MathFusion: Enhancing Mathematical Problem-solving of LLM through Instruction Fusion

Qizhi Pei[1,2], Lijun Wu[2*], Zhuoshi Pan[2,3], Yu Li[2], Honglin Lin[2],
Chenlin Ming[2,4], Xin Gao[2], Conghui He[2*], Rui Yan[1,5,6*]

[1]Gaoling School of Artificial Intelligence, Renmin University of China
[2]Shanghai Artificial Intelligence Laboratory   [3]Tsinghua University   [4]Shanghai Jiao Tong University
[5]Engineering Research Center of Next-Generation Intelligent Search and Recommendation, MOE
[6]School of Artificial Intelligence, Wuhan University
{qizhipei,ruiyan}@ruc.edu.cn   {wulijun,heconghui}@pjlab.org.cn
🤗 https://huggingface.co/collections/QizhiPei/MathFusion
 https://github.com/QizhiPei/MathFusion

Abstract

Large Language Models (LLMs) have shown impressive progress in mathematical reasoning. While data augmentation is promising to enhance mathematical problem-solving ability, current approaches are predominantly limited to instance-level modifications—such as rephrasing or generating syntactic variations—which fail to capture and leverage the intrinsic relational structures inherent in mathematical knowledge. Inspired by human learning processes, where mathematical proficiency develops through systematic exposure to interconnected concepts, we introduce **MathFusion**, a novel framework that enhances mathematical reasoning through cross-problem instruction synthesis. MathFusion implements this through three fusion strategies: (1) *sequential fusion*, which chains related problems to model solution dependencies; (2) *parallel fusion*, which combines analogous problems to reinforce conceptual understanding; and (3) *conditional fusion*, which creates context-aware selective problems to enhance reasoning flexibility. By applying these strategies, we generate a new dataset, **MathFusionQA**, followed by fine-tuning models (DeepSeekMath-7B, Mistral-7B, Llama3-8B) on it. Experimental results demonstrate that MathFusion achieves substantial improvements in mathematical reasoning while maintaining high data efficiency, boosting performance by 18.0 points in accuracy across diverse benchmarks while requiring only 45K additional synthetic instructions, representing a substantial improvement over traditional single-instruction approaches.

---

*   Corresponding authors: Lijun Wu (wulijun@pjlab.org.cn), Conghui He (heconghui@pjlab.org.cn), and Rui Yan (ruiyan@ruc.edu.cn).
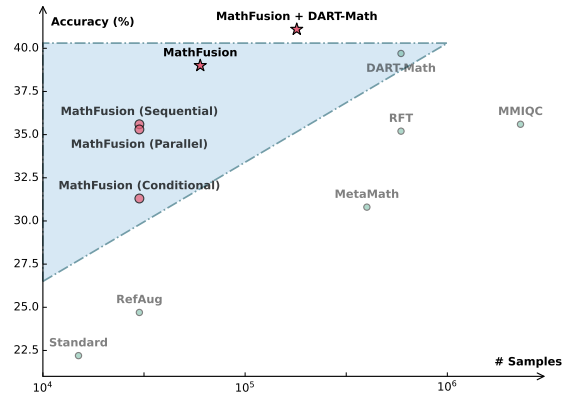
Figure 1: Average performance across six benchmarks of mathematical LLMs built on Llama3-8B, along with the respective # SFT samples. MathFusion yields superior performance with fewer synthetic instructions.

1 Introduction

Large Language Models (LLMs) have demonstrated remarkable capabilities in various reasoning tasks (Wei et al., 2022; Huang and Chang, 2023), with mathematical problem-solving emerging as a critical domain for assessing their cognitive abilities (Ahn et al., 2024). Specialized mathematical LLMs have emerged to address the unique challenges of solving complex mathematical problems (Yang et al., 2024; Shao et al., 2024; Ying et al., 2024; team, 2024). Current approaches to enhance mathematical reasoning primarily focus on four paradigms: continued pre-training with math corpora (Yang et al., 2024; Shao et al., 2024), reinforcement learning (RL) from human or automated feedback (Luo et al., 2023; Lu et al., 2024), test-time compute scaling (Wang et al., 2024a; Kang et al., 2024; Guan et al., 2025; Xi et al., 2024), and supervised fine-tuning (SFT) using problem-solution pairs (Tang et al., 2024; Tong et al., 2024; Lin et al., 2025). Among these, SFT is the most

widely adopted paradigm (Setlur et al., 2024) due to its simplicity. However, its effectiveness is often limited by the complexity and diversity of the mathematical training data (Luo et al., 2023) during SFT. To this end, data augmentation and synthesis have emerged as promising directions to enhance mathematical reasoning. For example, approaches such as MetaMath (Yu et al., 2024) and Wizard-Math (Luo et al., 2023) emphasize enhancing individual problems through rephrasing and difficulty variation.

While instance-level modifications have shown potential, they do not resolve the fundamental challenge: the inability of LLMs to effectively capture and leverage the intrinsic relational structures that characterize mathematical knowledge (Chu-Carroll et al., 2024; Srivatsa and Kochmar, 2024). This limitation becomes particularly apparent in real-world scenarios, where complex mathematical problems are often composed of interdependent sub-problems that form intricate dependency graphs (Bagherzadeh et al., 2019; Prabawa et al., 2023). For instance, solving a system of equations requires the sequential solution of individual equations, followed by the reconciliation of constraints.

Motivated by the way human learners develop proficiency through systematic exposure to interconnected ideas (Komarudin et al., 2021), we propose **MathFusion**, a novel framework that enhances mathematical reasoning by fusing different mathematical problems. The key insight behind MathFusion is that the strategic combination of complementary mathematical instructions can unlock deeper reasoning capabilities. Specifically, by combining two existing problems, MathFusion synthesizes a new math problem that encapsulates the relational and compositional aspects of the original two problems. To achieve this, we introduce three distinct fusion strategies: (1) *sequential fusion*, which links related problems by chaining them together through shared variables to model solution dependencies; (2) *parallel fusion*, which integrates analogy problems to enhance conceptual comprehension and generates a novel problem that encapsulates their shared mathematical essence; and (3) *conditional fusion*, which generates selective problems based on specific context to promote flexible reasoning.

Starting from existing datasets, we first identify pairs of problems that are suitable for fusion. Then we generate new problems by applying these fusion strategies to pairs of mathematical problems that share similar types and contexts. After that, we use strong LLMs to generate corresponding solutions. The resulting dataset, **MathFusionQA**, is then used to fine-tune LLMs including DeepSeekMath-7B, Mistral-7B, and Llama3-8B.

Experimental results demonstrate that Math-Fusion enables LLMs to effectively capture the underlying relational structures of mathematical tasks, thereby enhancing their capacity to resolve complex, multi-step problems. Moreover, MathFusion yields considerable improvements in mathematical reasoning accuracy across both in-domain and challenging out-of-domain benchmarks, outperforming traditional single-instruction fine-tuning by 18.0 points in accuracy on average while incorporating only 45K additional synthetic instructions. Further integration with the state-of-the-art (SOTA) data augmentation method DART-Math (Tong et al., 2024) and scaling MathFusion to larger size lead to additional improvements, surpassing DART-Math in accuracy on average while utilizing less than one-third of its data. This highlights the complementary nature and scalability of our approach.

2 Related Work

2.1 Individual Data Augmentation for Math

Existing mathematical data augmentation methods primarily focus on two aspects: *enhancing existing data* and *generating new data*. *Enhancing existing data* typically involves modifying the problem or solution. For the problem, strategies include altering the level of complexity/difficulty (Luo et al., 2023), rephrasing the wording (Yu et al., 2024; Li et al., 2024b), and employing backward reasoning (Yu et al., 2024). For the solution, methods such as generating diverse and high-quality mathematical reasoning paths through multiple calls (Yu et al., 2024; Li et al., 2024b; Zhang et al., 2024; Tong et al., 2024), and incorporating reflection (Zhang et al., 2024; Pan et al., 2025) are commonly used. *Generating new data* typically involves creating new mathematical problems based on key mathematical concepts (Tang et al., 2024), seed datasets (Ding et al., 2024), specific example (Li et al., 2024a), and then using strong mathematical models (OpenAI et al., 2023; Shao et al., 2024) to generate corresponding solutions. These methods, however, focus primarily on individual mathematical problems, overlooking the underlying relationships between different problems.
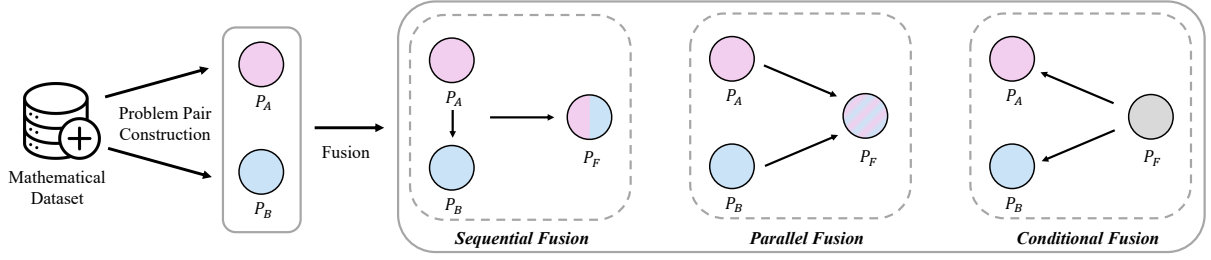
Figure 2: The overview of MathFusion. Given two mathematical problems P_A and P_B from the original mathematical dataset, MathFusion synthesizes a new mathematical problem P_F by fusing these two problems through three fusion strategies: *sequential fusion*, *parallel fusion*, and *conditional fusion*.

2.2 Compositional Data Augmentation

Most data augmentation methods focus on enhancing individual instances, while few consider the relationships between different instances. *mixup* (Zhang et al., 2018) is an augmentation technique that addresses this gap by generating synthetic training samples through linear interpolations between pairs of input data points and their corresponding labels, which has been shown to be effective across various tasks (Cao et al., 2025; Jin et al., 2024), such as image classification (Zhang et al., 2018; Thulasidasan et al., 2019), text classification (Guo et al., 2019; Zhang et al., 2020), and neural machine translation (Guo et al., 2020; Wu et al., 2021). Mosaic-IT (Li et al., 2024c) is a model-free data augmentation method that concatenates instruction data and trains LLMs with meta-instructions, thereby enhancing performance and reducing training costs. Some works also consider the composition of multiple skills or keypoints. Instruct-SkillMix (Kaur et al.) extracts core skills for instruction-following and generates new instructions by randomly combining pairs of skills. KPMath (Huang et al., 2024) shares the same idea with Instruct-SkillMix, but focuses on mathematical problems by extracting topics and key points from the problem and generates new problems by combining them.

In contrast to existing works, our approach primarily focuses on fusing mathematical problems and places particular emphasis on the logical coherence of the fusion.

3 MathFusion

The overview of MathFusion is shown in Figure 2. Given two mathematical problems P_A and P_B from the original mathematical training set, MathFusion synthesizes a new mathematical problem P_F by fusing these two problems. A simple example for P_A and P_B is shown in Example 3.1, and we show the corresponding P_F for three fusion

strategies in the following sections. More cases are shown in the Appendix G.

Example 3.1: Original Questions

P_A: During one day, there are 4 boat trips through the lake. The boat can take up to 12 people during one trip. How many people can the boat transport in 2 days?

P_B: The school is organizing a trip to the museum. 4 buses were hired to take the children and teachers to their destination. The second bus has twice the number of people on it as the first bus. The third bus has 6 fewer people than the second bus. The fourth bus has 9 more people than the first bus. If the first bus has 12 people, how many people are going to the museum in total?

In the following sections, we will first introduce the problem pair construction in Section 3.1, and then introduce the three fusion strategies: *sequential fusion* in Section 3.2, *parallel fusion* in Section 3.3, and *conditional fusion* in Section 3.4. Based on the augmented problem sets generated by these fusion strategies, we present the MathFusionQA dataset in Section 3.5.

3.1 Problem Pair Construction

To construct problem pairs for fusion, for each problem P_A, we need to identify a suitable problem P_B. A straightforward approach is to select a problem P_B that shares the same type and similar context with P_A. Formally, the problem pair set $\mathcal{D}_{\text{pair}}^{\text{p}}$ is defined as:

$$\mathcal{D}_{\text{pair}}^{\text{p}} = \left\{ (P_A, P_B) \mid P_A \in \mathcal{D}_{\text{train}}^{\text{p}}, P_B = \underset{P \in \mathcal{D}_{\text{train}}^{\text{p}} \setminus \{P_A\}}{\arg\max} \text{SIM}(P_A, P) \right\},$$

where $\mathcal{D}_{\text{train}}^{\text{p}}$ is a set of problems from the original training set, and $\text{SIM}(P_A, P_B)$ is the inner product of the embeddings of P_A and P_B using OpenAI embedding API *text-embedding-3-large* (OpenAI et al., 2023).

3.2 Sequential Fusion

In mathematical problem-solving, sequential reasoning is a common pattern where the solution

of the whole problem is the sequential combination of the solutions of the sub-problems. *Sequential fusion* constructs a new mathematical problem P_F^{seq} by establishing solution dependencies between two original problems P_A and P_B through shared variables, where the answer of P_A becomes a prerequisite for solving P_B. Formally, the *sequential fusion* process and the resulting augmented problem set are defined as:

$$P_F^{\text{seq}} = P_B(P_A), \mathcal{D}_{\text{seq}}^{\text{p}} = \{P_F^{\text{seq}} \mid (P_A, P_B) \in \mathcal{D}_{\text{pair}}^{\text{p}}\}.$$

The answer from solving P_A serves as a part of the input to P_B, thereby creating a chained dependency. A specific example of *sequential fusion* is shown in Example 3.2. The answer of P_A (the number of people transported by the boat) is used as the input for P_B (the number of people in the first bus).

Example 3.2: Sequential Fusion

P_F^{seq}: The school has organized a trip to a museum and needs to transport children and teachers. First, calculate how many people can be transported by a boat over 2 days, with 4 boat trips each day, and each trip can carry up to 12 people. Let this total be the number of people in the first bus. The second bus has twice the number of people on the first bus, the third bus has 6 fewer people than the second bus, and the fourth bus has 9 more people than the first bus. How many people are going to the museum in total?

3.3 Parallel Fusion

Analogous problems often share common mathematical concepts and essences. *Parallel fusion* leverages this by synthesizing P_F^{para} through the integration of two conceptually analogous problems P_A and P_B, thereby creating a new problem that encapsulates their shared mathematical essence. This approach emphasizes the conceptual relationships between problems rather than their sequential dependencies. The *parallel fusion* process and the resulting augmented problem set are formally defined as:

$$P_A \to P_A', P_B \to P_B', P_F^{\text{para}} = \Phi(P_A', P_B'),$$
$$\mathcal{D}_{\text{para}}^{\text{p}} = \{P_F^{\text{para}} \mid (P_A, P_B) \in \mathcal{D}_{\text{pair}}^{\text{p}}\},$$

where P_A' and P_B' denote the potentially modified problems from P_A and P_B, respectively, for the fused problem P_F^{para}. The function Φ encompasses various operations, such as algebraic composition and the enforcement of constraint satisfiability, to rigorously integrate the underlying mathematical structures. A concrete illustration of *parallel fusion* is provided in Example 3.3. The total number of people transported by boat and buses over 2 days is asked to be calculated, and the input of P_A' (the

number of trips made by the boat in one day) is different from that of P_A.

Example 3.3: Parallel Fusion

P_F^{para}: A school organizes a field trip to a museum and hires 4 buses and a boat. The boat makes 2 trips in one day, with a capacity of 12 people per trip. Each bus has a different number of people: the first bus bus has 12 people ... the fourth bus has 9 more people than the first bus. Calculate the total number of people transported by the boat and the buses over the course of 2 days. How many people can the boat and buses transport in total for the trip?

3.4 Conditional Fusion

Context-aware reasoning necessitates the dynamic selection or comparison of solutions based on conditional constraints. *Conditional fusion* synthesizes P_F^{cond} by integrating P_A and P_B into a cohesive real-world scenario, where the final solution is derived through contextual comparison or selection of outcomes from P_A and P_B. Formally, the *conditional fusion* process and the resulting augmented problem set are defined as:

$$P_F^{\text{cond}} = \Gamma(P_A, P_B), \mathcal{D}_{\text{cond}}^{\text{p}} = \{P_F^{\text{cond}} \mid (P_A, P_B) \in \mathcal{D}_{\text{pair}}^{\text{p}}\}.$$

Γ is a comparison function that contrasts P_A and P_B based on predefined logical or contextual rules. A concrete case is shown in Example 3.4, where the final solution is determined by comparing the answers of P_A (the capacity of the boat) and P_B (the capacity of the buses) in a real-world scenario (organizing a lake excursion and a museum trip).

Example 3.4: Conditional Fusion

P_F^{cond}: A local community is organizing two different outings. For a lake excursion, a boat operates 4 trips a day with a capacity of 12 people per trip. They plan to run this boat service for 2 days. Meanwhile, a school is arranging a trip to the museum with 4 buses. The first bus has 12 people, the second bus has twice as many people as the first, the third bus has 6 fewer people than the second, and the fourth bus has 9 more people than the first bus. Given these arrangements, which mode of transportation has a larger capacity for transporting people?

To clarify, the core difference between *parallel fusion* and *conditional fusion* is that: *parallel fusion* combines P_A and P_B to form a novel P_F^{para}, where the input of P_F^{para} may be different from the original P_A and P_B; while *conditional fusion* compares the results of P_A and P_B, the input of P_F^{cond} is the same as P_A and P_B, and the output is based on the comparison of the results of P_A and P_B.

3.5 MathFusionQA Dataset

After applying the three fusion strategies to $\mathcal{D}_{\text{pair}}^{\text{p}}$ and get the augmented problem sets $\mathcal{D}_{\text{seq}}^{\text{p}}, \mathcal{D}_{\text{para}}^{\text{p}}$,

Dataset	# Samples
WizardMath (Luo et al., 2023)	96K
MetaMathQA (Yu et al., 2024)	395K
MMIQC (Liu et al., 2024)	2294K
Orca-Math (Mitra et al., 2024)	200K
Xwin-Math-V1.1 (Li et al., 2024a)	1440K
KPMath-Plus (Huang et al., 2024)	1576K
MathScaleQA (Tang et al., 2024)	2021K
DART-Math-Uniform (Tong et al., 2024)	591K
DART-Math-Hard (Tong et al., 2024)	585K
RefAug (Zhang et al., 2024)	30K
MathFusionQA	60K

Table 1: Comparison between MathFusionQA and previous mathematical datasets. Our MathFusionQA is generally smaller than others.

and $\mathcal{D}_{\text{cond}}^{\text{p}}$, we use GPT-4o-mini (OpenAI et al., 2023) to generate corresponding solutions S for the augmented problems. The resultingaugmented data $\mathcal{D}_{\text{seq}}$, $\mathcal{D}_{\text{para}}$, and $\mathcal{D}_{\text{cond}}$ are combined with the original training set $\mathcal{D}_{\text{train}}$ to form the final MathFusionQA dataset as $\mathcal{D}_{\text{MathFusionQA}} = \mathcal{D}_{\text{train}} \cup \mathcal{D}_{\text{seq}} \cup \mathcal{D}_{\text{para}} \cup \mathcal{D}_{\text{cond}}$. We use GSM8K (Cobbe et al., 2021) and MATH (Hendrycks et al., 2021) as the original training set separately. We compare our MathFusionQA with other mathematical datasets in Table 1. Though MathFusionQA is overall smaller than other datasets except for RefAug (Zhang et al., 2024), we empirically show that MathFusionQA exhibits strong performance and is more effective than RefAug in Section 4.2. Then we fine-tune LLMs on the MathFusionQA dataset, resulting in MathFusion models. Given that some problems in MathFusionQA may be incomplete or incorrect, we conduct an analysis of problem evaluation and error correction, and present cases of unsuitable fusions in Appendix C.2.

4 Experiments

4.1 Experimental Setup

Data Synthesis: We use GPT-4o-mini (OpenAI et al., 2023) to fuse the problems and generate the corresponding solutions. The details about generation and corresponding prompts are shown in Appendix B.1 and A).

Training: We conduct standard instruction-tuning on our MathFusionQA. Following DART-Math (Tong et al., 2024), we conduct experiments on two categories of base models: 7B math-specialized base LLM, specifically DeepSeekMath-7B (Shao et al., 2024), and 7-8B general base

LLMs, specifically Mistral-7B (Jiang et al., 2023) and Llama3-8B (Dubey et al., 2024). Each base model is fine-tuned using three distinct fusion strategies: *sequential*, *parallel*, and *conditional*. For each strategy, the fine-tuning dataset (30K samples) comprises the union of the GSM8K, MATH datasets, and an augmented set generated by the respective fusion strategy. The MathFusionQA dataset (60K samples) is formed by the union of all these sub-datasets. Table 4 shows the statistics of the MathFusionQA collection. To demonstrate the scaling ability of MathFusion, we also enlarge MathFusionQA dataset by including top-2 to top-4 nearest neighbors, resulting in a total of 15K + 4 × (3 × 15K) = 195K examples. All models are trained for 3 epochs. More details about the training setup are provided in Appendix B.2.

Evaluation: Following DART-Math (Tong et al., 2024), we evaluate the models on two **in-domain** (ID) benchmarks: GSM8K (Cobbe et al., 2021) and MATH (Hendrycks et al., 2021), as our MathFusionQA dataset is built upon these two datasets. For **out-of-domain** (OOD) evaluation, we use the CollegeMath (Tang et al., 2024), DeepMind-Mathematics (Saxton et al., 2019), OlympiadBench-Math (He et al., 2024), and TheoremQA (Chen et al., 2023) benchmarks. We use greedy decoding to generate solutions for the problems in test sets. We report the accuracy in 0-shot setting for all models following Tong et al. (2024). More details about the evaluation setup and benchmarks are provided in the Appendix B.3.

Baselines: We mainly compare our MathFusion models with mathematical instruction-based models, which can be categorized into three groups: (1) Previous top-performing models, including MetaMath (Yu et al., 2024), WizardMath (Luo et al., 2023), RFT (rejection sampling fine-tuning from DART-Math) (Yuan et al., 2023; Tong et al., 2024), MMIQC (Liu et al., 2024), MathScale (Tang et al., 2024), DeepSeekMath-7B-Instruct (Shao et al., 2024), RefAug (Zhang et al., 2024), and DART-Math (Tong et al., 2024) (we report the *Prop2Diff* version as it generally performs better than the *Uniform* version); (2) Models instruction-tuned on the combination of GSM8K and MATH datasets (noted as "Standard" setting); (3) Models instruction-tuned on the sampled 60K version of previous top-performing methods to further evaluate the data efficiency of different mathematical data augmentation methods. Details about the sampling method are introduced in Appendix B.2.

Model	# Samples	In-Domain		Out-of-Domain				
		MATH	GSM8K	College	DM	Olympiad	Theorem	AVG
DeepSeekMath (7B Math-Specialized Base Model)								
DeepSeekMath-7B-RFT	590K	53.0	**88.2**	**41.9**	60.2	19.1	27.2	48.3
DeepSeekMath-7B-DART-Math	590K	53.6	86.8	40.7	61.6	21.7	**32.2**	49.4
DeepSeekMath-7B-Instruct	780K	46.9	82.7	37.1	52.2	14.2	28.1	43.5
DeepSeekMath-7B-MMIQC	2.3M	45.3	79.0	35.3	52.9	13.0	23.4	41.5
MathFusion-DSMath-7B	195K	**58.2**	79.5	40.3	**69.1**	25.5	27.0	**49.9**
DeepSeekMath-7B-Standard	15K	30.6	66.3	22.7	28.6	5.6	11.0	27.5
DeepSeekMath-7B-RefAug	30K	32.1	71.2	26.0	38.4	10.1	14.4	32.0
MathFusion-DSMath-7B (*Sequential*)	30K	49.9	76.6	38.8	64.6	21.6	22.8	45.7
MathFusion-DSMath-7B (*Parallel*)	30K	50.9	76.7	38.9	62.2	19.0	23.8	45.3
MathFusion-DSMath-7B (*Conditional*)	30K	48.5	74.6	37.0	55.2	19.3	19.0	42.3
DeepSeekMath-7B-MetaMath[†]	60K	40.0	79.0	33.2	45.9	9.5	18.9	37.8
DeepSeekMath-7B-MMIQC[†]	60K	26.3	60.6	19.2	41.5	10.4	6.8	27.5
DeepSeekMath-7B-RefAug[†]	60K	33.1	71.6	26.2	35.4	10.5	14.0	31.8
DeepSeekMath-7B-DART-Math[†]	60K	51.4	**82.9**	39.1	62.8	21.0	**27.4**	47.4
MathFusion-DSMath-7B	60K	**53.4**	77.9	**39.8**	65.8	23.3	24.6	**47.5**
Mistral-7B (7-8B General Base Model)								
Mistral-7B-MetaMath	400K	29.8	76.5	19.3	28.0	5.9	14.0	28.9
Mistral-7B-WizardMath-V1.1	418K	32.3	80.4	23.1	38.4	7.7	16.6	33.1
Mistral-7B-RFT	590K	38.7	**82.3**	24.2	35.6	8.7	16.2	34.3
Mistral-7B-DART-Math	590K	**45.5**	81.1	**29.4**	**45.1**	14.7	**17.0**	**38.8**
Mistral-7B-MathScale	2.0M	35.2	74.8	21.8	–	–	–	–
Mistral-7B-MMIQC	2.3M	37.4	75.4	28.5	38.0	9.4	16.2	34.2
Mistral-7B-Standard	15K	12.4	60.3	8.4	17.0	2.2	7.6	18.0
Mistral-7B-RefAug	30K	15.1	61.1	10.4	15.4	3.1	11.0	19.4
MathFusion-Mistral-7B (*Sequential*)	30K	32.7	73.9	18.9	29.3	9.3	15.5	29.9
MathFusion-Mistral-7B (*Parallel*)	30K	30.9	75.1	20.9	26.5	11.0	15.2	29.9
MathFusion-Mistral-7B (*Conditional*)	30K	26.3	73.0	15.6	21.4	7.3	12.8	26.1
Mistral-7B-MetaMath[†]	60K	22.7	70.8	14.1	27.2	5.0	12.2	25.3
Mistral-7B-MMIQC[†]	60K	17.3	61.4	11.1	13.5	5.0	5.9	19.0
Mistral-7B-RefAug[†]	60K	17.4	63.1	12.5	18.1	3.9	11.1	21.0
Mistral-7B-DART-Math[†]	60K	34.1	77.2	23.4	36.0	8.7	**18.2**	32.9
MathFusion-Mistral-7B	60K	**41.6**	**79.8**	**24.3**	**39.2**	13.6	18.1	**36.1**
Llama3-8B (7-8B General Base Model)								
Llama3-8B-MetaMath	400K	32.5	77.3	20.6	35.0	5.5	13.8	30.8
Llama3-8B-RFT	590K	39.7	**81.7**	23.9	41.7	9.3	14.9	35.2
Llama3-8B-MMIQC	2.3M	39.5	77.6	**29.5**	41.0	9.6	16.2	35.6
Llama3-8B-DART-Math	590K	**46.6**	81.1	28.8	**48.0**	14.5	**19.4**	**39.7**
Llama3-8B-Standard	15K	17.5	65.4	12.9	21.6	4.7	10.9	22.2
Llama3-8B-RefAug	30K	20.8	67.3	15.7	25.9	4.7	13.6	24.7
MathFusion-Llama3-8B (*Sequential*)	30K	38.8	77.9	25.1	42.0	12.6	17.0	35.6
MathFusion-Llama3-8B (*Parallel*)	30K	38.1	75.4	25.5	41.9	11.9	18.9	35.3
MathFusion-Llama3-8B (*Conditional*)	30K	34.7	76.9	21.2	27.4	11.9	15.5	31.3
Llama3-8B-MetaMath[†]	60K	28.7	78.5	19.7	31.3	5.3	16.1	29.9
Llama3-8B-MMIQC[†]	60K	24.4	69.7	13.4	30.9	5.2	10.6	25.7
Llama3-8B-RefAug[†]	60K	20.3	68.6	15.5	29.1	5.5	13.0	25.3
Llama3-8B-DART-Math[†]	60K	39.6	**82.2**	27.9	39.9	12.9	**22.9**	37.6
MathFusion-Llama3-8B	60K	**46.5**	79.2	27.9	**43.4**	17.2	20.0	**39.0**

Table 2: Performance comparison on mathematical benchmarks including MATH, GSM8K, CollegeMATH (College), DeepMind-Mathematics (DM), OlympiadBench-Math (Olympiad), and TheoremQA (Theorem). The table is organized by the base model and the number of training samples, using 60K as the threshold for splitting. The best results are highlighted in bold. Rows are sorted according to data size. Most of the baseline results are derived from DART-Math (Tong et al., 2024), except for the Standard, RefAug (Zhang et al., 2024), and baseline labeled with [†], which are our own runs. *Sequential*, *Parallel*, and *Conditional* indicate training on the union of GSM8K, MATH, and the respective fused dataset.

4.2 Main Results

The main results are shown in Table 2. We summarize several key findings as follows:

Finding 1: **Three fusion strategies consistently enhance the model performance.** For all three fusion strategies-*sequential*, *parallel*, and *conditional fusion*—the MathFusion models consistently surpass the standard settings across all base models and evaluation benchmarks. Specifically, on MATH and GSM8K test sets, using Llama3-8B as the base model, MathFusion (*sequential*) achieves 21.3 and 12.5 accuracy improvement; MathFusion (*parallel*) achieves 20.6 and 10.0 accuracy improvement; and MathFusion (*conditional*) achieves 18.0 and 11.9 accuracy improvement, respectively, compared to the standard setting. For four OOD benchmarks, the single fusion strategy also outperforms the standard setting, with a 9.9 accuracy improvement on average. These improvements demonstrate the effectiveness of the three fusion strategies in enhancing both the ID and OOD generalization performance of the models.

Finding 2: **Among three fusion strategies, *sequential fusion* and *parallel fusion* generally perform better than *conditional fusion*.** A possible reason is that the *conditional fusion* requires no modification of input structures or problem dependencies, merely performing a direct comparison or selection between the solutions of two independent problems without necessitating additional mathematical transformations or reformulations. We further investigate the difficulty of the problems generated by the three fusion strategies in Section 5.1.

Finding 3: **Combination of three fusion strategies further improves performance.** As the three fusion strategies capture different aspects of the problem fusion, we further investigate the performance of the combined fusion strategies. From Table 2, we observe that the combined fusion strategies consistently outperform each single fusion strategy, indicating that the combination of three fusion strategies can further enhance the model's mathematical ability. Additionally, the weaker the performance of the base model, the more enhancements the combined fusion strategies can bring. Specifically, the combined fusion strategies achieve an average accuracy improvement of 3.1 points on DeepSeekMath-7B, 4.9 points on Llama3-8B, and 7.5 points on Mistral-7B across all benchmarks.

Finding 4: **Compared with previous top-performing baselines, MathFusion models yields**

Method	Sequential	Parallel	Conditional	MATH	GSM8K
Standard	✗	✗	✗	17.5	65.4
MathFusion	✗	✓	✓	42.6	78.2
	✓	✗	✓	43.0	76.9
	✓	✓	✗	43.6	79.2
	✓	✓	✓	**45.6**	**79.9**

Table 3: Effect of three fusion strategies on Llama3-8B.

competitive performance and high data efficiency. For each single fusion strategy, MathFusion models outperform RefAug, which has the same data size as MathFusion, on all benchmarks. After combining the three fusion strategies, MathFusion outperforms previous top-performing baselines like MetaMath and DART-Math on average under the same data size setting. Specifically, MathFusion yields consistently better performance on MATH, DeepMind-Mathematics, and OlympiadBench-Math benchmarks. These results demonstrate the high data efficiency and generalization ability of MathFusion. MathFusion maintains also competitive efficacy compared to top-performing models in the full-data regime, exhibiting only a marginal average performance drop on Llama3-8B and DeepSeekMath-7B.

Finding 5: **MathFusion exhibits strong scalability and outperforms larger-scale baselines with fewer samples.** The results on DeepSeekMath-7B (the strongest base model for math) in Table 2 reveal that scaling MathFusion from 60K to 195K samples leads to consistent performance gains across all evaluation benchmarks. Notably, MathFusion-DSMath-7B (195K) surpasses DeepSeekMath-7B-DART-Math (590K) in average accuracy (49.9 vs. 49.4), despite using only one-third of the training data. This illustrates the high scalability and data efficiency of MathFusion. The substantial performance gains on challenging benchmarks such as MATH (+4.6), DeepMind-Mathematics (+7.5), and OlympiadBench-Math (+3.8) underscore the method's capability to generalize well under increased data volume. These findings demonstrate that MathFusion benefits substantially from larger synthetic training sets and can outperform significantly larger instruction-tuned models with less data.

4.3 Ablation Study

We further conduct an ablation study to investigate the contribution of each fusion strategy to the overall performance of combined fusion. The results over Llama3-8B on MATH and GSM8K are
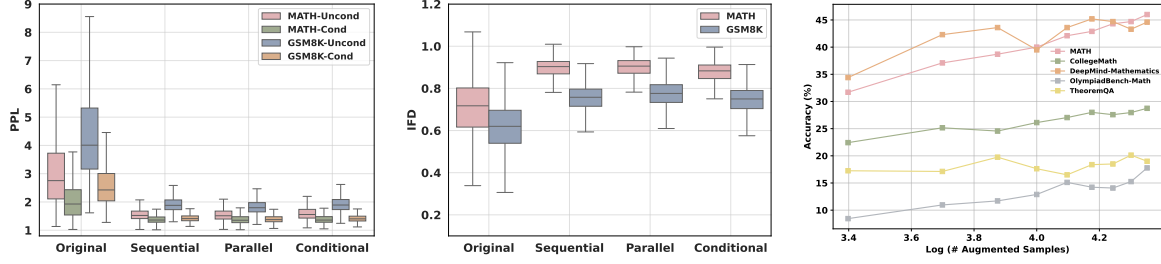
Figure 3: **(a)**: Unconditional and conditional PPL for the original and fused data on GSM8K and MATH datasets. **(b)**: IFD for the original and fused data on GSM8K and MATH datasets. **(c)**: Performance scaling behavior of the MathFusion on different sizes of augmented data on Llama3-8B.
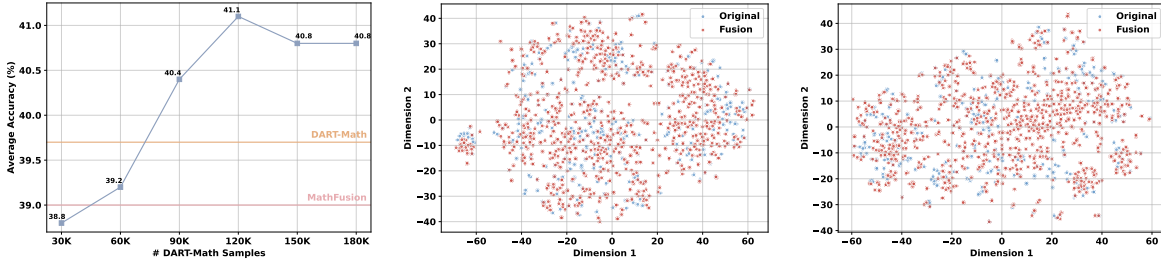

Figure 4: **(a)**: Average performance of the Llama3-8B models fine-tuned on the combined dataset of MathFusionQA and DART-Math-Hard with different sizes of sampled data. **(b) and (c)**: Problem embedding visualization for GSM8K and MATH datasets via t-SNE.

shown in Table 3, from which we observe that each fusion strategy contributes to the overall performance, with *conditional fusion* showing the least contribution, which aligns with Section 4.2. We further ablate on choice of teacher model for solution generation in Appendix D.

5 Analysis

We analyze the difficulty of the fused problem in Section 5.1, the relationship between augmented data size and performance in Section 5.2, the combination of MathFusionQA with other datasets in Section 5.3, and the diversity of fused problem in Section 5.4.

5.1 Difficulty Analysis

In this section, we explore why the three fusion strategies effectively enhance the model's performance. To achieve this, we evaluate both the perplexity (PPL) and instruction following difficulty (IFD) (Li et al., 2024d) for the original and fused data. We use Mathstral-7B (team, 2024), a model built upon Mistral-7B (Jiang et al., 2023) and specifically fine-tuned for mathematical reasoning, to ensure our analysis relies on a model specifically designed for mathematical tasks. Specifi-

cally, we denote the unconditioned PPL as $\mathrm{PPL}(S)$, the conditioned PPL as $\mathrm{PPL}(S \mid P)$, and $\mathrm{IFD} = \mathrm{PPL}(S \mid P)/\mathrm{PPL}(S)$, where P is the problem and S is the solution. The results are shown in Figure 3(a) and 3(b), from which we can see: (1) The PPL of the solution of the fused problems is significantly lower than that of the original problems. As analyzed in Yu et al. (2024), this may be due to the easy-to-learn nature of the generated solutions. (2) The IFD of the fused data is significantly higher than that of the original data, indicating that the fused data is more difficult to learn in the context of the problem. (3) The IFD of the MATH datasets, both the original or fused version, are higher than that of the GSM8K, consistent with the fact that MATH is generally more difficult than GSM8K.

5.2 Relationship between Augmented Data Size and Performance

We study the performance scaling behavior of the MathFusion on different sizes of augmented data on Llama3-8B. We select MATH as the original training set and gradually increase the size of the augmented fusion data from 0 to 22.5K, with a step size of 2.5K. The results on MATH and four OOD benchmarks are shown in Figure 3(c). We ob-

serve that the performance of the MathFusion models exhibits an approximate logarithmic growth with respect to the amount of augmented data, which is consistent with the findings in (Li et al., 2024b). Additionally, the augmented fusion data from MATH dataset can also generalize better to the OOD benchmarks as the size of the augmented data increases. In summary, the MathFusion shows consistent performance improvement with different sizes of augmented data.

5.3 Combination with Other Datasets

We further investigate the performance of MathFusion when combined with other data augmentation methods. Specifically, we downsample 30K-180K data from DART-Math-Hard (Tong et al., 2024), which is the SOTA method for mathematical data augmentation with 590K data. We combine the downsampled DART-Math-Hard with our MathFusionQA dataset and fine-tune Llama3-8B models on the combined dataset. The results are presented in Figure 4(a). As the size of sampled data increases, the average performance of the models also increases, and reaches the peak when the size of the sampled data is 120K. Notably, by only using 90K data sampled from DART-Math-Hard (*i.e.*, 150K samples in total), the resulting model achieves better performance than both DART-Math and Math-Fusion, yields SOTA average performance. These results show the potential of combining MathFusion with other data augmentation methods to further enhance the model's performance. We think that the enhancement arises from the complementary and orthogonal nature of the two methods: our MathFusion emphasizes fusing mathematical problems to generate more challenging and diverse problems, while DART-Math focuses on existing difficult problems and primarily generates additional solutions for them.

5.4 Diversity Analysis

To further investigate the effectiveness of the Math-Fusion in enhancing the data diversity, we visualize the problem embeddings of the GSM8K and MATH datasets generated by GPT-4o-mini using t-SNE (Van der Maaten and Hinton, 2008). The results are shown in Figure 4(b) and 4(c). We can observe that the MathFusion augmented problems are more evenly distributed in the embedding space, thereby enriching the diversity of the training examples and mitigating the risk of model overfitting.

6 Conclusion

In this paper, we focus on the fusion of mathematical problems. We propose a novel mathematical data augmentation method, MathFusion, which comprises three distinct fusion strategies—*sequential fusion*, *parallel fusion*, and *conditional fusion*—designed to synthesize augmented mathematical problems. Leveraging these fusion strategies, we construct the MathFusionQA dataset, which is subsequently employed to fine-tune LLMs. Extensive experiments on three base models and six benchmarks show that MathFusion exhibits robust performance in both the in-domain and out-of-domain benchmarks while maintaining high data efficiency.

Limitations

We utilize GPT-4o-mini to generate fused problems and solutions, but the generated problems or solutions may still contain errors or ambiguities, which are hard to detect and verify. The quality of the generated problems and solutions is limited by the capabilities of the teacher LLM. Stronger teacher model, like DeepSeek-R1 and Qwen3, are underexplored. We mainly explore the effectiveness of the three fusion strategies on problem pairs that are constructed by embedding similarity. The fusion of three or more problems and more effective ways to find similar problems, remain underexplored. The released MathFusionQA dataset currently contains only 60K examples, and scaling to millions of examples remains underexplored.

Acknowledgements

References

Janice Ahn, Rishu Verma, Renze Lou, Di Liu, Rui Zhang, and Wenpeng Yin. 2024. Large language

models for mathematical reasoning: Progresses and challenges. In *EACL (Student Research Workshop)*, pages 225–237. Association for Computational Linguistics.

Mehdi Bagherzadeh, Andrei Gurca, and Sabine Brunswicker. 2019. Problem types and open innovation governance modes: A project-level empirical exploration. *IEEE Transactions on Engineering Management*, 69(2):287–301.

Chengtai Cao, Fan Zhou, Yurou Dai, Jianping Wang, and Kunpeng Zhang. 2025. A survey of mix-based data augmentation: Taxonomy, methods, applications, and explainability. *ACM Comput. Surv.*, 57(2):37:1–37:38.

Wenhu Chen, Ming Yin, Max Ku, Pan Lu, Yixin Wan, Xueguang Ma, Jianyu Xu, Xinyi Wang, and Tony Xia. 2023. TheoremQA: A theorem-driven question answering dataset. In *The 2023 Conference on Empirical Methods in Natural Language Processing*.

Jennifer Chu-Carroll, Andrew Beck, Greg Burnham, David OS Melville, David Nachman, A Erdem Özcan, and David Ferrucci. 2024. Beyond llms: Advancing the landscape of complex reasoning. *arXiv preprint arXiv:2402.08064*.

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

Yuyang Ding, Xinyu Shi, Xiaobo Liang, Juntao Li, Qiaoming Zhu, and Min Zhang. 2024. Unleashing reasoning capability of llms via scalable question synthesis from scratch. *arXiv preprint arXiv:2410.18693*.

Abhimanyu Dubey, Abhinav Jauhri, Abhinav Pandey, Abhishek Kadian, Ahmad Al-Dahle, Aiesha Letman, Akhil Mathur, Alan Schelten, Amy Yang, Angela Fan, et al. 2024. The llama 3 herd of models. *arXiv 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*.

Demi Guo, Yoon Kim, and Alexander M. Rush. 2020. Sequence-level mixed sample data augmentation. In *EMNLP (1)*, pages 5547–5552. Association for Computational Linguistics.

Hongyu Guo, Yongyi Mao, and Richong Zhang. 2019. Augmenting data with mixup for sentence classification: An empirical study. *arXiv preprint arXiv:1905.08941*.

Chaoqun He, Renjie Luo, Yuzhuo Bai, Shengding Hu, Zhen Leng Thai, Junhao Shen, Jinyi Hu, Xu Han,

Yujie Huang, Yuxiang Zhang, et al. 2024. Olympiad-bench: A challenging benchmark for promoting agi with olympiad-level bilingual multimodal scientific problems. *arXiv preprint arXiv:2402.14008*.

Dan Hendrycks, Collin Burns, Saurav Kadavath, Akul Arora, Steven Basart, Eric Tang, Dawn Song, and Jacob Steinhardt. 2021. Measuring mathematical problem solving with the math dataset. In *Proceedings of the Neural Information Processing Systems Track on Datasets and Benchmarks*, volume 1.

Jie Huang and Kevin Chen-Chuan Chang. 2023. Towards reasoning in large language models: A survey. In *ACL (Findings)*, pages 1049–1065. Association for Computational Linguistics.

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, Chris Bamford, Devendra Singh Chaplot, Diego de las Casas, Florian Bressand, Gianna Lengyel, Guillaume Lample, Lucile Saulnier, Lélio Renard Lavaud, Marie-Anne Lachaux, Pierre Stock, Teven Le Scao, Thibaut Lavril, Thomas Wang, Timothée Lacroix, and William El Sayed. 2023. Mistral 7b. *Preprint*, arXiv:2310.06825.

Xin Jin, Hongyu Zhu, Siyuan Li, Zedong Wang, Zicheng Liu, Chang Yu, Huafeng Qin, and Stan Z Li. 2024. A survey on mixup augmentations and beyond. *arXiv preprint arXiv:2409.05202*.

Jikun Kang, Xin Zhe Li, Xi Chen, Amirreza Kazemi, Qianyi Sun, Boxing Chen, Dong Li, Xu He, Quan He, Feng Wen, et al. 2024. Mindstar: Enhancing math reasoning in pre-trained llms at inference time. *arXiv preprint arXiv:2405.16265*.

Simran Kaur, Simon Park, Anirudh Goyal, and Sanjeev Arora. Instruct-skillmix: A powerful pipeline for llm instruction tuning. In *NeurIPS 2024 Workshop on Fine-Tuning in Modern Machine Learning: Principles and Scalability*.

Komarudin Komarudin, Suherman Suherman, and Anita Anggraini. 2021. Analysis of mathematical concept understanding capabilities: The impact of makerspae stem learning approach models and student learning activities. *Journal of Innovation in Educational and Cultural Research*, 2(1):35–43.

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*.

Chengpeng Li, Zheng Yuan, Hongyi Yuan, Guanting Dong, Keming Lu, Jiancan Wu, Chuanqi Tan, Xiang Wang, and Chang Zhou. 2024b. Mugglemath:

Assessing the impact of query and response augmentation on math reasoning. In *ACL (1)*, pages 10230–10258. Association for Computational Linguistics.

Ming Li, Pei Chen, Chenguang Wang, Hongyu Zhao, Yijun Liang, Yupeng Hou, Fuxiao Liu, and Tianyi Zhou. 2024c. Mosaic it: Enhancing instruction tuning with data mosaics. *arXiv preprint arXiv:2405.13326*.

Ming Li, Yong Zhang, Zhitao Li, Jiuhai Chen, Lichang Chen, Ning Cheng, Jianzong Wang, Tianyi Zhou, and Jing Xiao. 2024d. From quantity to quality: Boosting LLM performance with self-guided data selection for instruction tuning. In *NAACL-HLT*, pages 7602–7635. Association for Computational Linguistics.

Honglin Lin, Zhuoshi Pan, Yu Li, Qizhi Pei, Xin Gao, Mengzhang Cai, Conghui He, and Lijun Wu. 2025. Metaladder: Ascending mathematical solution quality via analogical-problem reasoning transfer. *arXiv preprint arXiv:2503.14891*.

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

Zimu Lu, Aojun Zhou, Ke Wang, Houxing Ren, Weikang Shi, Junting Pan, Mingjie Zhan, and Hongsheng Li. 2024. Step-controlled dpo: Leveraging stepwise error for enhanced mathematical reasoning. *arXiv preprint arXiv:2407.00782*.

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*.

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*.

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

Zhuoshi Pan, Yu Li, Honglin Lin, Qizhi Pei, Zinan Tang, Wei Wu, Chenlin Ming, H Vicky Zhao, Conghui He, and Lijun Wu. 2025. Lemma: Learning from errors for mathematical advancement in llms. *arXiv preprint arXiv:2503.17439*.

Harsa Wara Prabawa, Rizky Rosjanuardi, and Elah Nurlaelah. 2023. Problem decomposition skills, mathematical maturity, and their relation to mathematics problem-solving in a computer science learning class. *Jurnal Kependidikan: Jurnal Hasil Penelitian dan Kajian Kepustakaan di Bidang Pendidikan, Pengajaran dan Pembelajaran*, 9(3):946–958.

David Saxton, Edward Grefenstette, Felix Hill, and Pushmeet Kohli. 2019. Analysing mathematical reasoning abilities of neural models. In *International Conference on Learning Representations*.

Amrith Setlur, Saurabh Garg, Xinyang Geng, Naman Garg, Virginia Smith, and Aviral Kumar. 2024. Rl on incorrect synthetic data scales the efficiency of llm math reasoning by eight-fold. *arXiv preprint arXiv:2406.14532*.

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

KV Aditya Srivatsa and Ekaterina Kochmar. 2024. What makes math word problems challenging for llms? In *NAACL-HLT (Findings)*, pages 1138–1148. Association for Computational Linguistics.

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

Mistral AI team. 2024. Learning to reason with llms.

Sunil Thulasidasan, Gopinath Chennupati, Jeff A. Bilmes, Tanmoy Bhattacharya, and Sarah Michalak. 2019. On mixup training: Improved calibration and predictive uncertainty for deep neural networks. In *NeurIPS*, pages 13888–13899.

Yuxuan Tong, Xiwen Zhang, Rui Wang, Ruidong Wu, and Junxian He. 2024. Dart-math: Difficulty-aware rejection tuning for mathematical problem-solving. In *NeurIPS*.

Shubham Toshniwal, Wei Du, Ivan Moshkov, Branislav Kisacanin, Alexan Ayrapetyan, and Igor Gitman. Openmathinstruct-2: Accelerating ai for math with massive open-source instruction data. In *The Thirteenth International Conference on Learning Representations*.

Laurens Van der Maaten and Geoffrey Hinton. 2008. Visualizing data using t-sne. *Journal of machine learning research*, 9(11).

Ke Wang, Houxing Ren, Aojun Zhou, Zimu Lu, Sichun Luo, Weikang Shi, Renrui Zhang, Linqi Song, Mingjie Zhan, and Hongsheng Li. 2024a. Mathcoder: Seamless code integration in llms for enhanced mathematical reasoning. In *ICLR*. OpenReview.net.

Ming Wang, Yuanzhong Liu, Xiaoyu Liang, Songlian Li, Yijie Huang, Xiaoming Zhang, Sijia Shen, Chaofeng Guan, Daling Wang, Shi Feng, et al. 2024b. Langgpt: Rethinking structured reusable prompt design framework for llms from the programming language. *arXiv preprint arXiv:2402.16929*.

Jason Wei, Xuezhi Wang, Dale Schuurmans, Maarten Bosma, Brian Ichter, Fei Xia, Ed H. Chi, Quoc V. Le, and Denny Zhou. 2022. Chain-of-thought prompting elicits reasoning in large language models. In *NeurIPS*.

Xueqing Wu, Yingce Xia, Jinhua Zhu, Lijun Wu, Shufang Xie, Yang Fan, and Tao Qin. 2021. mixseq: A simple data augmentation methodfor neural machine translation. In *IWSLT*, pages 192–197. Association for Computational Linguistics.

Zhiheng Xi, Dingwen Yang, Jixuan Huang, Jiafu Tang, Guanyu Li, Yiwen Ding, Wei He, Boyang Hong, Shihan Do, Wenyu Zhan, et al. 2024. Enhancing llm reasoning via critique models with test-time and training-time supervision. *arXiv preprint arXiv:2411.16579*.

An Yang, Beichen Zhang, Binyuan Hui, Bofei Gao, Bowen Yu, Chengpeng Li, Dayiheng Liu, Jianhong Tu, Jingren Zhou, Junyang Lin, et al. 2024. Qwen2. 5-math technical report: Toward mathematical expert model via self-improvement. *arXiv preprint arXiv:2409.12122*.

Huaiyuan Ying, Shuo Zhang, Linyang Li, Zhejian Zhou, Yunfan Shao, Zhaoye Fei, Yichuan Ma, Jiawei Hong, Kuikun Liu, Ziyi Wang, et al. 2024. Internlm-math: Open math large language models toward verifiable reasoning. *arXiv preprint arXiv:2402.06332*.

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*.

Zheng Yuan, Hongyi Yuan, Chengpeng Li, Guanting Dong, Chuanqi Tan, and Chang Zhou. 2023. Scaling relationship on learning mathematical reasoning with large language models. *arXiv preprint arXiv:2308.01825*.

Hongyi Zhang, Moustapha Cissé, Yann N. Dauphin, and David Lopez-Paz. 2018. mixup: Beyond empirical risk minimization. In *ICLR (Poster)*. OpenReview.net.

Rongzhi Zhang, Yue Yu, and Chao Zhang. 2020. Seqmix: Augmenting active sequence labeling via sequence mixup. In *EMNLP (1)*, pages 8566–8579. Association for Computational Linguistics.

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.

Yaowei Zheng, Richong Zhang, Junhao Zhang, Yanhan Ye, Zheyan Luo, Zhangchi Feng, and Yongqiang Ma. 2024. Llamafactory: Unified efficient fine-tuning of 100+ language models. In *Proceedings of the 62nd Annual Meeting of the Association for Computational Linguistics (Volume 3: System Demonstrations)*, Bangkok, Thailand. Association for Computational Linguistics.

A Prompts

We show the prompts used for *Sequential Fusion* in Prompt 1, *Parallel Fusion* in Prompt 2, and *Conditional Fusion* in Prompt 3. We also provide the problem evaluation prompts in Prompt 4, which is partially derived from WizardMath (Luo et al., 2023). We use LangGPT (Wang et al., 2024b) to format prompts in Markdown and polish them.

B General Settings

B.1 Data Synthesis

We synthesize the augmented data, both the fusion process and the generation of the corresponding solutions, using GPT-4o-mini(*gpt-4o-mini-2024-07-18*) (OpenAI et al., 2023). We set the temperature to 0.7 and the maximum length of generation to 4096. The statistics of the generated data, as well as the base GSM8K (Cobbe et al., 2021) and MATH (Hendrycks et al., 2021) datasets, are shown in Table 4.

Dataset	GSM8K	MATH	Total
Standard	7.5K	7.5K	15K
MathFusionQA (*Sequential*)	15K	15K	30K
MathFusionQA (*Parallel*)	15K	15K	30K
MathFusionQA (*Conditional*)	15K	15K	30K
MathFusionQA	30K	30K	60K

Table 4: Statistics of the MathFusionQA dataset and the original datasets GSM8K and MATH.

B.2 Training

We use LLaMA-Factory (Zheng et al., 2024) to fine-tune the models. All models, including our own reproductions of baselines, are fine-tuned for 3 epochs with a batch size of 128 on 8xNVIDIA A100 GPU. The peak learning rate is 5e-6 with a linear warm-up for the first 3% of the training steps, followed by cosine decay. The maximum sequence length is set to 4096.

In Table 2, we reproduce the results of the baselines with 60K data. For MetaMath (Yu et al., 2024), MMIQC (Liu et al., 2024), and DART-Math (Tong et al., 2024), we directly downsample 60K data from the original datasets randomly. For RefAug (Zhang et al., 2024), the original training set only contains 30K data, with 15K from GSM8K and MATH, and 15K from the augmented reflection data. To upsample the RefAug dataset to 60K, we re-generate the reflection data two times using GPT-4o-mini with the original prompts (Zhang et al., 2024), thus obtaining an additional 30K data and forming the 60K dataset.

B.3 Evaluation

We compare MathFusion models with baselines on the following six benchmarks:

- **GSM8K** (Cobbe et al., 2021) dataset includes 8,792 high-quality grade school math word problems, with 7,473 for training and 1,319 for testing. Each problem in GSM8K requires between 2 and 8 steps to solve.

- **MATH** (Hendrycks et al., 2021) dataset is composed of 12,500 problems from high school math competitions, with 7,500 for training and 5,000 for testing. Problems in MATH are categorized into 7 types (Prealgebra, Intermediate Algebra, Algebra, Precalculus, Geometry, Counting & Probability, and Number Theory) and 5 difficulty levels.

- **CollegeMath** (Tang et al., 2024) test set contains 2,818 college-level problems, which are curated from 9 college-level mathematics textbooks, covering 7 key mathematical disciplines: Algebra, Precalculus, Calculus, VectorCalculus, Probability, LinearAlgebra, and Differential Equations.

- **DeepMind-Mathematics** (Saxton et al., 2019) test set consists of 1,000 problems covering a wide range of mathematical reasoning tasks spanning algebra, arithmetic, calculus, and probability designed to evaluate the mathematical reasoning abilities of models.

- **OlympiadBench-Math** (He et al., 2024) benchmark including 675 Olympiad-level mathematical problems, and we only use the text-only English subset of Olympiad-Bench.

- **TheoremQA** (Chen et al., 2023) is a novel theorem-driven question-answering benchmark containing 800 problems based on 350 theorems. It is designed to evaluate LLM's ability to apply domain-specific theorems across fields such as Mathematics, Physics, Electrical Engineering, Computer Science, and Finance.

B.4 Templates

For most of the results from our own runs, we use the template *"Question: {problem}\nAnswer:"*

for training, and *"Question: {problem}\nAnswer: Let's think step by step."* for evaluation. There are two exceptions: (1) For reproduced DART-Math (Tong et al., 2024), we use its default Alpaca template: *"Below is an instruction that describes a task. Write a response that appropriately completes the request.\n\n###Instruction:\n{problem}\n\n### Response:\n"*. (2) For evaluation on the Deep-Mind Mathematics benchmark for models fine-tuned from Llama3-8B, we find the Alpaca template yields consistently better performance than the template above. Therefore we use the Alpaca template for all the Llama3-8B evaluation on this dataset.

C Analysis of Fused Problems

The embedding search naturally ensures a high degree of contextual similarity. In the following sections, we analyze the fused problems in terms of problem types and errors.

C.1 Fused Probelm Types

Regarding problem types, in the GSM8K (Cobbe et al., 2021) dataset, all problems are simple algebra questions. For the MATH dataset, we find that 83% of the problem pairs belong to the same category, further validating the feasibility of the embedding search. We plot the distribution of combination types of problems in MATH in Figure 5.

C.2 Fused Error Analysis

In practice, we find that some fused problems are unreasonable or ambiguous, which are shown in Section G. The reason may be that some problems are not suitable for fusion or the limited capacity of the model for generating fused problems. To verify the correctness of the fused problems and their influence on the model's performance, we conduct an error analysis on the fused problems. Specifically, borrowing the idea from rejection sampling (Yuan et al., 2023), we use GPT-4o-mini to verify the correctness and completeness of the fused problems. The corresponding evaluation prompt is shown in Section A. For each identified unreasonable problem, we adjust the temperature to 1.0 to enhance the diversity of generation, and re-generate the problems five times using the corresponding fusion strategy. If none of the five generated problems is reasonable, we consider the fusion to be unreasonable and discard it. Finally, 5.6% of the fused problems are identified as unreasonable, and the remaining reasonable problems are added to the dataset. The average performance of Llama3-8B fine-tuned only on the filtered MathFusionQA is 39.1, which is similar to the performance of the model fine-tuned on the original MathFusionQA (39.0), indicating that the unreasonable problems have little impact on the model's performance. This result aligns with the findings in OpenMathInstruct-2 (Toshniwal et al.), indicating models exhibit some robustness to low-quality data in SFT.

D Effect of Teacher Model

In MathFusion, we use GPT-4o-mini (OpenAI et al., 2023) as the teacher model to generate the solutions for the fused problems. To validate the performance improvement of MathFusion is not merely due to the stronger teacher model, we conduct two ablation studies: (1) use GPT-4o-mini to rewrite the solutions from the original training set; and (2) follow DART-Math (Tong et al., 2024) to use DeepSeekMath-7B-RL (Shao et al., 2024) to generate solutions for the fused problems. The results are shown in Table 6. We can see that the performance of the model fine-tuned on the rewritten solutions is better than the Standard setting, especially on the MATH and GSM8K datasets. However, the average improvement is only 2.9 points. Meanwhile, each fusion strategy of Math-Fusion still outperforms the rewritten solution by a large margin. Additionally, though DeepSeekMath-7B-RL underperforms GPT-4o-mini in distillation quality (34.6 v.s. 36.1 on average), it still outperforms DART-Math (34.6 v.s. 32.9 on average). These results indicate that the performance improvement of MathFusion mainly comes from the new problem generated by three fusion strategies rather than the stronger teacher model.

E Additional Baseline

A most recent work, Mosaic-IT (Li et al., 2024c), shares similar idea with our MathFusion. Mosaic-IT is a model-free data augmentation technique that operates by concatenating existing instruction-following datasets and subsequently training LLMs using these augmented data instances along with meta-instructions. We conduct a comparison with the "Primary Strategy" proposed in Mosaic-IT, where the question pairs (same as MathFusion) and corresponding solutions from the original GSM8K and MATH datasets are concatenated into a single sample for SFT, resulting in 15K data. To miti-

Model	In-Domain		Out-of-Domain				AVG
	MATH	GSM8K	College	DM	Olympiad	Theorem	
Standard #1	17.4	63.1	12.1	23.1	3.7	9.6	21.5
Standard #2	17.6	63.7	12.6	20.6	4.3	8.9	21.3
Standard #3	17.5	65.4	12.9	21.6	4.7	10.9	22.2
Standard (Avg.)	$17.5_{\pm0.1}$	$64.1_{\pm1.2}$	$12.5_{\pm0.4}$	$21.8_{\pm1.3}$	$4.2_{\pm0.5}$	$9.8_{\pm1.0}$	$21.7_{\pm0.5}$
MathFusion #1	45.6	79.9	27.1	44.4	17.2	19.5	39.0
MathFusion #2	45.3	79.8	27.5	45.4	17.0	19.4	39.1
MathFusion #3	46.5	79.2	27.9	43.4	17.2	20.0	39.0
MathFusion (Avg.)	$45.8_{\pm0.6}$	$79.6_{\pm0.4}$	$27.5_{\pm0.4}$	$44.4_{\pm1.0}$	$17.1_{\pm0.1}$	$19.6_{\pm0.3}$	$39.0_{\pm0.1}$

Table 5: Performance comparison between the standard setting and MathFusion across six benchmarks with three random runs. The average performance is reported with the standard deviation.

Model	# Samples	In-Domain		Out-of-Domain				AVG
		MATH	GSM8K	College	DM	Olympiad	Theorem	
Standard	15K	12.4	60.3	8.4	17.0	2.2	7.6	18.0
GPT Rewritten	15K	20.1	70.3	9.1	13.9	2.8	8.9	20.9
Mosaic-IT	15K	11.7	40.9	7.4	9.2	2.7	9.9	13.6
Mosaic-IT + Original GSM8K and MATH	30K	11.0	54.7	6.9	9.8	1.9	9.5	15.6
DART-Math	60K	34.1	77.2	23.4	36.0	8.7	18.2	32.9
MathFusion (*Sequential*)	30K	32.7	73.9	18.9	29.3	9.3	15.5	29.9
MathFusion (*Parallel*)	30K	30.9	75.1	20.9	26.5	11.0	15.2	29.9
MathFusion (*Conditional*)	30K	26.3	73.0	15.6	21.4	7.3	12.8	26.1
MathFusion (DeepSeekMath-7B-RL)	60K	42.0	78.1	24.0	36.5	13.0	13.8	34.6
MathFusion	60K	41.6	79.8	24.3	39.2	13.6	18.1	36.1

Table 6: Additional results based on Mistral-7B.

gate overfitting to the pattern of answering multiple questions jointly, we also conduct an additional experiment which combine Mosaic-IT and the original GSM8K and MATH training sets during training, resulting in 30K data in total. The results are shown in Table 6. We observe that the Mosaic-IT leads to inferior performance, even worse than the Standard setting (i.e., using only the original GSM8K and MATH training data). We suspect this may be due to the lack of logical integration between problems when simply concatenated—unlike MathFusion, which explicitly introduces semantic or reasoning connections (e.g., sequential dependency or comparative logic) through its fusion strategies. This highlights the advantage of model-driven, structure-aware fusion over model-free concatenation.

F Significant Test

We conduct error analysis on MathFusion on Llama3-8B model to verify the consistent performance improvement of our MathFusionQA. Specifically, we fine-tune the Llama3-8B model on the original training sets (Standard setting), and the combined fusion strategies, respectively. The results are shown in Table 5. We can see that the MathFusion models consistently outperform the standard setting across all benchmarks. We also conduct statistical significance tests using the paired t-test, and results show that the performance improvement of MathFusion is statistically significant ($p < 0.05$) on all benchmarks.

G More Cases

More cases, including the original problems P_A and P_B, the fused problem P_F, are shown below. Specifically, we show three reasonable cases in Case G.1, Case G.2, and Case G.3, and three unreasonable cases in Case G.4, Case G.5, and Case G.6.
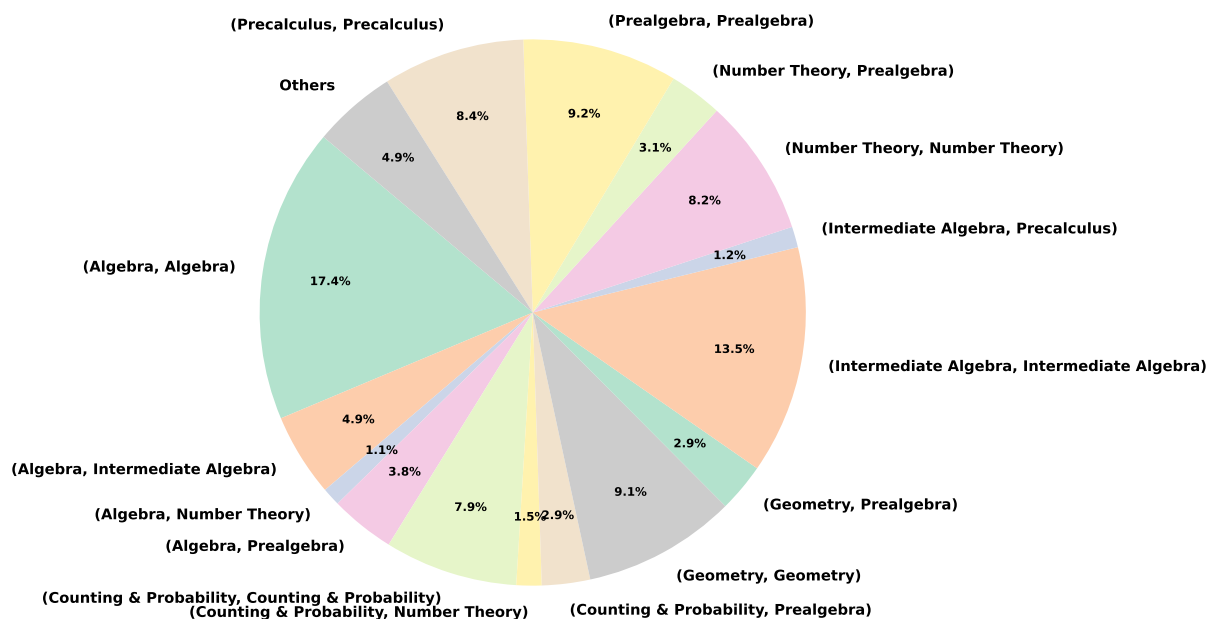
Figure 5: Distribution of combination types of problems in MATH dataset.

Prompt 1: *Sequential Fusion*

 # Role: Mathematical Problem Merger

Profile
Your role is to merge "#Problem 1#" and "#Problem 2#" into a combined problem.

Guidelines
Step 1: Identify input and output variables in both problems. Determine mathematical relationships and constraints in each problem. Locate variables between "#Problem 1#" and "#Problem 2#" that can form sequential dependencies.

Step 2: Formulate a comprehensive plan to merge the two problems by using "#Problem 1#"'s output variable to replace an input variable of "#Problem 2#"'s. Merge contextual elements by embedding both problems within a unified real-world scenario or extended narrative, aligning units and measurement systems.

Step 3: Create a single "#Combined Problem#" where solving "#Problem 1#" is a prerequisite for "#Problem 2#". Explicitly state variable dependencies and which variable is replaced. Adjust numerical ranges to maintain arithmetic consistency. The "#Combined Problem#" should contain no supplementary explanation or note.

Output Format
Please reply strictly in the following format:
#Elements Identified#:
#Plan#:
#Combined Problem#:

Input
#Problem 1#
{problem1}

#Problem 2#
{problem2}

Output

Prompt 2: *Parallel Fusion*

 # Role: Mathematical Problem Synthesizer

Profile Your role is to organically integrate "#Problem 1#" and "#Problem 2#" to create a novel problem that requires advanced synthesis of their mathematical essence.

Guidelines
Step 1: Conduct deep structural analysis of both problems by identifying their fundamental mathematical operations, contextual frameworks, and cognitive patterns. Extract the underlying logical architectures while preserving their distinctive solution pathways.

Step 2: Develop an innovative fusion mechanism by discovering non-obvious mathematical connections between the problems' core concepts. Construct a multidimensional scenario that naturally embeds both original contexts through temporal sequencing, spatial superposition, or conceptual analogy. Engineer hybrid parameters that inherit characteristics from both source problems while introducing emergent properties.

Step 3: Formulate the synthesized problem through strategic recombination of mathematical elements, ensuring the new problem requires concurrent application of both original solution strategies. Introduce controlled complexity through cross-domain constraints and self-verification mechanisms that establish mathematical consistency with both source problems' answers.

Output Format
Please reply strictly in the following format:
#Core Elements#:
#Synthesis Method#:
#New Problem#:

Input
#Problem 1#
{problem1}

#Problem 2#
{problem2}

Output

Prompt 3: *Conditional Fusion*

 # Role: Problem Integrator

Profile
Create a real-world problem where the solution requires solving both "#Problem 1#" and "#Problem 2#" independently.
***Ensure the the final answer is either from "#Problem 1#" or "#Problem 2#", depends on the "#New Question#"**.*

Guidelines
Step 1: Analyze "#Problem 1#" and "#Problem 2#" and make sure that the output variables they ask about are of the same type. If they are different (for example, one asks about time and the other asks about price), modify one of the problem so that it asks about the same variable as the other.
*Step 2: Design a unified problem scenario that combines "#Problem 1#" and "#Problem 2#". Introduce a "#New Question#", which must be related with both "#Problem 1#" and "#Problem 2#". Ensure that final answer of the "#New Question#" must either come from "#Problem 1#" or "#Problem 2#". This means that the "#New Question#" should be an **comparison** and **selection** of the previous answers, not their **combination**. There are some examples for the "#New Question#":*

  1. Who sells the most items?

  2. How much money does the top earner make?

  3. Which is the cheaper plan?

  4. Someone has 200 dollor, which item can he afford?

Step 3: Provide the "#New Problem#", which combine "#Problem 1#", "#Problem 2#", and "#New Question#" in a unified real-world scenario. Don't contain solution of "#Problem 1#" and "#Problem 2#" in "#New Problem#". Avoid using the phrases "#Problem 1#" and "#Problem 2#" in the generated "#New Problem#".

Output Format
Please reply strictly in the following format:
#Analysis#:
#New Question#:
#New Problem#:

Input
#Problem 1#
{problem1}

#Problem 2#
{problem2}

Output

Prompt 4: Problem Evaluation

 # Role: Mathematics Grading Teacher

Profile
You are a senior mathematics grading teacher in university, very skilled in high difficulty fields such as Intermediate Algebra, Precalculus, Prealgebra, Number Theory, Geometry, Counting & Probability, Algebra and so on.

Guidelines
Your task is to act as an impartial judge to evaluate the statement completeness and correctness of math problem according to the following rules:

1. *Assess the clarity and accuracy of the definition of each math problem. Ensure that the problem statement provides sufficient information, conditions, and constraints.*

2. *Consider whether the problem allows for multiple interpretations or if further clarification is needed.*

3. *Evaluate the clarity of mathematical notation and terminology used in the problem.*

4. *Evaluate whether the math problem is solvable. If the math problem meet the rules above, output "True" in "#Judgement#", else "False". You should also give your explanation in "#Explanation#".*

Output Format
Please reply strictly in the following format:
#Judgement#:
#Explanation#:

Input
{question}

Output

Example G.1: Case 1

P_A: Noah is a painter. He paints pictures and sells them at the park. He charges $60 for a large painting and $30 for a small painting. Last month he sold eight large paintings and four small paintings. If he sold twice as much this month, how much is his sales for this month?

P_B: Michael loves to paint and sells his creations. He charges $100 for a large painting and $80 for a small painting. At his last art show, he sold 5 large paintings and 8 small paintings. How much did he earn in all?

P_F^{seq}: Noah is a painter who charges $60 for a large painting and $30 for a small painting. Last month, he sold 8 large paintings and 4 small paintings. This month, he sold twice as many paintings as last month. Calculate Noah's total sales for this month. Michael, another painter, charges $100 for a large painting and $80 for a small painting. At his last art show, he sold 5 large paintings and the same amount in dollars as Noah's total sales from this month in small paintings. How much did Michael earn in all?

P_F^{para}: Noah and Michael are collaborating in an art exhibit. Noah charges $60 for a large painting and $30 for a small painting. Last month, he sold eight large paintings and four small paintings. This month, he plans to sell 50% more of each type. Michael charges $100 for a large painting and $80 for a small painting. At his last show, he sold 5 large paintings and 8 small paintings. For the exhibit, he aims to sell 40% more large paintings and 25% more small paintings than last time. How much total sales will they collectively earn from the exhibit?

P_F^{cond}: Noah is a painter who sells his artwork at the park. He charges $60 for a large painting and $30 for a small painting. Last month, he sold eight large paintings and four small paintings. This month, he has doubled his sales. Meanwhile, Michael, another painter, charges $100 for a large painting and $80 for a small painting. At his last art show, he sold 5 large paintings and 8 small paintings. Who earned more from their painting sales this month, Noah or Michael?

Example G.2: Case 2

P_A: There were 133 people at a camp. There were 33 more boys than girls. How many girls were at the camp?

P_B: At a party, there are 50 people. 30% of them are boys. How many girls are at this party?

- -

P_F^{seq}: At a camp, there were 133 people, with 33 more boys than girls. How many girls were at the camp? Subsequently, at a party where the total number of people is 50 plus the number of girls from the camp, 30% of them are boys. How many girls are at this party?

- -

P_F^{para}: At a camp, there are 133 people, and there are 33 more boys than girls. At a party, there are 50 people, and 30% of them are boys. What is the total number of girls at both the camp and the party?

- -

P_F^{cond}: At a summer camp, there were 133 people. There were 33 more boys than girls. At the same time, at a party, there are 50 people, and 30% of them are boys. How many girls are at each location, and which location has more girls?

Example G.3: Case 3

P_A: Add 53.463 to 12.9873 and round to the nearest thousandth.

P_B: Add 81.76 to 34.587 and round your answer to the nearest tenth.

- -

P_F^{seq}: First, add 53.463 to 12.9873 and round your answer to the nearest thousandth. Let this result be X. Then, add X to 81.76 and 34.587, and round your final answer to the nearest tenth.

- -

P_F^{para}: Calculate the sum of 53.463 and 81.76, then add this result to 34.587. Round the total to the nearest hundredth first, and then take that result and round it to the nearest whole number. What is the final answer?

- -

P_F^{cond}: A baker is preparing two different batches of cookies. For the first batch, she uses 53.463 ounces of flour and 12.9873 ounces of sugar. She wants to know the total weight of flour and sugar combined, rounded to the nearest thousandth. For the second batch, she uses $81.76 in ingredients and 34.587 in packaging. She needs to find the total cost of ingredients and packaging, rounded to the nearest tenth. After calculating both rounded totals, she wonders which amount is larger.

Example G.4: Error Case 1

P_A: What is the smallest positive integer with six positive odd integer divisors and twelve positive even integer divisors?

P_B: What is the smallest positive integer with exactly 10 positive integer divisors?

- -

P_F^{seq}: Find the smallest positive integer x that has six positive odd integer divisors and twelve positive even integer divisors. Once x is determined, calculate the smallest positive integer y that has exactly 10 positive integer divisors.

- -

Error: The output of P_A has no real relation to the input of P_B.

Example G.5: Error Case 2

P_A: In triangle ABC, $\cos(2A - B) + \sin(A + B) = 2$ and $AB = 4$. What is BC?

P_B: If triangle ABC has sides of length $AB = 6$, $AC = 5$, and $BC = 4$, then calculate

$$\frac{\cos \frac{A-B}{2}}{\sin \frac{C}{2}} - \frac{\sin \frac{A-B}{2}}{\cos \frac{C}{2}}.$$

P_F^{para}: In triangle ABC, the lengths of sides are $AB = 4$, $AC = 5$, and $BC = 6$. If it is given that $\cos(2A - B) + \sin(A + B) = 2$ holds true and the angles A, B, and C are interpreted through its trigonometric identities, compute the value of

$$\frac{\cos \frac{A-B}{2}}{\sin \frac{C}{2}} - \frac{\sin \frac{A-B}{2}}{\cos \frac{C}{2}}.$$

Further, verify your answer by ensuring that the relationships between side lengths corroborate the angle calculations used to derive it.

Error: The triangle can not satifsy both $AB = 6$, $AC = 5$, $BC = 4$, and $\cos(2A - B) + \sin(A + B) = 2$.

Example G.6: Error Case 3

P_A: Andrew and John are both Beatles fans. Their respective collections share nine of the same albums. Andrew has seventeen albums in his collection. Six albums are in John's collection, but not Andrew's. How many albums are in either Andrew's or John's collection, but not both?

P_B: Andy, Beth, Charlie, and Daniel take a test with thirty questions. Andy and Beth together get the same number of questions wrong as Charlie and Daniel together. Andy and Daniel together get four more questions wrong than Beth and Charlie do together. If Charlie gets five questions wrong, how many questions does Andy get wrong?

P_F^{cond}: Andrew and John are both Beatles fans. Their respective collections share nine of the same albums. Andrew has seventeen albums in his collection, while six albums are in John's collection, but not Andrew's. Meanwhile, Andy, Beth, Charlie, and Daniel take a test with thirty questions. Andy and Beth together get the same number of questions wrong as Charlie and Daniel together. Andy and Daniel together get four more questions wrong than Beth and Charlie do together. If Charlie gets five questions wrong, how many unique albums are in either Andrew's or John's collection, but not both, compared to how many questions Andy got wrong?

Error: There is no conditional relationship between the two problems.