MathMixup: Boosting LLM Mathematical Reasoning with Difficulty-Controllable Data Synthesis and Curriculum Learning

Xuchen Li[1,2,3*], **Jing Chen**[5*†], **Xuzhao Li**, **Hao Liang**,
Xiaohuan Zhou[5‡], **Taifeng Wang**[5], **Wentao Zhang**[3,4‡]
[1]Institute of Automation, Chinese Academy of Sciences
[2]School of Artificial Intelligence, University of Chinese Academy of Sciences
[3]Zhongguancun Academy, [4]Peking University, [5]ByteDance
s-lxc24@bza.edu.cn, wentao.zhang@pku.edu.cn

Abstract

In mathematical reasoning tasks, the advancement of Large Language Models (LLMs) relies heavily on high-quality training data with clearly defined and well-graded difficulty levels. However, existing data synthesis methods often suffer from limited diversity and lack precise control over problem difficulty, making them insufficient for supporting efficient training paradigms such as curriculum learning. To address these challenges, we propose MathMixup, a novel data synthesis paradigm that systematically generates high-quality, difficulty-controllable mathematical reasoning problems through hybrid and decomposed strategies. Automated self-checking and manual screening are incorporated to ensure semantic clarity and a well-structured difficulty gradient in the synthesized data. Building on this, we construct the MathMixupQA dataset and design a curriculum learning strategy that leverages these graded problems, supporting flexible integration with other datasets. Experimental results show that MathMixup and its curriculum learning strategy significantly enhance the mathematical reasoning performance of LLMs. Fine-tuned Qwen2.5-7B achieves an average score of 52.6% across seven mathematical benchmarks, surpassing previous state-of-the-art methods. These results fully validate the effectiveness and broad applicability of MathMixup in improving the mathematical reasoning abilities of LLMs and advancing data-centric curriculum learning.

1 Introduction

Recent advances in Large Language Models (LLMs), such as DeepSeek R1 (Guo et al., 2025a), Kimi K1.5 (Team et al., 2025) and OpenAI o3 (Achiam et al., 2023), have led to remarkable progress in mathematical reasoning tasks. These

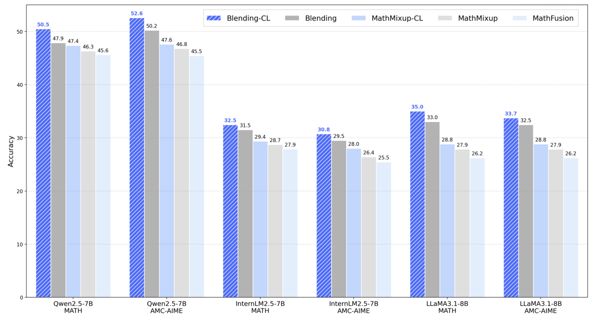

Figure 1: Average accuracy comparison of different datasets and curriculum learning (CL) strategies on seven mathematical reasoning benchmarks across three LLMs (Qwen2.5-7B, InternLM2.5-7B, and LLaMA3.1-8B), with all datasets synthesized separately from MATH and AMC-AIME seeds. MathMixup consistently outperforms the MathFusion baseline, and curriculum learning (MathMixup-CL) further improves performance. Blending MathMixupQA and MathFusionQA (Blending) yields additional gains, while Blending-CL achieves the highest accuracy across all models and settings. These results demonstrate that difficulty-controllable data and curriculum learning are both effective individually, and their combination leads to the greatest improvements in LLM mathematical reasoning.

models have demonstrated impressive performance across a variety of mathematical benchmarks (Tang et al., 2024a; Hendrycks et al., 2021; Cobbe et al., 2021; Saxton et al., 2019), narrowing the gap between artificial and human-level problem solving. However, despite these advancements, a critical bottleneck remains: further improvement in LLM mathematical reasoning is fundamentally constrained by the lack of high-quality training data with explicit and controllable difficulty gradients (Lu et al., 2024).

Against this backdrop, it is crucial to systematically examine the roles of data quality, diversity, and organization in mathematical reasoning tasks (Shen et al., 2025). In this context, data augmentation and synthesis (Wang et al., 2024; Zhou et al., 2024; Fedoseev et al., 2024; Chen et al., 2025b) have become promising approaches to providing

*Equal contribution.
†Project leader.
‡Correspondence authors.

Findings of the Association for Computational Linguistics: ACL 2026, pages 2497–2513
July 2-7, 2026 ©2026 Association for Computational Linguistics

high-quality and diverse data. Existing data synthesis methods can be mainly categorized into two types: the first type enhances or rewrites individual problems (Yu et al., 2024; Tang et al., 2024b; Liu et al., 2024a; Zhang et al., 2024), which, despite increasing the dataset size, often results in generated questions that are highly similar in semantics and difficulty, thus lacking diversity; the second type, exemplified by MathFusion (Pei et al., 2025), merges two similar problems to increase computational complexity. While these methods can increase problem complexity to some extent, they still lack systematic and explicit control over reasoning difficulty, which is essential for robust capability improvement. Meanwhile, curriculum learning (Liu et al., 2024b; Wang et al., 2025a) has proven to be an effective paradigm for further improving model reasoning abilities, and its success is fundamentally dependent on the availability of well-graded training data with clear and controllable difficulty progression (Chen et al., 2025a; Zhang et al., 2025a). However, current datasets and synthesis methods are inadequate in providing such difficulty stratification and gradients, which limits the potential of curriculum learning. Therefore, there is an urgent need for a new method capable of generating mathematical reasoning data with controllable and well-graded difficulty, which can not only enhance model capability directly but also facilitate further improvement through curriculum learning.

To address these challenges, we propose a novel data synthesis paradigm—MathMixup—which centers on difficulty-controllable hybrid and decomposed strategies to generate high-quality, explicitly graded mathematical reasoning data, thereby both directly improving model capability and laying the groundwork for subsequent curriculum learning. As illustrated in Figure 2, the overall process consists of three main stages: (1) Difficulty-Controllable Question Synthesis: Based on large-scale mathematical datasets, we use BGE embedding (Xiao et al., 2023) to construct pairs of similar questions with different difficulty levels, and employ both hybrid generation (to create more challenging new problems) and decomposed generation (to produce problems of intermediate difficulty), thereby achieving fine-grained control over difficulty. (2) Dataset Construction: All generated questions undergo automated self-checking and manual screening to ensure semantic clarity and well-graded difficulty. For each synthesized problem, we generate high-quality solutions using QwQ-32B, guided by auxiliary information from similar original questions and their answers. We then apply automated post-processing—including answer format validation and content deduplication—to further ensure the accuracy and consistency of the solutions. This pipeline results in a structured dataset, MathMixupQA, with explicit and reliable difficulty stratification. (3) Curriculum Learning: During model training, we fully leverage the graded data in MathMixupQA, not only implementing progressive curriculum learning on our own dataset, but also flexibly mixing MathMixupQA data with other public datasets to explore more efficient and diverse training strategies.

Based on the MATH (Hendrycks et al., 2021) and AMC-AIME training sets, we construct a new dataset, MathMixupQA, featuring controllable difficulty gradients. Experimental results show that curriculum learning based on difficulty-controllable data significantly improves model performance in mathematical reasoning tasks. After curriculum learning SFT on MathMixupQA, Qwen2.5-7B achieves an average score of 47.6% across seven mathematical benchmarks. Furthermore, when MathMixupQA is mixed with MathFusionQA for curriculum learning, the average performance of Qwen2.5-7B (Qwen et al., 2025) increases to 52.6%, setting a new SOTA.

In summary, this work makes three key contributions: **1)** We propose MathMixup, a framework for difficulty-controllable data synthesis, and construct the MathMixupQA dataset. MathMixup systematically generates mathematical reasoning problems with explicit and controllable difficulty, providing high-quality, well-structured data resources for improving LLM mathematical reasoning. **2)** We introduce a curriculum learning strategy that leverages the explicit difficulty gradients in MathMixupQA, and supports flexible integration with other datasets. By combining the graded data generated by MathMixupQA with existing datasets, we achieve more efficient and flexible curriculum learning for LLMs. **3)** Extensive experiments demonstrate the effectiveness and generalizability of our approach. Data synthesis paradigm and curriculum learning strategies of MathMixup significantly enhance the mathematical reasoning abilities of LLMs, outperforming existing baselines and achieving new state-of-the-art results.

2 Related Work

2.1 Mathematical Reasoning in LLMs

Large language models (LLMs) have made significant strides in mathematical reasoning, driven by advances in architecture, scaling laws, and large-scale mathematical datasets. Recent models such as DeepSeek R1 (Guo et al., 2025a), OpenAI o3 (Achiam et al., 2023), Kimi K1.5 (Team et al., 2025), and Qwen3 (Yang et al., 2025) have achieved higher accuracy and reasoning depth on various benchmarks. However, some LLMs (Grattafiori et al., 2024; Jiang et al., 2023; Cai et al., 2024) still struggle with complex multi-step reasoning and precise logical deduction. Techniques such as chain-of-thought prompting (Xu et al., 2025a; Zhang et al., 2025b), external tool augmentation (Jiang et al., 2025; Zheng et al., 2025), supervised fine-tuning (Wen et al., 2025b; Ma et al., 2025a; Tian et al., 2025), and reinforcement learning (Guan et al., 2025; Xu et al., 2025b; Huang et al., 2025; Chung et al., 2025) have been proposed to address these issues. Nevertheless, the lack of high-quality, difficulty-stratified training data remains a key bottleneck. Our work targets this gap by synthesizing training data with controllable difficulty for supervised fine-tuning of LLMs.

2.2 Data Synthesis for Math Reasoning

High-quality and diverse datasets (Saveliev et al., 2025; He et al., 2025; Li et al., 2024b,c) are essential for mathematical reasoning in LLMs. Recent works (Luo et al., 2023; Yu et al., 2024; Pei et al., 2025; Liu et al., 2024a) have explored automatic problem generation, paraphrasing, and augmentation to enrich data diversity, including template-based and adversarial example construction (Mitra et al., 2024; Li et al., 2024a; Huang et al., 2024; Tang et al., 2024b). However, most synthetic datasets (Zhang et al., 2024) still lack fine-grained control over problem complexity and often do not guarantee validity or solvability. Difficulty-controllable synthesis remains largely unexplored. In contrast, our approach enables explicit and fine-grained difficulty control during data synthesis, ensuring each question is both valid and appropriately challenging.

2.3 Curriculum Learning for Reasoning

Curriculum learning (Shi et al., 2025; Wang et al., 2025b) improves LLM training efficiency and generalization, especially for tasks with distinct difficulty levels, such as mathematical reasoning. By organizing data according to explicit difficulty levels or conceptual progression (Ma et al., 2025b; Deng et al., 2025), curriculum learning allows models to build foundational knowledge before tackling harder problems. Recent studies (Song et al., 2025; Wen et al., 2025a; Xia et al., 2025) have explored manual annotation, automatic difficulty estimation, and adaptive sampling for curriculum design. Yet, reliably constructing controllable difficulty gradations—particularly when blending synthetic data—remains challenging. Our approach addresses this by synthesizing mathematical questions with controllable difficulty, enabling more effective curriculum learning both within our dataset and in combination with others.

3 Method

3.1 Overview

The overall MathMixup pipeline is illustrated in Figure 2. Our method consists of three systematic stages: (1) **Controllable Question Synthesis**, which includes constructing pairs of similar questions and generating new questions with controllable difficulty via hybrid and decomposed strategies; (2) **Dataset Construction**, involving verifying question quality, solution generation (with auxiliary information) and automated post-processing to build the structured MathMixupQA dataset; and (3) **Curriculum Learning**, where we leverage the graded data for progressive curriculum learning, both on MathMixupQA alone and in combination with other datasets, enabling more flexible and effective model training.

3.2 Difficulty-Controllable Question Synthesis

3.2.1 Question Pairs Construction

To ensure the difficulty and quality of the synthesized data, we use MATH (Hendrycks et al., 2021) and AMC-AIME as seed training datasets to construct question pairs. The MATH training set contains a total of 7.5K math problems. For the AMC-AIME dataset, we compile AIME problems from 1984-2023 and AMC problems prior to 2023 as training data, totaling 4K math problems. Let the dataset be denoted as:

$$\mathcal{D} = \{(q_i, a_i, d_i)\}_{i=1}^{N}$$

where q_i is the i-th question, a_i is its answer, and d_i is the corresponding difficulty level. The difficulty labels d_i are sourced from official annotations. For

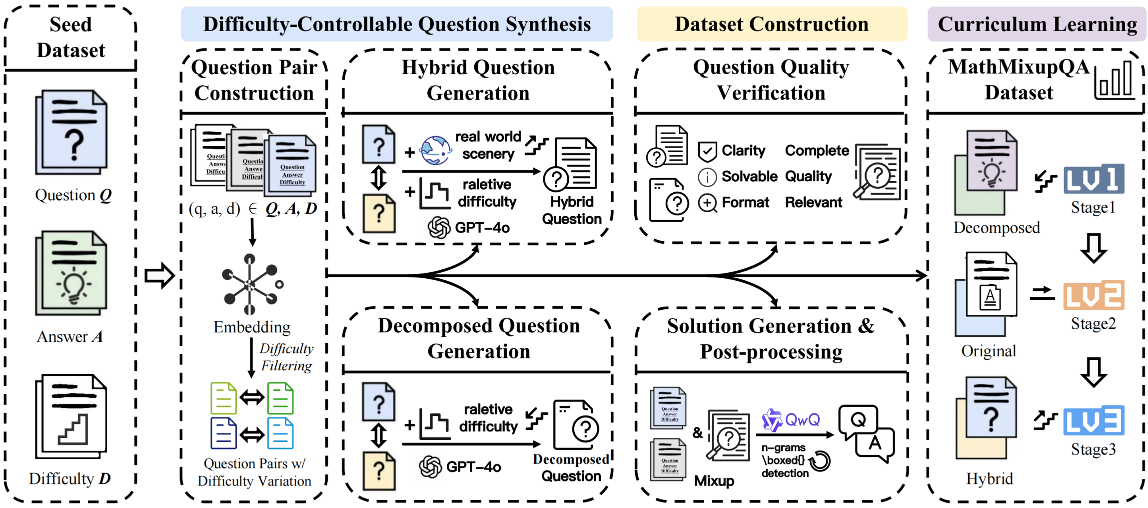

Figure 2: Overview of the MathMixup pipeline. The process includes question pairs construction, question generation (hybrid and decomposed), question verification, solution generation with auxiliary information, and automated post-processing. MathMixup enables the synthesis of high-quality data with controllable difficulty, and supports curriculum learning strategies that further enhance LLM mathematical reasoning performance.

each question q_i, we compute its embedding using BGE embeddings (Xiao et al., 2023):

$$\mathbf{e}_i = \text{BGE}(q_i)$$

where $\mathbf{e}_i \in \mathbb{R}^d$ is the embedding vector for question q_i. To identify similar question pairs, we calculate the cosine similarity between embeddings:

$$\text{sim}(q_i, q_j) = \frac{\mathbf{e}_i \cdot \mathbf{e}_j}{\|\mathbf{e}_i\| \|\mathbf{e}_j\|}$$

We then construct the set of similar question pairs:

$$\mathcal{P} = \left\{ \big((q_i, a_i, d_i), (q_j, a_j, d_j)\big) \;\middle|\; \begin{matrix} \text{sim}(q_i, q_j) > \tau, \\ d_i \neq d_j \end{matrix} \right\}$$

where τ is a predefined similarity threshold, and $d_i \neq d_j$ ensures that the paired questions have different difficulty levels. By retaining only those pairs with different difficulty levels, we enable better control over the difficulty when synthesizing new questions. This pairing strategy provides the foundation for generating both easier and harder variants of the original problems, supporting explicit difficulty stratification in subsequent synthesis.

3.2.2 Question Generation

After obtaining pairs of similar questions, we use GPT-4o (Achiam et al., 2023) to perform mixup on the two questions in two distinct ways.

For Hybrid Question Generation, the objective is to create a new question that is more difficult than the harder of the two originals. This is achieved by analyzing the mathematical structures, variables,

Case 1: *Hybrid Question Generation*

Original Question 1: *How many integers $n \geq 2$ are there such that whenever $z_1, z_2, \ldots, z_n$ are complex numbers with*

$$|z_1| = |z_2| = \cdots = |z_n| = 1 \quad \textit{and}$$

$$z_1 + z_2 + \cdots + z_n = 0,$$

then the numbers $z_1, z_2, \ldots, z_n$ are equally spaced on the unit circle in the complex plane?
Difficulty 1: *7.0*

Original Question 2: *How many nonzero complex numbers z have the property that 0, z, and z^3, when represented by points in the complex plane, are the three distinct vertices of an equilateral triangle?*
Difficulty 2: *4.0*

Hybrid Question: *A researcher is studying wave patterns generated by buoys anchored at points on the unit circle in the complex plane. The buoys are placed at n points $z_1, z_2, \ldots, z_n$ such that:*

1. *Each z_k is a complex number with $|z_k| = 1$.*
2. *The buoys are equally spaced around the unit circle (i.e., $z_k = e^{2\pi i k/n}$ for $k = 0, 1, \ldots, n-1$), so $z_1 + z_2 + \cdots + z_n = 0$.*
3. *For each buoy z_k, the points 0, z_k, and z_k^3 are three distinct vertices of an equilateral triangle in the complex plane.*

For how many values of $n \geq 2$ is it possible to place the buoys in this way?
Difficulty: *8.0*

and solution strategies of both problems, then identifying common themes or real-world contexts to construct a unified scenario. The resulting question integrates core elements from both problems and introduces realistic constraints, thereby increasing both difficulty and contextual richness. An example is shown in Case 1.

For Decomposed Question Generation, the goal is to synthesize a question whose difficulty lies between that of the two originals. This is done by decomposing and recombining elements from both problems, simplifying complex dependencies while retaining essential concepts. This process produces intermediate-difficulty questions that support a smooth and controllable difficulty progression for curriculum learning. A representative case is shown in Case 2.

Case 2: *Decomposed Question Generation*

Original Question 1: *How many integers $n \geq 2$ are there such that whenever $z_1, z_2, \ldots, z_n$ are complex numbers with*

$$|z_1| = |z_2| = \cdots = |z_n| = 1 \quad \textit{and}$$

$$z_1 + z_2 + \cdots + z_n = 0,$$

then the numbers $z_1, z_2, \ldots, z_n$ are equally spaced on the unit circle in the complex plane?
Difficulty 1: *7.0*

Original Question 2: *How many nonzero complex numbers z have the property that 0, z, and z^3, when represented by points in the complex plane, are the three distinct vertices of an equilateral triangle?*
Difficulty 2: *4.0*

Decomposed Question: *Let z_1, z_2, z_3, z_4 be complex numbers with $|z_1| = |z_2| = |z_3| = |z_4| = 1$ and $z_1 + z_2 + z_3 + z_4 = 0$. How many distinct quadruples (z_1, z_2, z_3, z_4) (up to rotation) are there such that the points are not all equally spaced on the unit circle?*
Difficulty: *5.0*

For both generation strategies, prompts guide the creation of logically clear, self-contained problems with unique answers, and include self-check steps to ensure completeness, solvability, and clarity. After hybrid and decomposed question generation, we obtain synthesized data with controllable difficulty compared to the original dataset. The explicit construction of decomposed, original, and hybrid questions provides a three-level difficulty gradient, which is directly leveraged in our curriculum learning framework. For detailed prompts, please refer to Appendix A.

3.3 Dataset Construction

3.3.1 Question Verification

To ensure the quality and reliability of synthesized questions, we adopt a two-stage verification process, focusing on key dimensions such as clarity, completeness, formatting, relevance, solvability, and logical flow. In the first stage, GPT-4o (Achiam et al., 2023) performs automated self-checks and corrections, filtering out questions with obvious issues in these aspects. In the second stage, we manually validate a randomly selected 10% sample of the questions, applying the same criteria. This ensures that each question is clear and understandable, contains all necessary conditions and parameters, uses standard mathematical notation, and is appropriate for the intended academic level. Questions exhibiting serious ambiguity, missing information, logical contradictions, or irrelevant assumptions are flagged as low-quality and excluded. This consistent two-stage process guarantees that all synthesized questions are logically consistent, meaningful, and suitable for inclusion in the dataset, thereby ensuring the overall quality of the difficulty-controlled data.

3.3.2 Solution Generation

We used QwQ-32B (Team, 2025) to generate solutions for the synthesized data. Existing approaches often rely solely on models like GPT-4o (Pei et al., 2025) or use majority voting (Guo et al., 2025b), which increases inference costs and may not guarantee correctness, especially for complex problems. In our method, we incorporate original questions and their answers as auxiliary information when generating solutions for new questions. This additional context guides the model to produce answers that are more relevant and better matched to the intended difficulty. Leveraging such mixup-based information reduces noisy outputs and improves efficiency compared to majority voting, which simply selects among independently generated answers.

3.3.3 Solution Post-processing

To further ensure the correctness of the generated solutions, we perform post-processing and, if necessary, regeneration. Specifically, we check each solution for the presence of properly formatted `\boxed{}` expressions, ensuring that the answer is complete and not empty. Additionally, we conduct n-gram (2-gram and 3-gram) analysis to detect and prevent repetitive or meaningless content. If any such issues are found, the solution is regenerated. This post-processing step further enhances the quality and clarity of the final solutions in our dataset, making them suitable for robust LLM training.

3.4 Curriculum Learning

In the final stage, we train models directly on MathMixupQA dataset and leverage the difficulty-controllable data for curriculum learning (CL) to further enhance the mathematical reasoning abilities of LLMs.

3.4.1 Curriculum Learning on MathMixupQA

We design a curriculum learning framework that exploits the explicit difficulty gradation in MathMixupQA. The synthesized questions are organized into several levels (decomposed, original, hybrid) according to their difficulty, and the model is trained progressively from easier to harder problems. This staged process enables the model to build foundational reasoning skills before tackling more complex concepts, providing a data-driven and controllable curriculum learning strategy.

3.4.2 Blended Curriculum Learning with Other Datasets

To further improve training flexibility and generalization, we extend our approach by blending MathMixupQA with MathFusionQA datasets. We merge and re-rank questions from both datasets based on their difficulty and categories in Figure 3, constructing a unified curriculum that covers a broader range of problem types and complexities. This blended curriculum allows the model to benefit from more data diversity and explicit difficulty stratification, promoting robust reasoning abilities.

4 MathMixupQA Dataset

4.1 Dataset Statistics

With MathMixup, we synthesize both Hybrid and Decomposed types of data for the original MATH and AMC-AIME datasets, forming the MathMixupQA dataset. As shown in Table 1, it contains three difficulty levels—original, hybrid, and decomposed—with 22.5K entries based on MATH and 12K entries based on AMC-AIME, respectively.

Dataset	MATH	AMC-AIME	Total
MathMixupQA (*Decomposed*)	7.5K	4K	11.5K
MathMixupQA (*Original*)	7.5K	4K	11.5K
MathMixupQA (*Hybrid*)	7.5K	4K	11.5K
MathMixupQA	22.5K	12K	34.5K

Table 1: Statistics of the MathMixupQA dataset.

4.2 Difficulty Analysis

As previously mentioned, MathMixup aims to synthesize data with controllable difficulty, forming three explicit difficulty levels to support curriculum learning and enhance model reasoning. To quantitatively analyze the difficulty distribution, we use the difficulty scoring prompts from DeepScaleR (Luo et al., 2025) and GPT-4o to assign scores to both MathMixupQA (Decomposed, Original and Hybrid) and MathFusionQA (Conditional, Parallel and Sequential) (Pei et al., 2025). The results, shown in Figure 3, indicate that the decomposed, original, and hybrid subsets in MathMixupQA form a clear and controllable difficulty gradient, which strongly supports curriculum learning. Notably, the hybrid subset consistently shows higher difficulty scores than MathFusionQA on both MATH and AMC-AIME seeds, further validating the quality and controllability of our data synthesis.

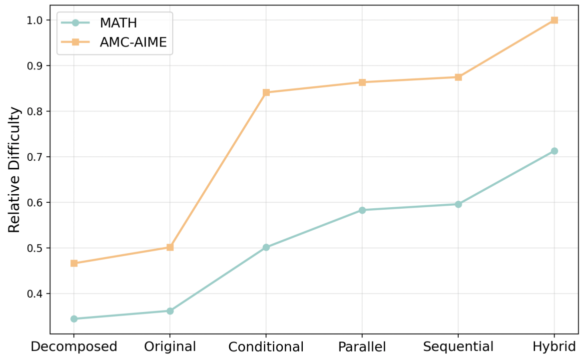

Figure 3: Difficulty scores of different components in the MathMixupQA and MathFusionQA datasets. Note that Conditional, Parallel, and Sequential are synthesis methods proposed in MathFusion.

5 Experiment

5.1 Implementation Details

5.1.1 Data Synthesis.

We use GPT-4o (Achiam et al., 2023) to synthesize new questions from similar question pairs, and use QwQ-32B (Team, 2025) to generate the long CoT responses. We empirically set the similarity threshold τ in the range of 0.75 to 0.9.

5.1.2 Training.

For supervised fine-tuning (SFT), we use 360-LLaMA-Factory (Zou et al., 2024) to train each model for 3 epochs, with a cosine learning rate schedule, 1% warm-up, maximum learning rate 1e-5, and sequence length 32,768 tokens. Experiments are conducted on Qwen2.5-7B (Qwen et al., 2025), LLaMA3.1-8B (Grattafiori et al., 2024), and InternLM2.5-7B (Cai et al., 2024). Each model is trained directly on MathMixupQA based on different seed datasets and we use a three-stage SFT process aligned with the decomposed, original, and hybrid subsets for curriculum learning. To ensure fair comparison, we also reproduce MathFusion using the same data synthesis protocol as our baseline. Further details are provided in the Appendix B. To evaluate generalizability, we conduct additional experiments by blending MathMixupQA with Math-

Method	Seed	# Samples	AIME25	AIME24	OlympiadBench	AMC23	MATH500	Minerva Math	GSM8K	AVG
Qwen2.5-7B										
MathFusion†	MATH	22.5K	27.19	26.77	33.93	44.58	74.60	25.00	87.11	45.60
MathMixup		22.5K	28.13	28.33	35.85	45.78	74.20	25.37	86.58	46.32
MathMixup-CL		7.5K*3	**28.33**	**28.75**	**36.74**	**46.99**	**76.80**	**26.47**	**87.49**	**47.37**
MathFusion†	AMC-AIME	12K	28.23	28.65	34.67	**45.78**	71.40	23.16	86.28	45.45
MathMixup		12K	27.81	30.94	35.70	**45.78**	73.80	26.10	87.49	46.80
MathMixup-CL		4K*3	**28.96**	**32.50**	**36.00**	44.58	**75.40**	**27.57**	**88.17**	**47.60**
InternLM2.5-7B										
MathFusion†	MATH	22.5K	9.27	6.35	18.52	**24.10**	53.80	8.82	74.37	27.89
MathMixup		22.5K	8.54	**7.50**	18.81	**24.10**	**56.80**	10.29	75.06	28.73
MathMixup-CL		7.5K*3	**11.25**	6.88	**19.70**	**24.10**	56.20	**12.13**	**75.28**	**29.36**
MathFusion†	AMC-AIME	12K	11.15	7.50	15.41	18.07	45.20	10.66	70.36	25.48
MathMixup		12K	**13.54**	7.60	**17.78**	19.28	46.40	9.56	70.74	26.41
MathMixup-CL		4K*3	12.29	**9.48**	17.04	**21.69**	**48.00**	**11.76**	**75.80**	**28.01**
LLaMA3.1-8B										
MathFusion†	MATH	22.5K	10.00	5.63	17.33	19.28	48.40	10.29	72.71	26.23
MathMixup		22.5K	**11.25**	**8.13**	17.04	21.69	**52.80**	11.03	73.01	27.85
MathMixup-CL		7.5K*3	10.10	6.15	**19.26**	**26.51**	52.60	**11.40**	**75.89**	**28.84**
MathFusion†	AMC-AIME	12K	11.67	6.67	15.11	14.46	42.20	11.40	68.23	24.25
MathMixup		12K	13.13	8.13	**17.78**	16.87	**43.20**	9.56	73.01	25.95
MathMixup-CL		4K*3	**13.75**	**9.27**	16.89	**18.07**	**43.20**	**12.50**	**73.16**	**26.69**

Table 2: Performance comparison on mathematical benchmarks including AIME25, AIME24, OlympiadBench, AMC23, MATH500, Minerva Math and GSM8K. The table is organized by the base model and seed datasets. Baseline labeled with †, which is our own runs. The best results are highlighted in bold.

FusionQA, constructing a mixed training set. We perform both standard SFT (Blending) and curriculum learning SFT (Blending-CL), with curriculum stages determined by the difficulty gradients in Figure 3. This setup allows us to assess the benefits of integrating our difficulty-controllable data with existing high-quality datasets. Note that we group every two consecutive difficulty levels into one training stage.

5.2 Evaluation and Metrics

We evaluate the fine-tuned models on AMC2023, MATH (Hendrycks et al., 2021), Minerva Math (Lewkowycz et al., 2022), OlympiadBench (He et al., 2024), GSM8K (Cobbe et al., 2021), AIME2024, and AIME2025, using the Light-R1 framework (Wen et al., 2025c). For the first five benchmarks, we report avg@1. Due to the smaller size of AIME2024 and AIME2025, we use the average of 32 random samplings (Avg@32) for more stable results. For detailed hyperparameter settings, please refer to the Appendix C.

5.2.1 SFT on MathMixupQA

As shown in Table 2, MathMixup consistently outperforms the MathFusion baseline across all models and both seed datasets under the same data volume. For example, Qwen2.5-7B fine-tuned on MathMixupQA based on AMC-AIME achieves 46.80% average accuracy, surpassing MathFusionQA's 45.45%. This trend holds across individual benchmarks and other models. Applying curriculum learning (MathMixup-CL) further improves performance for all models, with Qwen2.5-7B reaching 47.60% and notable gains on challenging benchmarks such as OlympiadBench. We also observe that models with higher baseline performance, such as Qwen2.5-7B, benefit more from increased data diversity and curriculum learning, especially when trained with more challenging data. In contrast, models with lower baseline abilities, like LLaMA3.1-8B, gain less from high-difficulty data without sufficient curriculum support. These results highlight the importance of tailoring data difficulty and curriculum design to match model capacity. Overall, MathMixup provides higher-quality, difficulty-controllable training data than existing baselines. The improvements across models and benchmarks demonstrate the robustness and generalizability of our approach.

5.2.2 SFT on Blending Datasets

As illustrated in Figure 4, blending MathMixupQA and MathFusionQA (Blending) further improves model performance compared to using either dataset alone. This effect is amplified with curriculum learning (Blending-CL), as shown by the outward expansion of the radar plots. To account for the large differences in absolute scores across benchmarks, we normalize the results. The gains are consistent across both easier (e.g., GSM8K, MATH500) and more challenging (e.g., AIME24, OlympiadBench) benchmarks. These results confirm that combining difficulty-controllable data with existing high-quality datasets, together with curriculum learning, can fully unlock the potential of LLMs for mathematical reasoning, achieving new state-of-the-art performance. Detailed numerical results are provided in the Appendix E.

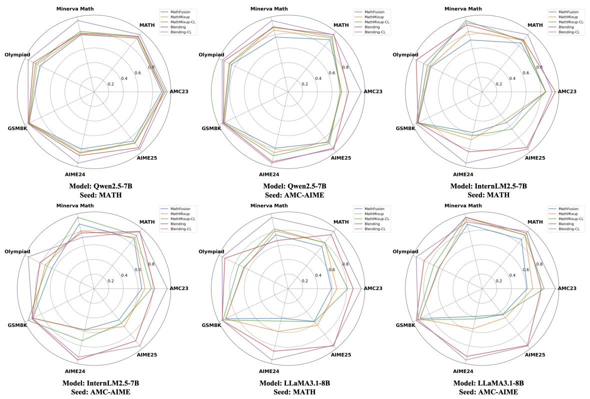

Figure 4: Normalized performance on seven mathematical reasoning benchmarks for three LLMs using different data synthesis and training strategies. "Blending" denotes SFT on the mixed dataset of MathMixupQA and MathFusionQA, while "Blending-CL" denotes curriculum learning SFT on the same mixture.

5.3 Ablation Study

5.3.1 Training Data Components

As shown in Table A4, both hybrid and decomposed data play critical roles in enhancing model reasoning performance. Consistent findings emerge across the two seed datasets (MATH and AMC-AIME): By comparing combinations 1 vs. 3 and 2 vs. 3, we observe that decomposed and hybrid data contribute more significantly to improving reasoning abilities than original data. Furthermore, combination 3 (integrating both decomposed and hybrid data) achieves the optimal results, validating the value of these two data components. Note that we show the average accuracy.

Seed	#	Decomposed	Original	Hybrid	AVG
MATH	1		✓	✓	44.37
	2	✓	✓		43.36
	3	✓		✓	45.67
AMC-AIME	1		✓	✓	44.83
	2	✓	✓		43.59
	3	✓		✓	45.65

Table 3: Effect of data components on Qwen2.5-7B.

5.3.2 Curriculum Leaning Stage Order

As shown in Table A5, curriculum learning that starts with decomposed data followed by hybrid data (Decomposed+Hybrid) consistently achieves the highest accuracy across seven challenging benchmarks. Using original data as the first stage yields lower accuracy, and Decomposed+Original performs worst. These findings demonstrate that introducing decomposed data early in the curriculum better prepares the model for complex questions, highlighting the importance of explicit difficulty control in curriculum design. We provide more results of ablation studies in Appendix G.

Seed	Stage1 Data	Stage2 Data	AVG
MATH	Decomposed	Original	43.24
	Original	Hybrid	46.19
	Decomposed	Hybrid	46.88
AMC-AIME	Decomposed	Original	43.90
	Original	Hybrid	46.72
	Decomposed	Hybrid	47.26

Table 4: Effect of curriculum learning stage order on Qwen2.5-7B across seven benchmarks.

5.3.3 Effectiveness of Solution Generation

As illustrated in Figure 5, the solution generation strategy consistently outperforms majority voting with 16 candidate solutions across benchmarks. While majority voting is commonly used to filter noisy solutions, it is computationally expensive and does not always guarantee correctness. By incorporating relevant original questions and answers as auxiliary information, we yield higher-quality and more reliable solutions, while also reducing inference costs. These improvements further strengthen the overall effectiveness of the MathMixup.

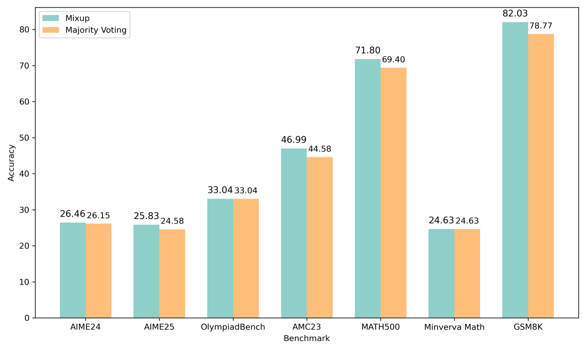

Figure 5: Effectiveness and efficiency of solution generation in MathMixup.

6 Conclusion

We propose MathMixup, a novel method for boosting LLM mathematical reasoning with difficulty-controllable data synthesis and curriculum learning. By explicitly generating and organizing problems with controllable difficulty levels, MathMixup enables the construction of the MathMixupQA dataset and supports the design of controllable, data-driven curricula for LLMs.

Limitations

Although MathMixup provides a principled way to synthesize difficulty-controllable mathematical reasoning data, several constraints remain. The automatic generation and validation process cannot fully eliminate subtle errors in problem statements or solutions. Due to the high cost of manual review, only about 10% of the synthesized problems were randomly inspected, which may leave undetected issues in the remaining data. Moreover, our study primarily investigates two-level fusion strategies, leaving richer compositions of multiple problems and alternative retrieval mechanisms largely unexplored. Finally, the constructed MathMixupQA dataset, though useful for curriculum learning, is modest in size compared with broader mathematical corpora, and its domain coverage is not yet exhaustive.

Acknowledgments

This work is supported by the Zhongguancun Academy (Grant No.s C20250508), Fundamental and Interdisciplinary Disciplines Breakthrough Plan of the Ministry of Education of China (JYB2025XDXM113), National Natural Science Foundation of China (92470121, 62402016), National Key R&D Program of China (2024YFA1014003), Zhongguancun Academy (C20250204, C20250602), Beijing Major Science and Technology Project (Z251100008125043, Z251100008425023) and High-performance Computing Platform of Peking University.

References

Josh Achiam, Steven Adler, Sandhini Agarwal, Lama Ahmad, Ilge Akkaya, Florencia Leoni Aleman, Diogo Almeida, Janko Altenschmidt, Sam Altman, Shyamal Anadkat, and 1 others. 2023. Gpt-4 technical report. *arXiv preprint arXiv:2303.08774*.

Zheng Cai, Maosong Cao, Haojiong Chen, and 1 others. 2024. Internlm2 technical report. *Preprint*, arXiv:2403.17297.

Xiaoyin Chen, Jiarui Lu, Minsu Kim, Dinghuai Zhang, Jian Tang, Alexandre Piché, Nicolas Gontier, Yoshua Bengio, and Ehsan Kamalloo. 2025a. Self-evolving curriculum for llm reasoning. *arXiv preprint arXiv:2505.14970*.

Zui Chen, Tianqiao Liu, Mi Tian, Qing Tong, Weiqi Luo, and Zitao Liu. 2025b. Advancing math reasoning in language models: The impact of problem-solving data, data synthesis methods, and training stages. *arXiv preprint arXiv:2501.14002*.

Stephen Chung, Wenyu Du, and Jie Fu. 2025. Learning from failures in multi-attempt reinforcement learning. *arXiv preprint arXiv:2503.04808*.

Karl Cobbe, Vineet Kosaraju, Mohammad Bavarian, Mark Chen, Heewoo Jun, Lukasz Kaiser, Matthias Plappert, Jerry Tworek, Jacob Hilton, Reiichiro Nakano, and 1 others. 2021. Training verifiers to solve math word problems. *arXiv preprint arXiv:2110.14168*.

Huilin Deng, Ding Zou, Rui Ma, Hongchen Luo, Yang Cao, and Yu Kang. 2025. Boosting the generalization and reasoning of vision language models with curriculum reinforcement learning. *arXiv preprint arXiv:2503.07065*.

Timofey Fedoseev, Dimitar Iliev Dimitrov, Timon Gehr, and Martin Vechev. 2024. Constraint-based synthetic data generation for llm mathematical reasoning. In *The 4th Workshop on Mathematical Reasoning and AI at NeurIPS'24*.

Aaron Grattafiori, Abhimanyu Dubey, Abhinav Jauhri, and 1 others. 2024. The llama 3 herd of models. *Preprint*, arXiv:2407.21783.

Xinyu Guan, Li Lyna Zhang, Yifei Liu, Ning Shang, Youran Sun, Yi Zhu, Fan Yang, and Mao Yang. 2025. rstar-math: Small llms can master math reasoning with self-evolved deep thinking. *arXiv preprint arXiv:2501.04519*.

Daya Guo, Dejian Yang, Haowei Zhang, Junxiao Song, Ruoyu Zhang, Runxin Xu, Qihao Zhu, Shirong Ma, Peiyi Wang, Xiao Bi, and 1 others. 2025a. Deepseek-r1: Incentivizing reasoning capability in llms via reinforcement learning. *arXiv preprint arXiv:2501.12948*.

Yiduo Guo, Zhen Guo, Chuanwei Huang, Zi-Ang Wang, Zekai Zhang, Haofei Yu, Huishuai Zhang, and Yikang Shen. 2025b. Synthetic data rl: Task definition is all you need. *arXiv preprint arXiv:2505.17063*.

Chaoqun He, Renjie Luo, Yuzhuo Bai, Shengding Hu, Zhen Leng Thai, Junhao Shen, Jinyi Hu, Xu Han, Yujie Huang, Yuxiang Zhang, and 1 others. 2024. Olympiadbench: A challenging benchmark for promoting agi with olympiad-level bilingual multimodal scientific problems. *arXiv preprint arXiv:2402.14008*.

Pengfei He, Yue Xing, Han Xu, Zhen Xiang, and Jiliang Tang. 2025. Multi-faceted studies on data poisoning can advance llm development. *arXiv preprint arXiv:2502.14182*.

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. *arXiv preprint arXiv:2103.03874*.

Chengyu Huang, Zhengxin Zhang, and Claire Cardie. 2025. Hapo: Training language models to reason concisely via history-aware policy optimization. *arXiv preprint arXiv:2505.11225*.

Yiming Huang, Xiao Liu, Yeyun Gong, Zhibin Gou, Yelong Shen, Nan Duan, and Weizhu Chen. 2024. Key-point-driven data synthesis with its enhancement on mathematical reasoning. *arXiv preprint arXiv:2403.02333*.

Albert Q. Jiang, Alexandre Sablayrolles, Arthur Mensch, and 1 others. 2023. Mistral 7b. *Preprint*, arXiv:2310.06825.

Pengcheng Jiang, Jiacheng Lin, Lang Cao, Runchu Tian, SeongKu Kang, Zifeng Wang, Jimeng Sun, and Jiawei Han. 2025. Deepretrieval: Hacking real search engines and retrievers with large language models via reinforcement learning. *arXiv preprint arXiv:2503.00223*.

Aitor Lewkowycz, Anders Andreassen, David Dohan, Ethan Dyer, Henryk Michalewski, Vinay Ramasesh, Ambrose Slone, Cem Anil, Imanol Schlag, Theo Gutman-Solo, and 1 others. 2022. Solving quantitative reasoning problems with language models. *Advances in Neural Information Processing Systems*, 35:3843–3857.

Chen Li, Weiqi Wang, Jingcheng Hu, Yixuan Wei, Nanning Zheng, Han Hu, Zheng Zhang, and Houwen Peng. 2024a. Common 7b language models already possess strong math capabilities. *arXiv preprint arXiv:2403.04706*.

Xuchen Li, Xiaokun Feng, Shiyu Hu, Meiqi Wu, Dailing Zhang, Jing Zhang, and Kaiqi Huang. 2024b. Dtllm-vlt: Diverse text generation for visual language tracking based on llm. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, pages 7283–7292.

Xuchen Li, Shiyu Hu, Xiaokun Feng, Dailing Zhang, Meiqi Wu, Jing Zhang, and Kaiqi Huang. 2024c. Dtvlt: A multi-modal diverse text benchmark for visual language tracking based on llm. *arXiv preprint arXiv:2410.02492*.

Haoxiong Liu, Yifan Zhang, Yifan Luo, and Andrew Chi-Chih Yao. 2024a. Augmenting math word problems via iterative question composing. *Preprint*, arXiv:2401.09003.

Yinpeng Liu, Jiawei Liu, Xiang Shi, Qikai Cheng, Yong Huang, and Wei Lu. 2024b. Let's learn step by step: Enhancing in-context learning ability with curriculum learning. *arXiv preprint arXiv:2402.10738*.

Zimu Lu, Aojun Zhou, Houxing Ren, Ke Wang, Weikang Shi, Junting Pan, Mingjie Zhan, and Hongsheng Li. 2024. Mathgenie: Generating synthetic data with question back-translation for enhancing mathematical reasoning of llms. *arXiv preprint arXiv:2402.16352*.

Haipeng Luo, Qingfeng Sun, Can Xu, Pu Zhao, Jianguang Lou, Chongyang Tao, Xiubo Geng, Qingwei Lin, Shifeng Chen, and Dongmei Zhang. 2023. Wizardmath: Empowering mathematical reasoning for large language models via reinforced evol-instruct. *arXiv preprint arXiv:2308.09583*.

Michael Luo, Sijun Tan, Justin Wong, Xiaoxiang Shi, William Y. Tang, Manan Roongta, Colin Cai, Jeffrey Luo, Li Erran Li, Raluca Ada Popa, and Ion Stoica. 2025. Deepscaler: Surpassing o1-preview with a 1.5b model by scaling rl. Notion Blog.

Xinyin Ma, Guangnian Wan, Runpeng Yu, Gongfan Fang, and Xinchao Wang. 2025a. Cot-valve: Length-compressible chain-of-thought tuning. *arXiv preprint arXiv:2502.09601*.

Xuetao Ma, Wenbin Jiang, and Hua Huang. 2025b. Problem-solving logic guided curriculum in-context learning for llms complex reasoning. *arXiv preprint arXiv:2502.15401*.

Arindam Mitra, Hamed Khanpour, Corby Rosset, and Ahmed Awadallah. 2024. Orca-math: Unlocking the potential of slms in grade school math. *arXiv preprint arXiv:2402.14830*.

Qizhi Pei, Lijun Wu, Zhuoshi Pan, Yu Li, Honglin Lin, Chenlin Ming, Xin Gao, Conghui He, and Rui Yan. 2025. Mathfusion: Enhancing mathematic problem-solving of llm through instruction fusion. *arXiv preprint arXiv:2503.16212*.

Qwen, :, An Yang, Baosong Yang, and Others. 2025. Qwen2.5 technical report. *Preprint*, arXiv:2412.15115.

Evgeny Saveliev, Jiashuo Liu, Nabeel Seedat, Anders Boyd, and Mihaela van der Schaar. 2025. Towards human-guided, data-centric llm co-pilots. *arXiv preprint arXiv:2501.10321*.

David Saxton, Edward Grefenstette, Felix Hill, and Pushmeet Kohli. 2019. Analysing mathematical reasoning abilities of neural models. *arXiv preprint arXiv:1904.01557*.

Chengyu Shen, Zhen Hao Wong, Runming He, Hao Liang, Meiyi Qiang, Zimo Meng, Zhengyang Zhao, Bohan Zeng, Zhengzhou Zhu, Bin Cui, and 1 others. 2025. Let's verify math questions step by step. *arXiv preprint arXiv:2505.13903*.

Taiwei Shi, Yiyang Wu, Linxin Song, Tianyi Zhou, and Jieyu Zhao. 2025. Efficient reinforcement finetuning via adaptive curriculum learning. *arXiv preprint arXiv:2504.05520*.

Mingyang Song, Mao Zheng, Zheng Li, Wenjie Yang, Xuan Luo, Yue Pan, and Feng Zhang. 2025. Fastcurl: Curriculum reinforcement learning with progressive context extension for efficient training r1-like reasoning models. *arXiv preprint arXiv:2503.17287*.

Zhengyang Tang, Xingxing Zhang, Benyou Wang, and Furu Wei. 2024a. Mathscale: Scaling instruction tuning for mathematical reasoning. *arXiv preprint arXiv:2403.02884*.

Zhengyang Tang, Xingxing Zhang, Benyou Wang, and Furu Wei. 2024b. Mathscale: Scaling instruction tuning for mathematical reasoning. In *ICML*. OpenReview.net.

Kimi Team, Angang Du, Bofei Gao, Bowei Xing, Changjiu Jiang, Cheng Chen, Cheng Li, Chenjun Xiao, Chenzhuang Du, Chonghua Liao, and 1 others. 2025. Kimi k1. 5: Scaling reinforcement learning with llms. *arXiv preprint arXiv:2501.12599*.

Qwen Team. 2025. Qwq-32b: Embracing the power of reinforcement learning.

Xiaoyu Tian, Sitong Zhao, Haotian Wang, Shuaiting Chen, Yiping Peng, Yunjie Ji, Han Zhao, and Xiangang Li. 2025. Deepdistill: Enhancing llm reasoning capabilities via large-scale difficulty-graded data training. *arXiv preprint arXiv:2504.17565*.

Ke Wang, Jiahui Zhu, Minjie Ren, Zeming Liu, Shiwei Li, Zongye Zhang, Chenkai Zhang, Xiaoyu Wu, Qiqi Zhan, Qingjie Liu, and 1 others. 2024. A survey on data synthesis and augmentation for large language models. *arXiv preprint arXiv:2410.12896*.

Zhenting Wang, Guofeng Cui, Kun Wan, and Wentian Zhao. 2025a. Dump: Automated distribution-level curriculum learning for rl-based llm post-training. *arXiv preprint arXiv:2504.09710*.

Zixuan Wang, Eshaan Nichani, Alberto Bietti, Alex Damian, Daniel Hsu, Jason D Lee, and Denny Wu. 2025b. Learning compositional functions with transformers from easy-to-hard data. *arXiv preprint arXiv:2505.23683*.

Cheng Wen, Tingwei Guo, Shuaijiang Zhao, Wei Zou, and Xiangang Li. 2025a. Sari: Structured audio reasoning via curriculum-guided reinforcement learning. *arXiv preprint arXiv:2504.15900*.

Liang Wen, Yunke Cai, Fenrui Xiao, Xin He, Qi An, Zhenyu Duan, Yimin Du, Junchen Liu, Lifu Tang, Xiaowei Lv, and 1 others. 2025b. Light-r1: Curriculum sft, dpo and rl for long cot from scratch and beyond. *arXiv preprint arXiv:2503.10460*.

Liang Wen, Yunke Cai, Fenrui Xiao, Xin He, Qi An, Zhenyu Duan, Yimin Du, Junchen Liu, Lifu Tang, Xiaowei Lv, and 1 others. 2025c. Light-r1: Curriculum sft, dpo and rl for long cot from scratch and beyond. *arXiv preprint arXiv:2503.10460*.

Tianle Xia, Liang Ding, Guojia Wan, Yibing Zhan, Bo Du, and Dacheng Tao. 2025. Improving complex reasoning over knowledge graph with logic-aware curriculum tuning. In *Proceedings of the AAAI Conference on Artificial Intelligence*, pages 12881–12889.

Shitao Xiao, Zheng Liu, Peitian Zhang, and Niklas Muennighoff. 2023. C-pack: Packaged resources to advance general chinese embedding. *Preprint*, arXiv:2309.07597.

Silei Xu, Wenhao Xie, Lingxiao Zhao, and Pengcheng He. 2025a. Chain of draft: Thinking faster by writing less. *arXiv preprint arXiv:2502.18600*.

Xin Xu, Yan Xu, Tianhao Chen, Yuchen Yan, Chengwu Liu, Zaoyu Chen, Yufei Wang, Yichun Yin, Yasheng Wang, Lifeng Shang, and 1 others. 2025b. Teaching llms according to their aptitude: Adaptive reasoning for mathematical problem solving. *arXiv preprint arXiv:2502.12022*.

An Yang, Anfeng Li, Baosong Yang, Beichen Zhang, Binyuan Hui, Bo Zheng, Bowen Yu, Chang Gao, Chengen Huang, Chenxu Lv, and 1 others. 2025. Qwen3 technical report. *arXiv preprint arXiv:2505.09388*.

Longhui Yu, Weisen Jiang, Han Shi, Jincheng YU, Zhengying Liu, Yu Zhang, James Kwok, Zhenguo Li, Adrian Weller, and Weiyang Liu. 2024. Metamath: Bootstrap your own mathematical questions for large language models. In *The Twelfth International Conference on Learning Representations*.

Enci Zhang, Xingang Yan, Wei Lin, Tianxiang Zhang, and Qianchun Lu. 2025a. Learning like humans: Advancing llm reasoning capabilities via adaptive difficulty curriculum learning and expert-guided self-reformulation. *arXiv preprint arXiv:2505.08364*.

Yufeng Zhang, Xuepeng Wang, Lingxiang Wu, and Jinqiao Wang. 2025b. Enhancing chain of thought prompting in large language models via reasoning patterns. In *Proceedings of the AAAI Conference on Artificial Intelligence*, pages 25985–25993.

Zhihan Zhang, Tao Ge, Zhenwen Liang, Wenhao Yu, Dian Yu, Mengzhao Jia, Dong Yu, and Meng Jiang. 2024. Learn beyond the answer: Training language models with reflection for mathematical reasoning. In *EMNLP*, pages 14720–14738. Association for Computational Linguistics.

Yuxiang Zheng, Dayuan Fu, Xiangkun Hu, Xiaojie Cai, Lyumanshan Ye, Pengrui Lu, and Pengfei Liu. 2025. Deepresearcher: Scaling deep research via reinforcement learning in real-world environments. *arXiv preprint arXiv:2504.03160*.

Kun Zhou, Beichen Zhang, Zhipeng Chen, Xin Zhao, Jing Sha, Zhichao Sheng, Shijin Wang, Ji-Rong Wen, and 1 others. 2024. Jiuzhang3. 0: Efficiently improving mathematical reasoning by training small data synthesis models. *Advances in Neural Information Processing Systems*, 37:1854–1889.

Haosheng Zou and 1 others. 2024. 360-llama-factory.

Appendix

A Prompts

We present the key steps of the prompts used for Hybrid Generation in Prompt 3, Decomposed Generation in Prompt 4 and Question Verification in Prompt 5, which are partially derived from MathFusion.

B MathFusion Reproduction

Since MathFusion does not perform data synthesis on the AMC-AIME seed dataset and the GSM8K seed dataset it uses is relatively simple, we reproduce MathFusion using the MATH and AMC-AIME seed datasets to ensure a fair comparison. We adopt the three synthesis prompts provided in the original paper (Conditional, Sequential, and Parallel), without controlling for difficulty. Apart from the prompt types and the lack of difficulty control, all other procedures remain consistent with those used in MathMixup to guarantee comparability between the two methods. Additionally, during training, we downsample the synthetic data generated by our MathFusion reproduction to match the amount of synthetic data used in our own method.

C Hyperparameter Settings

C.1 Data Synthesis.

For data synthesis, we set GPT-4o's temperature to 0.7 and the maximum generation length to 4096 tokens. For QwQ-32B, we use the official recommended hyperparameters: temperature 0.6, TopP 0.95, MinP 0, TopK 40, and no repetition penalty. The maximum response length for QwQ-32B is set to 32,768 tokens.

C.2 Training.

For supervised fine-tuning (SFT), we train all parameters on the MathMixupQA dataset using DeepSpeed ZeRO Stage 3 for efficient scaling. The effective batch size per device is 1, with gradient accumulation over 4 steps, and sequence parallelism of 8. We use the Adam optimizer ($\beta_1 = 0.9$, $\beta_2 = 0.95$, $\epsilon = 1 \times 10^{-8}$), weight decay 0.1, and gradient clipping at norm 1.0. Training uses bfloat16, FlashAttention v2, and gradient checkpointing to save memory. All experiments use a fixed random seed for reproducibility.

C.3 Evaluation.

For the standard avg@1 setting, each model is evaluated with temperature 0 and one sample per input. The model generates responses up to 32,768 tokens, with a batch size of 2048 and top-p 0.95. For avg@32, we follow SimpleRL-Zoo's temperature of 1.0 and generate 32 samples per input, keeping other parameters unchanged.

D Dataset Statistics and Comparison

In Table A1, we compare MathMixupQA with previous mathematical datasets. Despite its relatively moderate scale, our experiments demonstrate that MathMixupQA achieves superior performance under the same data volume, surpassing the state-of-the-art (SOTA) MathFusion method. Furthermore, the explicit difficulty control within each component of MathMixupQA enables effective curriculum learning, which further improves the model's mathematical reasoning performance.

E Results of SFT on Blending Datasets

Tables A2 and A3 provide the detailed numerical results for three representative models (Qwen2.5-7B, InternLM2.5-7B, and LLaMA3.1-8B) on seven mathematical benchmarks, using the MathMixupQA dataset with both Blending and Blending-CL settings. Results are reported separately for the MATH and AMC-AIME seed datasets. The raw scores presented in these tables correspond to the original (unnormalized) benchmark results, which serve as the basis for the normalized radar charts shown in Figure 4 of the main text.

F Method

The detailed pseudocode of the MathMixup method is provided in Algorithm 1.

G Ablation Study

We provide the evaluation results of the fine-tuned Qwen2.5-7B on seven benchmarks in Tables A4 and A5.

Case 3: ***Hybrid Question Generation***

Role: Hybrid Scenario Problem Architect

Profile
You are an expert in designing advanced, real-world mathematical problems. Your task is to create a hybrid real-world scenario that seamlessly blends "#Problem 1#" and "#Problem 2#", ensuring their essential mathematical elements are preserved and deeply intertwined, leading to a single, culminating challenge. The new problem must be self-contained and must not reference or mention Problem 1 or Problem 2.

Guidelines
Step 1: Carefully analyze the mathematical structures, variables, solution strategies, standard answers, and difficulty levels of both problems, as well as their possible real-world interpretations.

Step 2: Identify common themes, physical principles, or practical contexts that can naturally link the two problems together.

Step 3: Construct a hybrid scenario where these themes converge, introducing realistic constraints and details that bind the mathematical frameworks of both problems into a coherent, practical setting.

Step 4: Design a sophisticated, unified problem statement where solving the underlying mathematical challenges from both original problems is necessary to resolve the real-world scenario.

Step 5: When constructing the new problem, take into account the standard answers and difficulty levels of both original problems. Avoid making the new problem's answer a simple sum, product, or direct reuse of the original answers. Ensure the new problem's difficulty is at least as high as the more difficult of the two original problems, and ideally, it should present a new layer of challenge or synthesis.

Step 6: The new problem must be a single, standalone question with only one main objective and must not be split into multiple subproblems, parts, or steps. The answer should be unique and clearly defined. Do not require the solver to provide multiple separate answers, perform open-ended analysis, or combine results from different objectives.

Self-Check and Correction:

After constructing the hybrid problem, carefully review it by answering the following:
- Does the problem statement include all necessary information and constraints?
- Does the scenario contain any irrelevant or unnecessary details?
- Is the problem well-posed and solvable (i.e., is there at least one solution, and is the solution unique or well-defined)?
- Does the problem require only one specific answer, and is the objective clearly stated as a single maximization, minimization, or unique solution?

If any issues are found, revise and improve the problem statement to ensure it is complete, relevant, solvable, and meets the single-objective requirement.

Only output the final, corrected problem statement. Do not include the self-check process or any commentary in your final output.

Important:
- Do not mention or allude to Problem 1, Problem 2, or their answers/difficulty in the new problem.
- The final problem must have only one main objective and require only one answer.
- The new problem's objective must be clearly defined and not require separate maximization and minimization.
- Do not split the new problem into multiple subproblems, parts, or steps. Do not require multiple distinct answers or open-ended analysis. It must be a single, unified question with a single, well-defined answer.
- After outputting the new problem statement, do not add any further explanation, commentary, or additional information. End your output immediately after the new problem statement.

Output Format
Please reply strictly in the following format:

#Core Elements#:
(Briefly list the main mathematical concepts, variables, or techniques from both problems that will be integrated.)

#Scenario Integration#:
(Describe how the real-world scenario is constructed to blend the mathematical elements of both problems, and how the scenario ensures the necessity of resolving both underlying mathematical challenges.)

#New Problem#:
(Present the fully integrated, standalone, and self-checked real-world problem statement with a single, clearly defined objective and a unique answer. Do not split it into multiple subproblems or parts. Do not require multiple answers or open-ended analysis. End your output here.)

Case 4: ***Decomposed Question Generation***

Role: Advanced Mathematical Problem Decomposer

Profile:
You excel at decomposing advanced mathematical problems, transforming originally complex problems into easier-to-master questions, while retaining their educational value and challenge. Your task is to analyze a combination of two similar but differently difficult problems, namely "#Problem 1#" and "#Problem 2#". Based on the lower difficulty problem, simplify the higher difficulty problem by reducing its complex parts while retaining the essential concepts. The result should be a new problem that still offers learning and thought value. The new problem must be self-contained and should provide clear steps to reach the solution, without mentioning or referencing Problem 1 or Problem 2.

Guidelines:
Step 1: Carefully analyze the mathematical structures, variables, solution strategies, and difficulty levels of the two challenging problems provided, especially the problem with high difficulty.

Step 2: Identify opportunities to simplify the problem:
- Consider the similarities between the high and low difficulty problems and simplify complex interdependencies between variables to more direct relationships.
- Break the problem down into simple components focused on key concepts.
- Design the new problem so that one logical method leads to the solution.
- Ensure it maintains a logical sequence, clarity, and avoids any confusion. If new variables or conditions are introduced, make their determination clear and unambiguous.

Step 3: Identify common themes of the two problems. After that, consider the way to discompose the two problems and design a new problem. When designing the new problem, consider the standard answers and difficulty levels of the two original problems. Avoid making the answer a simple subtraction, division, or direct reuse of the original answers. Ensure that the difficulty of the new problem is at least as high as the simpler of the two original problems. Ideally, it should present a clearer problem statement.

Step 4: Craft a coherent problem statement that encompasses simplified elements derived from the original complex problems, ensuring all necessary information is provided for solving the problem. Design a clear problem statement fully considering elements from both problems, ensuring all interdependencies are clear and necessary for the solution. The new problem must be a single, standalone question with only one main objective and must not be divided into multiple subproblems, parts, or steps. The answer should be unique and clearly defined.

Self-check and Revision:

After creating the new problem, review it by answering the following questions:
- Does the problem statement include all necessary information and constraints?
- Is the problem well-posed and solvable (i.e., is there at least one solution, and is the solution clear)? If any issues are found, revise and improve the problem statement to ensure it is complete, relevant, and solvable.
- Adjust parts that could lead to misunderstandings or ambiguity.
- Thoroughly review the output to spot any unreasonable elements, ensuring it is an easier version of "#Problem 1#", but has higher difficulty than "#Problem 2#".

Important:
- Do not mention or imply specific details or answers of the original challenging problems in the new problem.
- The final problem must have only one main objective and require only one answer.
- Ensure the new problem remains unified and is not split into multiple subproblems or parts.
- After presenting the new problem statement, do not include further explanation, commentary, or additional information. End your output immediately after the new problem statement.
- For a given set of problem1 and problem2, output only one decomposed problem.
- The generated problem must not be presented in a multiple-choice format.
- Conclude the output immediately after presenting the new statement.

Output Format:
Please respond strictly in the following format, do not include any content from the input.:

#Core Elements#:
(Briefly list the key mathematical concepts, variables, or techniques that will be simplified and integrated into the new problem.)

#Simplification Strategy#:
(Explain how simplification is achieved by reducing complexity and focusing on key concepts. Describe how the new problem's difficulty compares to the original challenging problems.)

#New Problem#:
(Present the new, simplified problem statement, ensuring it has a single objective, is clear, and solvable. End your output here.)

Case 5: *Question Verification*

Your task is to act as an impartial judge to evaluate the statement completeness, correctness, and overall quality of a synthesized math problem according to the following rules:

1. Clarity: Is the problem statement mostly clear and understandable, even if some wording is informal or not perfectly concise?

2. Completeness: Are the main conditions, constraints, and parameters provided, so that a reasonably skilled student could attempt the problem? Minor omissions or the need for standard mathematical assumptions are acceptable.

3. Formatting: Is the problem readable and uses standard mathematical notation, even if there are minor formatting or typographical inconsistencies?

4. Relevance: Is the problem generally relevant and appropriate for the intended academic level and topic?

5. Solvability: Is the problem likely solvable by standard mathematical methods, with at least one reasonable solution? (It is acceptable if the solution is not unique, as long as the problem is meaningful.)

6. Logical Flow: Is the problem statement overall logical and consistent, and free from major irrelevant or confusing information? Minor awkwardness or redundancy is acceptable.

Dataset	**Total Samples**	**Difficulty Levels**
WizardMath	96K	–
MetaMathQA	395K	–
MMIQC	2294K	–
Orca-Math	200K	–
Xwin-Math-V1.1	1440K	–
KPMath-Plus	1576K	–
MathScaleQA	2021K	–
DART-Math-Uniform	591K	–
DART-Math-Hard	585K	–
RefAug	30K	–
MathFusionQA	60K	–
MathMixupQA-M (MATH)	22.5K	**Explicit: 3**
MathMixupQA-A (AMC-AIME)	12K	**Explicit: 3**

Table A1: Comparison between MathMixupQA and previous mathematical datasets. MathMixupQA is the only dataset that explicitly controls difficulty levels during data synthesis, resulting in three well-defined difficulty gradients (Original, Hybrid, Decomposed) for both MATH and AMC-AIME seeds.

Model	Type	AMC23	MATH	Minerva Math	OlympiadBench	GSM8K	AIME24	AIME25
Qwen2.5-7B	Blend	45.78	76.40	25.74	38.07	88.32	29.90	30.83
	Blend-CL	**48.19**	**79.20**	**31.25**	**41.33**	**88.40**	**33.44**	**31.56**
InternLM2.5-7B	Blend	**27.71**	57.20	11.40	**23.70**	74.30	9.38	16.88
	Blend-CL	26.51	**63.00**	**11.76**	21.04	**76.42**	**11.15**	**17.29**
LLaMA3.1-8B	Blend	28.92	62.00	9.19	24.74	**76.88**	11.88	**17.50**
	Blend-CL	**32.53**	**65.60**	**13.60**	**25.78**	76.72	**13.54**	17.29

Table A2: Performance of three models on seven benchmarks with MathMixupQA based on MATH seed dataset.

Algorithm 1: MathMixup Method

Require: Seed datasets $\mathcal{D}_{\text{seed}}$, original data $\mathcal{D}_{\text{orig}}$, (optional) external data $\mathcal{D}_{\text{ext}}$
// 1. Difficulty-Controllable Question Synthesis
for all $\mathcal{D}$ in $\mathcal{D}_{\text{seed}}$ **do**
  for all $(q_i, a_i, d_i) \in \mathcal{D}$ **do**
    $\mathbf{e}_i = \text{BGE}(q_i)$
    for all $(q_j, a_j, d_j) \in \mathcal{D},\ j \neq i$ **do**
      $\mathbf{e}_j = \text{BGE}(q_j)$; $s_{ij} = \cos(\mathbf{e}_i, \mathbf{e}_j)$
      if $s_{ij} > \tau$ **and** $d_i \neq d_j$ **then**
        Add $((q_i, a_i, d_i), (q_j, a_j, d_j))$ to $\mathcal{P}$
      end if
    end for
  end for
end for
for all pair $((q_1, a_1, d_1), (q_2, a_2, d_2)) \in \mathcal{P}$ **do**
  Generate and verify $q^{\text{hyb}}, q^{\text{dec}}$; add to $\mathcal{D}_{\text{syn}}$
end for
// 2. Dataset Construction
for all $q \in \mathcal{D}_{\text{syn}}$ **do**
  Find auxiliary set $\mathcal{A}_q$
  Generate solution $a = \text{QwQ}(q, \mathcal{A}_q)$
  Post-process and add (q, a) to $\mathcal{D}_{\text{MathMixupQA}}$
end for
// 3. Curriculum Learning
Split $\mathcal{D}_{\text{MathMixupQA}}$ by difficulty
if $\mathcal{D}_{\text{ext}} \neq \emptyset$ **then**
  $\mathcal{D}_{\text{blend}} = \text{BlendAndRank}(\mathcal{D}_{\text{MMQA}}, \mathcal{D}_{\text{ext}})$
  split by difficulty
end if
for all curriculum stage $\mathcal{D}_{\text{stage}}$ **do**
  Fine-tune LLM on $\mathcal{D}_{\text{stage}}$
end for

Model	Type	AMC23	MATH	Minerva Math	OlympiadBench	GSM8K	AIME24	AIME25
Qwen2.5-7B	Blend	51.81	**78.20**	27.21	39.26	87.49	35.52	**31.98**
	Blend-CL	**62.65**	77.80	**30.15**	**40.30**	**89.31**	**36.25**	31.46
InternLM2.5-7B	Blend	22.89	**51.60**	**9.19**	19.70	**71.49**	**13.02**	18.65
	Blend-CL	**27.71**	51.40	8.46	**24.00**	70.81	12.50	**20.42**
LLaMA3.1-8B	Blend	25.30	54.60	**11.40**	23.11	**77.18**	13.75	21.98
	Blend-CL	**31.33**	**56.20**	10.66	**26.22**	75.13	**14.48**	**22.19**

Table A3: Performance of three models on seven benchmarks with MathMixupQA based on AMC-AIME seed dataset.

Seed	Decompose	Original	Hybrid	AIME25	AIME24	OlympiadBench	AMC23	MATH500	Minerva Math	GSM8K
MATH		✓	✓	24.90	26.35	35.11	34.94	74.60	27.21	87.49
	✓	✓		23.54	25.00	31.26	39.76	73.00	24.26	86.73
	✓		✓	27.40	25.21	33.04	46.99	75.20	25.37	86.50
ACM-AIME		✓	✓	24.90	27.19	33.04	45.78	71.20	24.26	87.41
	✓	✓		24.17	23.85	33.78	40.96	70.80	24.63	86.96
	✓		✓	27.29	28.13	34.52	44.58	72.20	27.21	85.60

Table A4: Effect of data components on Qwen2.5-7B across seven benchmarks.

Seed	Stage1 Data	Stage2 Data	AIME25	AIME24	OlympiadBench	AMC23	MATH500	Minerva Math	GSM8K
MATH	Decompose	Original	23.44	23.65	32.89	37.35	72.20	26.10	87.04
	Original	Hybrid	24.90	28.02	36.00	43.37	76.60	27.94	86.50
	Decompose	Hybrid	26.35	26.25	37.78	44.58	77.00	28.68	87.49
AMC-AIME	Decompose	Original	24.79	24.69	35.11	36.14	71.40	26.10	89.08
	Original	Hybrid	26.46	29.58	36.15	46.99	75.00	26.10	86.73
	Decompose	Hybrid	23.96	27.81	36.89	49.40	76.00	28.68	88.10

Table A5: Effect of data curriculum learning on Qwen2.5-7B across seven benchmarks.