Samples for synthetic QA pairs of different difficulty levels.

H1
Sample 1: Question: Which file extension is typically associated with pip executables on Windows? A. .py B. .sh C. .exe D. .dll Answer: C. .exe ***Sample 2:*** Question: A regular octahedron has 6 vertices and 12 edges. A segment that joins two vertices not joined by an edge is called a diagonal. How many diagonals does a regular octahedron have? Answer: 3
H2
Sample 1: Question: What is the primary environmental concern regarding offshore drilling in Arctic conditions? A. Air emissions from rig operations B. Ice-scouring damage to subsea infrastructure C. Drill cuttings dispersion in cold water D. Noise pollution affecting marine mammals Answer index: B. Ice-scouring damage to subsea infrastructure ***Sample 2:*** Question: The chord length of the circle $x^2 + y^2 = 9$ intersecting the line $4x + 3y + c = 0$ is 6. Find the value of c. Answer: 0
H3
Sample 1: Question: Which phenomenon best illustrates the transformation process described in the definition? A. A novelist copyrighting their manuscript B. A TikTok story becoming a published anthology C. Academic peer-review process D. Direct translation of sacred texts Answer: B. A TikTok story becoming a published anthology ***Sample 2:*** Question: The pulley block connects objects with masses m1=3kg and m2=2kg. Find (a) the increase in system kinetic energy within 2 seconds after release; (b) the change in mechanical energy of m1. (Pulley mass and friction are disregarded, $g=10m/s^2$) Answer: (a) 40J; (b) -96J
H4
Sample 1: Question: Which of the following expressions satisfies the property of being divisible by 3^{n+1} but not by 3^{n+2}? A. $7^{2^n} - 1$ B. $4^{3^n} + 1$ C. $5^{2^n} - 2$ D. $2^{3^n} + 1$ Answer: D.$2^{3^n} + 1$ ***Sample 2:*** Question: When the difference in the two electron integrals of the two programs increases significantly with the increase in atomic spacing, the most likely misaligned parameter is: A. Integral storage format B. Schwarz screening threshold C. Density fitting accuracy D. Basis set hypershrinkage parameters Answer: B. Schwarz screening threshold
H5
Sample 1: Question: Define $g(x)$ as the highest power of 5 dividing x. For $n > 0$, $S_n = \sum_{k=1}^{5^{n-1}} g(5k)$. If the largest $n < 1000$ with S_n a perfect square is 499, what is X? A. X = 2 B. X = 3 C. X = 5 D. X = 7 Answer: C. X = 5 ***Sample 2:*** Question: When the photon energy approaches the Planck scale, the influence of quantum fluctuations on Einstein's field equations is mainly reflected in: A. Renormalization of energy-momentum tensors B. Fractional-dimensionalization of differential structures C. Topological invariance breaking of the connection term D. Supersymmetric extension of the curvature tensor Answer: A. Renormalization of energy-momentum tensors