FLAMES: Improving LLM Math Reasoning via a Fine-Grained Analysis of the Data Synthesis Pipeline

Parker Seegmiller[‡*], **Kartik Mehta**[†], **Soumya Saha**[†], **Chenyang Tao**[†], **Shereen Oraby**[†], **Arpit Gupta**[†], **Tagyoung Chung**[†], **Mohit Bansal**[†§], **Nanyun Peng**[†¶]
[†]Amazon AGI Foundations, [‡]Dartmouth College,
[§]UNC Chapel Hill, [¶]University of California, Los Angeles
pkseeg.gr@dartmouth.edu kartikmehta.iitd@gmail.com

Abstract

Recent works improving LLM math reasoning with synthetic data have used unique setups, making comparison of data synthesis strategies impractical. This leaves many unanswered questions about the roles of different factors in the synthetic data pipeline, such as the impact of filtering low-quality problems. To address this gap, we introduce FLAMES, a Framework for LLM Assessment of Math rEasoning Data Synthesis, and perform a systematic study of 10 existing data synthesis strategies and multiple other factors impacting the performance of synthetic math reasoning data. Our FLAMES experiments provide several valuable insights about the optimal balance of difficulty and diversity of synthetic data. First, data agents designed to increase problem complexity lead to best improvements on most math metrics. Second, with a fixed data generation budget, keeping higher problem coverage is more important than keeping only problems with reliable solutions. Third, GSM8K- and MATH-based synthetic data can lead to improvements on competition-level benchmarks, showcasing easy-to-hard generalization. Leveraging insights from our FLAMES experiments, we design two novel data synthesis strategies for improving out-of-domain generalization and robustness. Further, we develop the FLAMES dataset, an effective blend of our novel and existing data synthesis strategies, outperforming public datasets on OlympiadBench (+15.7), CollegeMath (+4.5), GSMPlus (+6.5), and MATH (+3.1). Fine-tuning Qwen2.5-Math-7B on the FLAMES dataset achieves 81.4% on MATH, surpassing larger Llama3 405B, GPT-4o and Claude 3.5 Sonnet.

1 Introduction

Solving math problems is a key measure of evaluating the reasoning ability of large language models (LLMs) (Wei et al., 2022; Cobbe et al., 2021; Hendrycks et al., 2021). Due to the challenges of creating a large-scale human-crafted dataset, LLM-based generation of synthetic data has been explored and proven effective in improving LLM math reasoning capabilities (Yu et al., 2023a; Huang et al., 2024a; Ding et al., 2024). While early works on math reasoning data synthesis focused on synthesis of solutions (reasoning traces) for existing problems, e.g. rejection sampling (Tong et al., 2024), recent works have shifted focus towards increasing problem difficulty and diversity by synthesizing new problems. Recent works such as OrcaMath (Mitra et al., 2024), MetaMath (Yu et al., 2023a), and OpenMath-Instruct-2 (Toshniwal et al., 2024a) have proposed one or more strategies[1] for generating new math problems. However, these works often use different setups (see Figure 2), making the comparison of data synthesis strategies across studies infeasible. This lack of standardization deprives researchers and practitioners of practical insights, leaving them uninformed about the real factors driving performance improvements

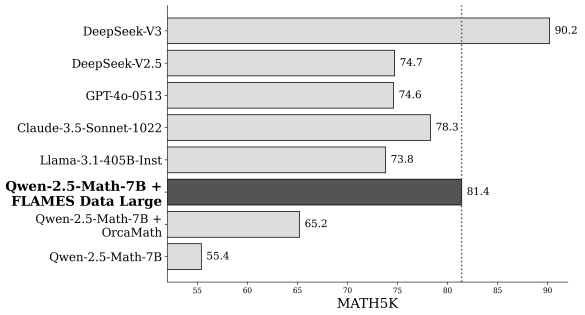

Figure 1: MATH benchmark scores for popular LLMs. Qwen2.5-Math-7B + X (FLAMES or OrcaMath) denotes results obtained by finetuning Qwen2.5-Math-7B model with X dataset. Comparison of FLAMES data with other public Math datasets is shown in Tables 3, 4.

[*]Work done while interning at Amazon AGI Foundations.

[1]Data synthesis strategies often use one or multiple LLM calls and are commonly referred to as data agents. We use the terms *strategies* and *agents* interchangeably in this paper.

Findings of the Association for Computational Linguistics: EMNLP 2025, pages 24746–24766
November 4-9, 2025

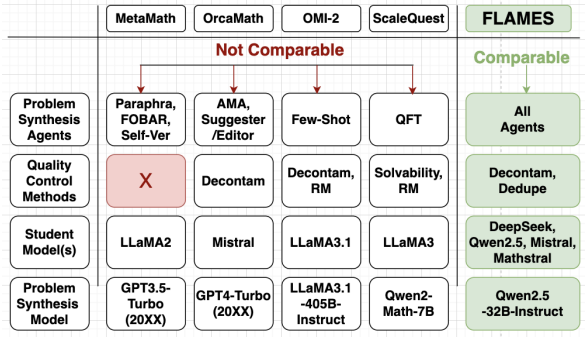

Figure 2: Landscape of recent math data synthesis works. Each work uses non standardized setups, such as different synthesis models, student models, and quality control, making comparison across works impractical. Detailed version is shown in Appendix Figure 5

via math reasoning data synthesis.

Figure 2 summarizes setups of four popular math reasoning data synthesis works in terms of four experimental factors, highlighting the lack of standardization. As different problem synthesis models and student models have been used in the OrcaMath and ScaleQuest papers, for example, it is infeasible to conclude whether Suggester-Editor (Mitra et al., 2024) is a better data synthesis strategy or question fine-tuning (QFT) (Ding et al., 2024). This lack of standardization poses several important research questions about math reasoning data synthesis, which are currently under-studied. **RQ1:** How do different strategies of data quality control affect the performance of synthetic data, and what strategy works the best? **RQ2:** Which data synthesis strategy performs optimally for improving student model math reasoning? **RQ3:** With a fixed compute budget for synthetic problem generation, should we mix data from multiple strategies or just scale the best strategy? **RQ4:** How does the choice of problem and solution generation models impact performance with math synthetic data?

To study these open research questions, we propose the Framework for LLM Assessment of Math rEasoning with Data Synthesis (FLAMES). Figure 3 shows a flow diagram of the FLAMES framework. In this framework, we create a comprehensive list of the factors that may impact performance within the synthetic data pipeline, and we choose fixed values for them based on existing literature and our experimental findings. The FLAMES framework enables controlled experiments where we vary only one factor, keeping other factors fixed. We perform a controlled study of ten existing data synthesis strategies, six data quality control strategies, two problem generation models, and two solution generation models. These experiments provide many valuable insights into data synthesis for LLM math reasoning. **First**, we find that data agents designed to increase problem complexity lead to best improvements on most math metrics. **Second**, with a fixed data generation budget, keeping higher coverage of synthetic problems (even with some inaccuracy) is more important than keeping only problems with reliable solutions. **Third**, synthetic data generated based on GSM8K and MATH can lead to improvements on competition-level benchmarks, showcasing easy-to-hard generalization capability. **Fourth**, choice of solution generation model impacts student model performance more than choice of problem generation model.

Leveraging the findings from our study of existing data synthesis strategies (Table 2), we identify out-of-domain (OOD) reasoning [2] and robustness to distracting information [3] as two weaknesses in existing data synthesis strategies. We design novel agents, **Taxonomy-Based Key Concepts** and **Distraction Insertion**, for addressing these weaknesses. We demonstrate with the FLAMES framework that the data generated using these two agents leads to superior performance gains on relevant OOD and distraction benchmarks.

Further, we study the impact of mixing data from different agents, and design an effective blend of agent data. We show empirically that this blend (FLAMES Small) leads to balanced performance across five evaluation datasets, with performance surpassing all ten existing agents on OOD and competition-level benchmarks. We scale this blend to construct the FLAMES Large (1M) and FLAMES XL (1.5M) datasets. Our experiments across five student models (Table 4), show that FLAMES Large leads to better performance than any existing public math dataset. With FLAMES XL, we observe drastic improvements over all public datasets across OlympiadBench (+12.8), GSM-plus (+4.9), CollegeMath (+5.8) and Math (+1.3). Fine-tuning Qwen2.5-Math-Base on the FLAMES Large dataset leads to 81.4% Math, surpassing larger Llama3 405B, GPT-4o and Claude 3.5 Sonnet (see Figure 1). These performance gains can be attributed to 1) findings from our FLAMES framework experiments on problem quality control, data agents, and teacher models, 2) our two novel data

[2] All ten agents lead to CollegeMath scores less than 41.0.

[3] At least 13 points difference between performance on GSM8K and GSMplus-Distraction for all ten agents.

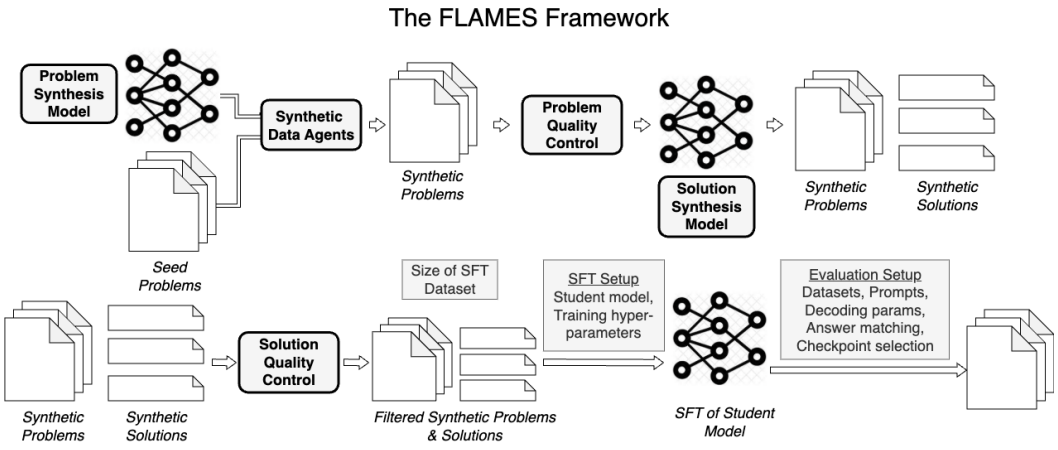

Figure 3: Overview of the FLAMES framework, showing fine-grained components of the Math data synthesis pipeline. Components for which we run **controlled experiments** are shown in bold font.

agents leading to improved performance on OOD and robustness metrics, and 3) our effective blend of data agents leading to balanced performance across all evaluation datasets.

In summary, our paper makes four contributions.
1. We propose the FLAMES framework for controlled study of multiple factors involved in the math synthetic data pipeline. This enables systematic comparison of existing and future math data agents.
2. We leverage the FLAMES framework to perform analysis of 12 data agents, 6 data quality control strategies, 2 problem generation and 2 solution generation models. Our experiments provide valuable insights for researchers working on improving LLM math reasoning with synthetic data.
3. We design 2 novel data synthesis agents, and empirically establish that they lead to enhanced robustness and out-of-domain improvements compared to existing data agents.
4. We develop the FLAMES dataset, consisting of an effective blend of our 2 novel agents and 2 strong existing data agents, showing drastic improvements over existing public math datasets.

2 The FLAMES Framework

Figure 3 shows a flow diagram of the FLAMES framework. There are multiple factors that play key roles in final student model performance after fine-tuning with synthetic data. As part of the FLAMES framework, we list these factors and provide their standard values based on either controlled experiments or existing literature. We highlight the factors for which we perform controlled experiments in Figure 3, and discuss those briefly in this section. We refer readers to Appendix A for a detailed discussion of these factors.

Problem Synthesis Agents: Math data synthesis agents interact with one or more seed problems and a problem generation model to synthesize novel math problems. These data agents play a crucial role in the synthetic math data pipeline, and are the focus of many recent works (Ding et al., 2024; Yu et al., 2023a; Mitra et al., 2024). In our work, we use the train splits from the popular GSM8K (Cobbe et al., 2021) and MATH (Hendrycks et al., 2021) data for seed problems. We discuss details of 10 existing and 2 novel data agents in Section 3.

Quality Control: All problems generated in the FLAMES framework are first deduplicated using exact match. We remove any synthetic problem if there is a high amount of token overlap between the synthetic problem and a test set problem. We study additional data quality control techniques in Section 4.

Student Model: Choice of student model plays a crucial role for studying LLM math reasoning improvements with synthetic data. DeepSeek-Math-7B (Shao et al., 2024) is a relatively-performant non-instruction-tuned base model, and is commonly used for measuring LLM math reasoning with synthetic data (Ding et al., 2024; Huang et al., 2024a). We use this model as the underlying student model for the FLAMES framework.

Problem Synthesis Model: To generate problems, data agents interact with Qwen2.5-32B-Instruct (Yang et al., 2024b), which is chosen based on a controlled experiment (see Section 5.5).

Solution Synthesis: An LLM acts as a solution generation teacher model providing solutions for

synthetic problems. Qwen2.5-Math-7B-Instruct (Yang et al., 2024b) has been shown to offer a good blend of performance and speed, scoring 95.2 and 83.6 on GSM8K and MATH, respectively. We use Qwen2.5-Math-7B-Instruct as the solution synthesis model in the FLAMES framework. We compare alternative models in Section 5.5.

Evaluation of Student Models: We evaluate fine-tuned student models on 5 datasets across 4 categories. For *in-domain evaluation*, we use the GSM8K (1,319 samples) (Cobbe et al., 2021) and the MATH5K test sets (5,000) (Hendrycks et al., 2021). These datasets are considered in-domain as their training datasets are used as seed problems for synthetic problem generation. To evaluate fine-tuned models on an *out-of-domain* set of problems which are similar in grade-level to the seed problems (MATH and GSM8K), we use the College-Math test set (2,818) [4], which contains exercises sourced from university-level textbooks across several math subjects. For *robustness* evaluation, we use GSMPlus dataset (Li et al., 2024), an adversarial grade school math dataset which includes GSM8K questions subject to numerical variation, arithmetic variation, and paraphrasing. We exclude the critical thinking portion for easy evaluation and separately report metrics for the distraction insertion subset. To evaluate *competition-level* reasoning, we use OlympiadBench [5] (675), which contains competition math problems covering difficult topics such as combinatorics and number theory.

Training Details: We do full-parameter fine-tuning on Amazon EC2 P4 instances[6] using Deepspeed Zero3 (Rajbhandari et al., 2020) for distributed training across 8 A100 GPUs. We train using a batch size of 4 for 5 epochs, saving 10 checkpoints. We implement SFT using the SWIFT library (Zhao et al., 2024), using default hyperparameters.

3 Data Synthesis Agents

We select 10 existing problem generation agents based on recent works in math data synthesis. Based on the primary objective of these agents, we categorize them as belonging to one of four categories: 1) In-domain practice, 2) In-domain complexity enhancing, 3) Robustness enhancing, and 4) Out-of-domain enhancing. Additionally, based on experimental findings detailed in Section 5.2, we propose two novel agents under the categories of robustness enhancing and out-of-domain enhancing. We briefly describe these 12 agents here, and refer readers to Appendix C for details.

3.1 Existing Agents

In-Domain Practice agents are designed to increase the quantity of in-domain data available for model training. We implement the **Few-Shot** agent (Toshniwal et al., 2024a) and the **Paraphrasing** agent (Yu et al., 2023a), along with the **Key Concepts** and **Seeded Key Concepts** agents (Huang et al., 2024a).

In-Domain Complexity Enhancing agents aim to increase complexity of existing math word problems (Mitra et al., 2024; Liu et al., 2024b; Luo et al., 2023). We implement the **Suggester-Editor** agent (Mitra et al., 2024) and the **Iterative Question Composing (IQC)** agent (Liu et al., 2024b).

Robustness Enhancing agents aim to alter existing problems to increase the robustness of the student model's reasoning process. These agents include the **Ask Me Anything** agent (Mitra et al., 2024), the **Self-Verification** agent (Yu et al., 2023a), and the **FOBAR** agent (Yu et al., 2023a).

Out-of-Domain Enhancing agents are not directly seeded with in-domain problems. We implement the **Question Fine-Tuning (QFT)** agent (Ding et al., 2024) for this category.

3.2 Novel Agents

We propose the novel **Taxonomy-Based Key Concepts** and **Distraction Insertion** agents, targeting out-of-domain generalization and robustness to distraction, respectively.

The **Taxonomy-Based Key Concepts** agent performs a two step generation of synthetic problems based on a novel curated taxonomy of math subjects. The first step generates a list of key concepts relevant to a given math subject, followed by the second step of generating novel problems based on each key concept. We curate our math taxonomy by combining taxonomies from a variety of sources (Huang et al., 2024a; Liu et al., 2024c; Huang et al., 2024b; Didolkar et al., 2024). This agent can be considered an out-of-domain enhancing agent, as it uses no seed problems, unlike other key concepts agents (Huang et al., 2024a).

[4] https://huggingface.co/datasets/qq8933/College_Math_Test

[5] https://huggingface.co/datasets/realtreetune/olympiadbench

[6] https://aws.amazon.com/ec2/instance-types/p4/

The **Distraction Insertion** agent inserts distracting information into an existing problem, without changing problem-relevant information. This agent is designed to improve reasoning performance in the presence of adversarial distracting information, which needs to be ignored for solving the problem.

See Figure 4 for examples and Appendix C.2 for the location of relevant prompts.

4 Data Quality Control

A challenge for LLM-based math problem synthesis is in controlling the quality of synthetic data, i.e. the removal of unsolvable or ill-formatted problems and incorrect solutions. Existing works for math data synthesis primarily rely on proprietary models like GPT-4 for problem and solution generation and assume that generated data is of good quality. Consequently, the impact of data quality control for math synthetic data has been an understudied problem. Our work is the first to study this factor in a principled way and provide insights on what works best for math synthetic data.

Recent work (Ding et al., 2024) has introduced the use of an LLM-based solvability filter to remove ill-formatted or unsolvable problems and a reward model (InternLM2-7B-Reward) to select the preferred solution. Self-consistency (Wang et al., 2023) is another viable alternative for ensuring problem and answer quality, where multiple (three in our case) solutions are generated for a problem with temperature sampling and only those problems are kept where at least a pre-determined number of the solutions lead to the same answer.

We use these to design six data filtering strategies for our experiments. In **Strict Self-Consistency**, only problems with a matching answer in all 3 solutions are kept and one solution is randomly selected. **Majority Self-Consistency** (or Majority) keeps problems where at least 2 solutions contain matching answers, and one of those solutions is randomly selected. **Solvability + RM** first filters problems which are deemed "unsolvable[7]" by the Qwen2.5-Math-7B-Instruct model, then selecting the solution which receives the highest reward according to the InternLM2-7b-Reward reward model (RM) (Cai et al., 2024). **Majority + First** is similar to Majority Self-Consistency, but uses the first solution in case no majority is found. This method doesn't remove any problems as part of the data quality control. **Solvability + First** filters unsolvable problems, then includes the first solution for each problem considered solvable. **First** keeps the first generated solution for each synthetic problem, keeping all the problems.

We note that these different techniques are designed to remove noisy synthetic data, but may also end up removing some legitimate problems and solutions, especially harder problems. We choose these six strategies to evaluate a diverse array of coverage, difficulty and accuracy levels in filtered synthetic datasets. We discuss findings of our systematic study of these strategies in Section 5.1.

5 Experimental Results

Here we explore quality control methods for synthetic math reasoning data (Section 5.1). We utilize the FLAMES framework to compare math reasoning data synthesis strategies (Section 5.2). We introduce and show the effectiveness of the FLAMES datasets compared with existing math datasets (Section 5.3 and Section 5.4). Last, we evaluate the relative impact of problem and solution generation teacher models (Section 5.5).

5.1 Quality Control of Synthetic Data

To study the impact of quality control strategies in the math synthetic data pipeline, we perform fine-tuning of the DeepSeek-Math-7B model using datasets with varying types of quality control applied. We start with 150k synthetic problems generated using the Suggester-Editor agent[8] with the Qwen2.5-32B-Instruct model, and synthesize three independent solutions to each problem using the Qwen2.5-Math-7B-Instruct model. After applying each quality control method from Section 4 for removing problems and selecting solutions, we use the resulting dataset for finetuning. Finally, we evaluate the fine-tuned model on in-domain test sets. We perform two experiments: **varying coverage**, where we use the resulting dataset after filtering, and **fixed coverage**, where we subsample each filtered dataset to have the same training size (45K)[9].

Table 1 shows results of this experiment. We observe four key findings. **First**, we find that solvability filtering (in its current form) is not effective

[7]The solvability filtering prompt can be found in Appendix F.

[8]We use Suggester-Editor data agent for this study, as we find this agent to be highly performant.

[9]45K matches the size of the smallest dataset (Strict Self-Consistency) after filtering.

			Varying Coverage				Fixed Coverage (45K)		
Method	**Row**	**Precision**	**Size**	**GSM8K**	**MATH**	**Avg**	**GSM8K**	**MATH**	**Avg**
Strict Self-Consistency	R1	Highest	45K (0.3)	82.9	56.4	69.7	82.9	56.4	69.7
Solvability + RM	R2	High	70K (0.47)	84.5	57.9	71.2	82.9	56.3	69.6
Solvability + First	R3	Low	70K (0.47)	84.1	58.2	71.2	83.2	55.6	69.4
Majority Self-Consistency	R4	High	90K (0.6)	84.5	59.4	72.0	82.0	56.1	69.1
First	R5	Low	**150K (1.0)**	84.8	**61.1**	73.0	**83.4**	55.8	69.6
Majority + First	R6	Medium	**150K (1.0)**	**86.4**	60.9	**73.7**	83.0	**56.6**	**69.8**

Table 1: Results of synthetic data quality control strategies for two settings of *Varying Coverage* and *Fixed Coverage*. Size column refers to number of problems remaining after filtering followed by fraction of actual in parenthesis. Each result is using DeepSeek-Math-7B fine-tuned on problems synthesized using Suggester-Editor agent.

	In-Domain			OOD	Robustness		Competition	
Model	**GSM8K**	**MATH5K**	**Average**	**College Math**	**Distraction**	**GSMPlus**	**Olympiad Bench**	**Average**
DeepSeek-Math-7B (Base)	64.2	36.2	50.2	15.9	17.6	22.6	5.3	28.8
DeepSeek-Math-7B-Instruct	82.9	51.7	67.3	33.9	60.0	70.1	13.8	50.5
Agents	**In-Domain Practice**							
Seeded Key Concepts	81.7	58.7	70.2	38.6	67.8	70.9	22.4	54.5
Key Concepts	79.2	56.2	67.7	38.6	66.1	69.4	22.7	53.2
Paraphrase	81.2	57.5	69.3	38.0	62.6	69.7	22.5	53.8
Few-Shot	81.9	57.2	69.6	37.2	66.2	70.1	23.1	53.9
Agents	**In-Domain Complexity Enhancing**							
IQC	86.0	59.9	73.0	38.7	72.1	75.7	24.4	56.9
Suggester-Editor	85.3	**61.0**	**73.2**	39.4	72.1	**75.9**	25.2	57.4
Agents	**Out-of-Domain Enhancing**							
QFT	79.0	57.5	68.3	40.5	67.0	68.6	23.6	53.8
Agents	**Robustness Enhancing**							
Ask Me Anything	83.2	56.4	69.8	40.0	65.1	71.4	23.7	54.9
FOBAR	80.5	55.8	68.2	37.9	61.2	69.6	19.9	52.7
Self-Verification	82.7	57.9	70.3	38.1	60.9	71.7	23.4	54.8
Agents	**Novel Agents**							
Distraction Insertion (Ours)	83.3	59.4	71.3	39.6	**72.4**	73.5	24.7	56.1
Taxonomy Key Concepts (Ours)	77.1	56.1	66.6	40.9	64.7	67.3	21.8	52.6
Dataset	**FLAMES Data Mixture**							
FLAMES Small	85.2	60.0	72.6	**41.4**	72.2	74.7	**26.1**	**57.5**

Table 2: Results comparing 10 existing and 2 novel data agents. We also include results for FLAMES_small (150K) for easy comparison with all agents. DeepSeek-Math-7B is fine-tuned using 150K problems for each setting.

as *Solvability* + *First* (R3) shows inferior performance to *First* (R5). This is likely due to filtering out genuine problems with the solvability filter. Using the solvability filter on the human-crafted MATH test set, we observe that it removes 30% of the real problems (30% and 50% from difficulty levels 4 and 5, respectively), indicating that this filtering is unreliable (see Appendix D.3 for details). **Second**, coverage of problems matters more than the accuracy of the solutions. We observe superior performance with *Majority* + *First* (R6) than *Majority* (R4), where the only difference (between R6 and R4) is inclusion of additional 60K problems in R6 where first solution is used. This shows that including more problems, even with potentially incorrect solutions, leads to better performance. **Third**, with the same scale and problem complexity, precision of the solution matters, as *Majority* + *First* (R6) shows better performance than *First* (R5) strategy. This is also evident from *fixed coverage* analysis, where we observe higher Math scores for strategies designed for higher precision[10]. **Fourth**, *Majority* + *First* (R6) provides a good blend of coverage of problems and accuracy of the solutions, and superior performance than other strategies. With comparable performance, the *First* (R5) strategy is computationally better than the *Majority* + *First* (R6). Hence, we use *First* strategy for all our further studies.

5.2 Comparison of Data Generation Agents

To assess which synthetic data agents generate the most performant data, we perform SFT of the DeepSeek-Math-7B base model with datasets gen-

[10]However, we observe less variation and a different trend on GSM8K problems. This is likely due to smaller difference between solutions with First and Majority strategy on relatively easier GSM8K level problems.

erated using each agent from Section 3. We generate 150K unique problems with each agent, using the *First* strategy for data quality control. We compare data agents by evaluating performance of their respective fine-tuned DeepSeek-Math-7B models.

Table 2 shows the results of each fine-tuned model across the 5 benchmark datasets. We observe four major findings from this study. **First**, we observe that complexity-enhancing agents (IQC and Suggester-Editor) lead to best improvement on in-domain metrics. Surprisingly, they also lead to best competition-level and robustness improvements. **Second**, the out-of-domain enhancing agents (particularly, our novel Taxonomy Key Concepts agent) lead to best improvement on out-of-domain reasoning, surpassing complexity-enhancing agents. **Third**, we find that our novel Distraction Insertion agent leads to best performance on the distraction insertion benchmark within the GSMPlus dataset, showcasing that the FLAMES framework enables design of agents to solve specific reasoning tasks. **Fourth**, we observe that data agents, while using only GSM8K and MATH as seed sets, lead to improvement over competition level benchmarks (as evident from OlympiadBench metrics). This is particularly interesting as it demonstrates "easy-to-hard generalization", and provides evidence that math data synthesis may be used to improve performance on more difficult math reasoning tasks.

5.3 Comparison of FLAMES Datasets

We study the impact of mixing data from different agents (refer to experiments in Appendix D.2) and observe best results by mixing 50% Suggester-Editor, 20% IQC, 20% Taxonomy Key Concept (our proposed) and 10% Distraction Insertion (our proposed) agents. Based on this blend, we develop 3 versions of FLAMES datasets — FLAMES Small (150K), FLAMES Large (1M), and FLAMES XL (1.5M). We compare FLAMES Small with individual agents in Table 2, and observed balanced performance across the five evaluation datasets, with performance surpassing all 12 agents on CollegeMath and Olympiad-Bench.

For comparing the performance of FLAMES Large and FLAMES XL datasets, we conduct finetuning of the DeepSeek-Math-7B model on a diverse set of existing math datasets, including GSM8K (Cobbe et al., 2021), MATH (Hendrycks et al.), NuminaMath (LI et al., 2024), MetaMathQA (Yu et al., 2023a), OrcaMath (Mitra et al., 2024), OpenMathInstruct2[11] (Toshniwal et al., 2024a), MMIQC (Liu et al., 2024b), and ScaleQuest (Ding et al., 2024). Since different synthetic datasets were synthesized with different solution generation models, we compare results with "refreshed" versions of each synthetic dataset, where one solution for each unique problem is generated using Qwen2.5-Math-7B-Instruct (same as used for our FLAMES dataset, for fair comparison). Results with original datasets, which performed worse in each case, are given in Appendix D.1.

Table 3 shows results for this study. We observe that ScaleQuest (refreshed with Qwen2.5-Math-7B-Instruct solutions) leads to best average performance (60.0) among the public datasets. At the same scale (1M), our FLAMES Large dataset achieves better MATH5K (68.3), CollegeMath (41.9), GSMPlus (79.4) and average score (61.7) than existing public datasets. On scaling our dataset to 1.5M size (FLAMES XL), we observe drastic improvements as compared to the public datasets across OlympiadBench (+12.8), College Math (+5.8), GSMPlus (+4.9) and Math (+1.3). It is important to note that unlike other synthetic datasets, our FLAMES datasets do not use any proprietary model and rely only on open source models. These findings validate the effectiveness of the FLAMES datasets, showcasing the impact of an optimized mixture of data agents and targeted augmentation in synthetic data generation.

5.4 Comparison with multiple Student Models

In this experiment, we compare our FLAMES Large dataset across several underlying student models. To ensure robust findings, we use four diverse student models — Qwen2.5-Math-7B and Qwen2.5-14B (Yang et al., 2024b), Mathstral-7B [12], and Mistral-7B-v0.3 [13]. We compare results with ScaleQuest refreshed as it has best results for public models in Table 3. Table 4 shows results for this experiment[14]. We observe that FLAMES Large outperforms refreshed ScaleQuest, achieving drastic gains on competition level benchmarks (+4.8 for DeepSeek-7B, +4.6 for Mathstral-7B, +4.4 for Mistral-7B-v0.3 and +3.7 for Qwen2.5-

[11] We include only unique problems from OpenMathInstruct2 for fair comparison, since ScaleQuest and FLAMES datasets contain unique problems.

[12] https://huggingface.co/mistralai/Mathstral-7B-v0.1

[13] https://huggingface.co/mistralai/Mistral-7B-v0.3

[14] See Table 10 in Appendix D.1 for detailed results.

		In-Domain		OOD	Robustness		Competition	
Model		**GSM8K**	**MATH5K**	**College Math**	**Distraction**	**GSMPlus**	**Olympiad Bench**	**Average**
DeepSeek-Math-7B (Base)		64.2	36.2	15.9	17.6	22.6	5.3	28.8
DeepSeek-Math-7B-Instruct		82.9	51.7	33.9	60	70.1	13.8	50.5
Dataset	**Dataset Size**	**Existing Math Datasets**						
GSM8K/MATH	15K	70.1	34.9	33.9	60	70.1	13.8	44.6
NuminaMath	860K	81.4	53.1	36.6	61.9	70.2	20.4	52.3
OpenMathInstruct2 (R)	600K	84.5	66.4	40.7	64.7	72.9	30.5	59.0
MetaMathQA (R)	150K	85.0	55.8	36.7	64.1	72.9	21.6	54.4
OrcaMath (R)	200K	84.5	52.8	36.9	68.1	74.9	19.7	53.8
MMIQC (R)	220K	84.8	61.5	38.4	63.7	73.6	24.4	56.5
ScaleQuest (R)	1M	**89.4**	66.0	41.4	71	78.3	25	60.0
Dataset	**Dataset Size**	**FLAMES Data**						
FLAMES Large	1M	89.2	**68.3**	41.9	76.1	79.4	29.8	61.7
FLAMES XL	1.5M	87.7	67.7	**47.2**	**80.3**	**83.2**	**43.3**	**65.8**

Table 3: Comparison of FLAMES datasets with open-source math reasoning datasets. DeepSeek-Math-7B is fine-tuned using unique problems for each dataset, alongside refreshed (R) solutions using Qwen2.5-Math-7B-Instruct for fair comparison.

		In-Domain		Competition	Avg
Student Model	**Dataset**	**GSM8K**	**MATH**	**Olympiad Bench**	
DeepSeek-7B	ScaleQuest (R)	**89.4**	66.0	25.0	60.0
DeepSeek-7B	FLAMES (L)	89.2	**68.3**	**29.8**	**61.7**
Qwen2.5M-7B	ScaleQuest (R)	**93.9**	80.7	**41.3**	69.2
Qwen2.5M-7B	FLAMES (L)	93.5	**81.4**	40.9	**69.5**
Mathstral-7B	ScaleQuest (R)	88.4	65.7	24.6	59.0
Mathstral-7B	FLAMES (L)	**89.2**	**68.1**	**29.2**	**61.1**
Mistral-7B	ScaleQuest (R)	84.7	59.3	19.7	54.8
Mistral-7B	FLAMES (L)	**86.4**	**63.6**	**24.1**	**57.5**
Qwen2.5-14B	ScaleQuest (R)	93.1	75.6	34.2	66.2
Qwen2.5-14B	FLAMES (L)	**93.3**	**76.7**	**37.9**	**67.4**

Table 4: Comparison of FLAMES_large and ScaleQuest across diverse student models. DeepSeek-7B refers to DeepSeek-Math-7B, Mistral-7B refers to Mistral-7B-v0.3, Qwen2.5M-7B refers to Qwen2.5-Math-7B, FLAMES (L) refers to FLAMES_large, ScaleQuest (R) refers to ScaleQuest problems with Qwen2.5-Math-7B-Instruct solutions. Both datasets contain 1M problems.

14B). These results showcase the efficacy of our FLAMES datasets across diverse student models.

5.5 Comparison of Problem and Solution Generation Models

For the FLAMES framework we use the Qwen2.5-32B-Instruct and Qwen2.5-Math-7B-Instruct (baseline setting) as the problem generation and solution generation model (Yang et al., 2024b), respectively. We now study the impact of updating the problem and solution generation models with weaker DeepSeek models. We generate 50k problems in each setting, using the performant Suggester-Editor agent, and fine-tune the DeepSeek-Math-7B as the student model. Table 5 shows results for this study. We observe a big drop (-7.1% points) in MATH5K numbers on replacing the answer generation model with DeepSeek-Math-7B-RL, but only a small drop (-1.5% points) on replacing the problem generation model with DeepSeek-v2.5. These results suggest that quality of the solution generation model plays a relatively bigger role than the quality of the problem generation model for math synthetic data pipeline. On the contrary, we observe small improvements (+1.0% and +2.6%) in GSM8K numbers. This is likely due to smaller difference in GSM8K numbers for Qwen and DeepSeek, and better alignment of DeepSeek based synthetic data for the DeepSeek-Math-7B student model.

Problem Generation	**Solution Generation**	**GSM8K**	**MATH**	**Average**
Qwen2.5-32B	Qwen2.5-7B	82.5	**56.4**	**69.5**
DeepSeek-v2.5	Qwen2.5-7B	83.5	54.9	69.2
Qwen2.5-32B	DeepSeek-7B	**85.1**	49.3	67.2

Table 5: Results of varying problem and solution generation models. Qwen2.5-32B refers to Qwen2.5-32B-Instruct model, DeepSeek-7B refers to DeepSeek-Math-7B-RL. Finetuning is done with 50K Suggester-Editor problems for DeepSeek-Math-7B as student model.

6 Conclusion

We introduced the FLAMES framework, which enables fine-grained analysis of the math data synthesis pipeline. We performed a controlled study of 12 data synthesis strategies (including 2 novel strategies), 6 data quality control strategies, 2 problem generation models, and 2 solution generation models, providing valuable insights for improving LLM math reasoning with synthetic data. We studied the impact of mixing data from different agents and designed an effective blend (the FLAMES datasets) without use of any proprietary models. Our rigorous evaluations establish drastic improvements

with our proposed FLAMES datasets over public math datasets.

7 Limitations and Potential Risks

One limitation of our work is reliance on a strong teacher model for solution generation. This assumption may be violated when working with less common languages which lack sufficient math data to begin with. While this may be mitigated by translating our English math data to those languages, we leave study of other languages for future work.

We acknowledge that there is a potential risk of inherent bias in our generated data due to bias in the problem and solution generating models or the original seed data (LeBrun et al., 2022; Yu et al., 2023b). This may be alleviated by introducing an additional step in the FLAMES framework for detecting and filtering biased samples, however we leave this for future work.

References

Bo Adler, Niket Agarwal, Ashwath Aithal, Dong H Anh, Pallab Bhattacharya, Annika Brundyn, Jared Casper, Bryan Catanzaro, Sharon Clay, Jonathan Cohen, et al. 2024. Nemotron-4 340b technical report. *arXiv preprint arXiv:2406.11704*.

Zheng Cai, Maosong Cao, Haojiong Chen, Kai Chen, Keyu Chen, Xin Chen, Xun Chen, Zehui Chen, Zhi Chen, Pei Chu, et al. 2024. Internlm2 technical report. *arXiv preprint arXiv:2403.17297*.

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

Yew Ken Chia, Guizhen Chen, Luu Anh Tuan, Soujanya Poria, and Lidong Bing. 2023. Contrastive chain-of-thought prompting. *Preprint*, arXiv:2311.09277.

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*.

DeepSeek-AI. 2024. Deepseek-v2: A strong, economical, and efficient mixture-of-experts language model. *Preprint*, arXiv:2405.04434.

Aniket Didolkar, Anirudh Goyal, Nan Rosemary Ke, Siyuan Guo, Michal Valko, Timothy Lillicrap, Danilo Rezende, Yoshua Bengio, Michael Mozer, and Sanjeev Arora. 2024. Metacognitive capabilities of llms: An exploration in mathematical problem solving. *Preprint*, arXiv:2405.12205.

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*.

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

Peter Hase, Mohit Bansal, Peter Clark, and Sarah Wiegreffe. 2024. The unreasonable effectiveness of easy training data for hard tasks. *Preprint*, arXiv:2401.06751.

Xuan He, Da Yin, and Nanyun Peng. 2024. Guiding through complexity: What makes good supervision for hard reasoning tasks? *Preprint*, arXiv:2410.20533.

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

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

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

Yinya Huang, Xiaohan Lin, Zhengying Liu, Qingxing Cao, Huajian Xin, Haiming Wang, Zhenguo Li, Linqi Song, and Xiaodan Liang. 2024b. Mustard: Mastering uniform synthesis of theorem and proof data. *Preprint*, arXiv:2402.08957.

Seungone Kim, Juyoung Suk, Xiang Yue, Vijay Viswanathan, Seongyun Lee, Yizhong Wang, Kiril Gashteovski, Carolin Lawrence, Sean Welleck, and Graham Neubig. 2024. Evaluating language models as synthetic data generators. *Preprint*, arXiv:2412.03679.

Benjamin LeBrun, Alessandro Sordoni, and Timothy J O'Donnell. 2022. Evaluating distributional distortion in neural language modeling. *arXiv preprint arXiv:2203.12788*.

Chengpeng Li, Zheng Yuan, Hongyi Yuan, Guanting Dong, Keming Lu, Jiancan Wu, Chuanqi Tan, Xiang Wang, and Chang Zhou. 2024. Mugglemath: Assessing the impact of query and response augmentation on math reasoning. In *Proceedings of the 62nd Annual Meeting of the Association for Computational Linguistics (Volume 1: Long Papers)*, pages 10230–10258.

Jia LI, Edward Beeching, Lewis Tunstall, Ben Lipkin, Roman Soletskyi, Shengyi Costa Huang, Kashif Rasul, Longhui Yu, Albert Jiang, Ziju Shen, Zihan Qin, Bin Dong, Li Zhou, Yann Fleureau, Guillaume Lample, and Stanislas Polu. 2024. Numinamath. [https://huggingface.co/AI-MO/NuminaMath-CoT](https://github.com/project-numina/aimo-progress-prize/blob/main/report/numina_dataset.pdf).

Qintong Li, Leyang Cui, Xueliang Zhao, Lingpeng Kong, and Wei Bi. 2024. Gsm-plus: A comprehensive benchmark for evaluating the robustness of llms as mathematical problem solvers. *arXiv preprint arXiv:2402.19255*.

Aixin Liu, Bei Feng, Bin Wang, Bingxuan Wang, Bo Liu, Chenggang Zhao, Chengqi Dengr, Chong Ruan, Damai Dai, Daya Guo, et al. 2024a. Deepseek-v2: A strong, economical, and efficient mixture-of-experts language model. *arXiv preprint arXiv:2405.04434*.

Haoxiong Liu, Yifan Zhang, Yifan Luo, and Andrew C Yao. 2024b. Augmenting math word problems via iterative question composing. In *ICLR 2024 Workshop on Navigating and Addressing Data Problems for Foundation Models*.

Hongwei Liu, Zilong Zheng, Yuxuan Qiao, Haodong Duan, Zhiwei Fei, Fengzhe Zhou, Wenwei Zhang, Songyang Zhang, Dahua Lin, and Kai Chen. 2024c. Mathbench: Evaluating the theory and application proficiency of llms with a hierarchical mathematics benchmark. *Preprint*, arXiv:2405.12209.

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. In *Proceedings of the 62nd Annual Meeting of the Association for Computational Linguistics (Volume 1: Long Papers)*, pages 2732–2747.

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*.

Mistral AI. 2025. Mathstral: Advancing Mathematical Reasoning with Language Models. Accessed: 2025-02-12.

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*.

Samyam Rajbhandari, Jeff Rasley, Olatunji Ruwase, and Yuxiong He. 2020. Zero: Memory optimizations toward training trillion parameter models. *Preprint*, arXiv:1910.02054.

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*.

Gemma Team, Morgane Riviere, Shreya Pathak, Pier Giuseppe Sessa, Cassidy Hardin, Surya Bhupatiraju, Léonard Hussenot, Thomas Mesnard, Bobak Shahriari, Alexandre Ramé, et al. 2024. Gemma 2: Improving open language models at a practical size. *arXiv preprint arXiv:2408.00118*.

Yuxuan Tong, Xiwen Zhang, Rui Wang, Rui Min Wu, and Junxian He. 2024. Dart-math: Difficulty-aware rejection tuning for mathematical problem-solving. *ArXiv*, abs/2407.13690.

Shubham Toshniwal, Wei Du, Ivan Moshkov, Branislav Kisacanin, Alexan Ayrapetyan, and Igor Gitman. 2024a. Openmathinstruct-2: Accelerating ai for math with massive open-source instruction data. In *The 4th Workshop on Mathematical Reasoning and AI at NeurIPS'24*.

Shubham Toshniwal, Ivan Moshkov, Sean Narenthiran, Daria Gitman, Fei Jia, and Igor Gitman. 2024b. Openmathinstruct-1: A 1.8 million math instruction tuning dataset. *arXiv preprint arXiv:2402.10176*.

Xuezhi Wang, Jason Wei, Dale Schuurmans, and Quoc V Le. 2023. H. chi, sharan narang, aakanksha chowdhery, and denny zhou. 2023. self-consistency improves chain of thought reasoning in language models. In *The Eleventh International Conference on Learning Representations*.

Jason Wei, Xuezhi Wang, Dale Schuurmans, Maarten Bosma, Fei Xia, Ed Chi, Quoc V Le, Denny Zhou, et al. 2022. Chain-of-thought prompting elicits reasoning in large language models. *Advances in neural information processing systems*, 35:24824–24837.

An Yang, Baosong Yang, Binyuan Hui, Bo Zheng, Bowen Yu, Chang Zhou, Chengpeng Li, Chengyuan Li, Dayiheng Liu, Fei Huang, et al. 2024a. Qwen2 technical report. *arXiv preprint arXiv:2407.10671*.

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

Longhui Yu, Weisen Jiang, Han Shi, YU Jincheng, Zhengying Liu, Yu Zhang, James Kwok, Zhenguo Li, Adrian Weller, and Weiyang Liu. 2023a. Metamath: Bootstrap your own mathematical questions for large language models. In *The Twelfth International Conference on Learning Representations*.

Yue Yu, Yuchen Zhuang, Jieyu Zhang, Yu Meng, Alexander J Ratner, Ranjay Krishna, Jiaming Shen, and Chao Zhang. 2023b. Large language model as attributed training data generator: A tale of diversity and bias. *Advances in Neural Information Processing Systems*, 36:55734–55784.

Yuze Zhao, Jintao Huang, Jinghan Hu, Xingjun Wang, Yunlin Mao, Daoze Zhang, Zeyinzi Jiang, Zhikai Wu, Baole Ai, Ang Wang, Wenmeng Zhou, and Yingda Chen. 2024. Swift:a scalable lightweight infrastructure for fine-tuning. *Preprint*, arXiv:2408.05517.

Contents of Appendix

A FLAMES Framework Details

Figure 3 shows a flow diagram of the FLAMES framework. Broadly, the synthetic data pipeline can be considered as consisting of the following steps. 1) Generating synthetic problems with a problem synthesis LLM and some seed problems (or taxonomy) according to some data agent, 2) generating solutions for synthetic problems with a solution synthesis LLM, 3) filtering the problems and solutions for quality control, 4) training a student model with this synthetic dataset, and 5) evaluating the student model on multiple evaluation benchmarks. There are multiple factors (or design decisions) that play key role in final student model performance with synthetic data. As part of the FLAMES framework, we list these factors. We include a table of these factors, their values in the FLAMES framework, and whether we choose values based on a controlled experiment or prior work in Table 6.

Problem Generation: Seeded problem generation agents in the FLAMES framework utilize the train splits from the popular GSM8K (Cobbe et al., 2021) and MATH (Hendrycks et al., 2021) training sets. These agents interact with the Qwen2.5-32B-Instruct model (Yang et al., 2024b) to generate new problems, an open-source instruction-tuned

Factor	Value	Controlled/Prior	Location
Problem Generation			
Data Synthesis Agent	10 Existing, 2 Novel	Controlled	Section 5.2
Problem Generation Model	Qwen2.5-32B-Instruct	Controlled	Section 5.5
Problem Generation Decoding Parameters	See Appendix A	Prior Work	Appendix A
Deduplication	Exact Match	Prior Work	Section 2
Benchmark Decontamination	N-Gram Overlap	Prior Work	Section 2
Solution Generation			
Solution Generation Model	Qwen2.5-Math-7B-Instruct	Controlled	Section 5.5
Solution Generation Decoding Parameters	See Appendix A	Prior Work	Appendix A
Solution Sampling Strategy	First Solution	Controlled	Section 5.1
Solution Verification Strategy	None	Controlled	Section 5.1
Supervised Fine-Tuning			
Student Model	DeepSeek-Math-7B, Qwen2.5-Math-7B, Mathstral-7B, Mistral-7B-v0.3, Qwen2.5-14B	Controlled	Section 5.4
Training Hyperparameters	See Section 2	Prior Work	Section 2
Training Setup	See Section 2	Prior Work	Section 2
Evaluation			
Checkpoint Selection	See Appendix A	Prior Work	Appendix A
Evaluation Prompts	See Appendix A	Prior Work	Appendix A
Evaluation Decoding Parameters	See Appendix A	Prior Work	Appendix A
Answer Extraction & Matching	See Appendix A	Prior Work	Appendix A

Table 6: Values of all factors fixed in the FLAMES framework. Each value is chosen based on either controlled experiments of prior work. Locations of framework factor details, and their related controlled experiments, are also given.

model which achieves 95.9 on GSM8K and 83.1 on MATH5K. We show in Section 5.5 that this model leads to better performance than the larger open-source DeepSeek-v2.5 model (DeepSeek-AI, 2024), while being more computationally efficient. Problems are generated using a temperature of 0.7 with sampling hyperparameters of $top\_p = 0.9$, $top\_k = 50$, and a repetition penalty of 1 for a maximum of 2,048 new tokens.

Solution Generation: Since solution generation can be less efficient due to higher inference time compute, we choose to use the smaller Qwen2.5-Math-7B-Instruct model (Yang et al., 2024b) for solution synthesis in the FLAMES framework. This follows (Ding et al., 2024), who use the earlier Qwen2-Math-7B-Instruct (Yang et al., 2024a) for both problem and solution synthesis. Solution generation uses the same generation parameters as in problem synthesis. As we show in Section 5.1, taking the first solution generated by Qwen2-Math-7B-Instruct for each synthetic problem leads to good performance while significantly reducing required generation compute.

Quality Control: All problems generated in the FLAMES framework are first deduplicated using exact match. While we do not use test splits of GSM8K or MATH when synthesizing problems, we exercise caution and follow (Mitra et al., 2024) in decontaminating synthetic problems against the GSM8K and MATH test sets. We remove any synthetic problem if there is a high amount of token overlap between the synthetic problem and a test set problem. In other words, if 95% of the 8-grams in a test set problem are generated in a synthetic problem, we remove that synthetic problem.

Dataset Size: Using these problem and solution generation processes, we generate 150k problems *after problem deduplication and decontamination* for each agent in the FLAMES framework. In other words, we fix the size of the SFT dataset for each data agent at 150k. Of these 150k problems, 75k are generated using GSM8K seed problems and 75k are generated using MATH seed problems[15]. Due to their reliance on seed problem permutations and the limited size of the human-crafted training datasets in GSM8K and MATH, the FOBAR and Self-Verification agents are limited in number of unique synthetic problems they generate.

Training Details: We use the DeepSeek-Math-7B model (Shao et al., 2024) as the student model

[15]FOBAR and Self-Consistency agents use fixed values in existing problems, and are limited in the number of unique problems which they generate. For these, we include as many unique problems as are available.

in the FLAMES framework. We do full-parameter fine-tuning on Amazon EC2 P4 instances[16] using Deepspeed Zero3 (Rajbhandari et al., 2020) for distributed training across 8 A100 GPUs. We train using a batch size of 4 for 5 epochs, saving 10 total checkpoints. We implement training using the SWIFT library[17] (Zhao et al., 2024), using default hyperparameters.

Evaluation: We evaluate fine-tuned student models using the Qwen2.5-Math evaluation framework[18] (Yang et al., 2024b). We report scores of the checkpoint with the highest average score on GSM8K and MATH. All models are evaluated using the same system prompt recommended by the Qwen2.5-Math framework (see Appendix F for the system prompt). The evaluation decoding was done with temperature 0, allowing for a maximum of 2,048 new tokens. Answer extraction and matching is handled by the Qwen2.5-Math framework, which is borrowed from the release of the MATH dataset (Hendrycks et al., 2021).

B Related Works

B.1 LLM Math Reasoning

The ability to solve math word problems is considered a key measure of large language model performance (Cobbe et al., 2021; Hendrycks et al., 2021). Recent large language models have used math reasoning as an important metric to highlight model reasoning capability (Hendrycks et al.; Team et al., 2024; Dubey et al., 2024; Yang et al., 2024b; Guo et al., 2025). Various strategies have been proposed to improve large language model performance on math reasoning, including instruction tuning (Luo et al., 2023; Ding et al., 2024), prompting techniques (Wei et al., 2022; Chia et al., 2023), and solution construction (Toshniwal et al., 2024b). Our work focuses on evaluating problem synthesis techniques for improving LLM math reasoning.

B.2 Data Synthesis for LLM Math Reasoning

Existing works in evaluating synthetic data agents for improving LLM math reasoning have shown that various methods for generating synthetic math word problems can improve downstream performance of underlying student models fine-tuned on synthetic problems (Chen et al., 2025; Kim et al., 2024; Hase et al., 2024; He et al., 2024; Lu et al., 2024). MetaMathQA (Yu et al., 2023a) rewrites questions from multiple perspectives, enabling augmentation without additional knowledge. OpenMath-Instruct-2 uses few-shot LLM prompting to generate novel problems (Toshniwal et al., 2024a). Other works introduce multi-step approaches to synthesizing questions (Mitra et al., 2024; Luo et al., 2023; Liu et al., 2024b).

Agent	Work
In-Domain Practice	
Few-Shot	(Toshniwal et al., 2024a)
Paraphrasing	(Yu et al., 2023a)
Key Concepts	(Huang et al., 2024a)
Seeded Key Concepts	(Huang et al., 2024a)
In-Domain Complexity Enhancing	
Suggester-Editor	(Mitra et al., 2024)
IQC	(Liu et al., 2024b)
Robustness Enhancing	
Ask Me Anything	(Mitra et al., 2024)
Self-Verification	(Yu et al., 2023a)
FOBAR	(Yu et al., 2023a)
Distraction Insertion	Our
Out-Of-Domain Enhancing	
Question Fine-Tuning (QFT)	(Ding et al., 2024)
Taxonomy Key Concepts	Our

Table 7: List of all agents evaluated in the FLAMES framework.

C Data Synthesis Agents for LLM Math Reasoning

C.1 Agent Descriptions

We now categorize 4 types of approaches and detail the 10 existing agents compared in the FLAMES framework (see Section 3). For reference, we include citations and agent types in Table 7.

In-Domain Practice: In-domain practice agents are designed to increase the amount of in-domain data available for model training. The goal of in-domain practice agents is to synthesize a large amount of math word problems which are similar in complexity, topic, and grade-level to problems in the training data. The **Few-Shot** agent, proposed in (Toshniwal et al., 2024a), uses existing problems as in-context examples to generate a novel problem. The **Paraphrasing** agent ((Yu et al., 2023a)) paraphrases existing problems. (Huang et al., 2024a) introduce the use of key concepts as an intermediate problem representation for synthesizing novel problems. We evaluate a **Key Concepts** agent, which first extracts key concepts from an existing problem, and then uses those key concepts to generate novel problems. The related **Seeded Key**

[16] https://aws.amazon.com/ec2/instance-types/p4/

[17] https://github.com/modelscope/ms-swift

[18] https://github.com/QwenLM/Qwen2.5-Math

Concepts agent uses both extracted key concepts and the original existing problem to guide the generation of novel problems.

In-Domain Complexity Enhancing: There have been several proposed agents which aim to increase the complexity of existing math word problems. We select two recent agents to evaluate this type of strategy in our work, although other similar strategies have been proposed including Evol-Instruct (Luo et al., 2023; Li et al., 2024). The **Suggester-Editor** agent first suggests edits for problems to introduce additional reasoning steps, then synthesizes a novel problem by making those edits (Mitra et al., 2024). The **Iterative Question Composing (IQC)** agent turns an existing problem into a subproblem of a more complex synthetic problem (Liu et al., 2024b). Both the Suggester-Editor and IQC agents may repeat their processes multiple times.

Robustness Enhancing: Some proposed agents aim to alter existing problems to increase the robustness of the student model's reasoning process. These agents include the **Ask Me Anything** agent (Mitra et al., 2024), which converts an existing problem and solution into a statement, then prompts the teacher model to ask questions about the statement. Similarly, the **Self-Verification** agent (Yu et al., 2023a) converts an existing problem and solution into a statement, then replaces a number in the problem with a variable and asks the model to reason backwards to obtain the value of the variable. The **FOBAR** agent (Yu et al., 2023a) is similar to Self-Verification, except instead of having the teacher model rewrite the problem and solution into a statement before variable replacement, there is a template sentence added to the end of the problem which includes the problem's answer and the resulting question[19].

Out-of-Domain Enhancing: Recently, (Ding et al., 2024) proposed a method for synthesizing math reasoning data by using only a "small-size" (e.g. 7B) open-source model without a complex augmentation strategy. Their method revolves around the **Question Fine-Tuning (QFT)** process, which lightly fine-tunes a 7B math specialist model on the GSM8K and MATH training datasets, then prompts the fine-tuned model with only the system prompt to synthesize novel math word problems. We replicate their proposed process using Qwen2.5-Math-7B-Instruct (Yang et al., 2024b) to create the QFT agent, designed to enhance general math reasoning performance. We call this an out-of-domain agent, as the agent prompt is not seeded with any specific in-domain problem.

[19] See Appendix F for specific prompts

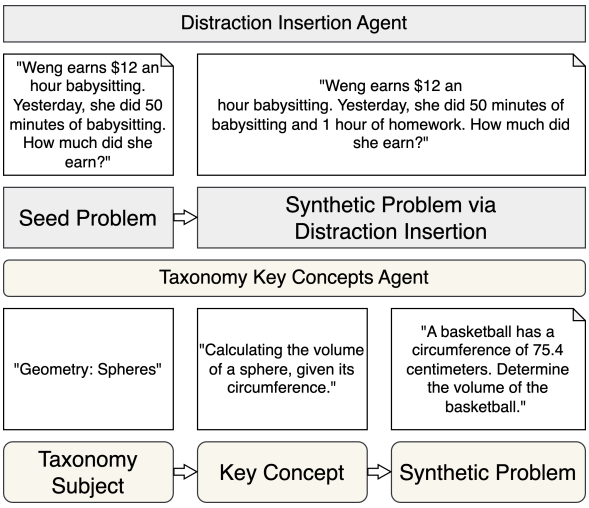

Figure 4: Examples of our novel Distraction Insertion and Taxonomy-Based Key Concepts data agents.

Agent	Prompt Location
In-Domain Practice	
Few-Shot	(Toshniwal et al., 2024a) Appendix D.2
Paraphrasing	(Yu et al., 2023a) Section 3.2
Key Concepts	Appendix F
Seeded Key Concepts	Appendix F
In-Domain Complexity Enhancing	
Suggester-Editor	(Mitra et al., 2024) Section 2
IQC	(Liu et al., 2024b) Figures 5, 6
Robustness Enhancing	
Ask Me Anything	(Mitra et al., 2024) Section 2
Self-Verification	(Yu et al., 2023a) Section 3.3
FOBAR	(Yu et al., 2023a) Section 3.3
Novel Agents	
Taxonomy Key Concepts	Prompts: (Adler et al., 2024) Appendix B.5, Taxonomy: Section 3
Distraction Insertion	Appendix F

Table 8: Locations of prompts used for agents in the FLAMES framework.

Novel Agents are described in Section 3. Examples of synthetic problems generated by novel agents can be seen in Figure 4.

C.2 Agent Prompt Locations

In Table 8 we provide the locations of each prompt used in the FLAMES framework. Prompts for the novel Distraction Insertion agent can be found in Appendix F.

		In-Domain		OOD	Robustness		Competition	
Model		**GSM8K**	**MATH5K**	**College Math**	**Distraction**	**GSMPlus**	**Olympiad Bench**	**Average**
DeepSeek-Math-7B (Base)		64.2	36.2	15.9	17.6	22.6	5.3	28.8
DeepSeek-Math-7B-Instruct		82.9	51.7	33.9	60	70.1	13.8	50.5
Dataset	**Dataset Size**	**Original Dataset**						
GSM8K/MATH	15K	70.1	34.9	33.9	60	70.1	13.8	44.6
NuminaMath	860K	81.4	53.1	36.6	61.9	70.2	20.4	52.3
MetaMathQA	400K	78.5	41.6	30.9	53.4	65.3	9.9	45.2
OrcaMath	200K	77.9	42.3	29.1	63.4	67.8	13.5	46.1
OpenMathInstruct2	600K	83.5	55.3	36.8	64.4	72.3	19.6	53.5
MMIQC	1M	77.7	42.3	31.6	53.8	63.4	11.9	45.4
ScaleQuest	1M	86.4	64.6	42.7	71.1	76.7	27.6	59.6
Dataset	**Dataset Size**	**Refreshed with Qwen2.5-Math-7B solutions**						
OpenMathInstruct2	600K	84.5	66.4	40.7	64.7	72.9	30.5	59.0
MetaMathQA	150K	85.0	55.8	36.7	64.1	72.9	21.6	54.4
OrcaMath	200K	84.5	52.8	36.9	68.1	74.9	19.7	53.8
MMIQC	220K	84.8	61.5	38.4	63.7	73.6	24.4	56.5
ScaleQuest	1M	89.4	66.0	41.4	71	78.3	25	60.0
Dataset	**Dataset Size**	**FLAMES Data**						
FLAMES_large	1M	89.2	68.3	41.9	76.1	79.4	29.8	61.7
FLAMES_xl	1.5M	87.7	67.7	47.2	80.3	83.2	43.3	65.8

Table 9: Comparison of open-source math reasoning datasets with the FLAMES_large and FLAMES_xl datasets. DeepSeek-Math-7B is fine-tuned using unique problems from each dataset, alongside refreshed (R) solutions (using Qwen2.5-Math-7B-Instruct) for fair comparison. Resulting models are evaluated on in-domain, out-of-domain generalization (OOD), robustness, and competition benchmarks.

		In-Domain		OOD	Robustness		Competition	
Student Model	Dataset	GSM8K	MATH5K	College Math	Distraction	GSMPlus	Olympiad Bench	Average
DeepSeek-Math-7B	ScaleQuest Refreshed	**89.4**	66.0	41.4	71.0	78.3	25.0	60.0
DeepSeek-Math-7B	FLAMES_large	89.2	**68.3**	41.9	76.1	79.4	29.8	**61.7**
Qwen2.5-Math-7B	ScaleQuest Refreshed	**93.9**	80.7	46.6	78.4	83.6	41.3	69.2
Qwen2.5-Math-7B	FLAMES_large	93.5	**81.4**	47.1	82.9	84.5	40.9	**69.5**
Mathstral-7B	ScaleQuest Refreshed	88.4	65.7	40.0	69.1	76.5	24.6	59.0
Mathstral-7B	FLAMES_large	**89.2**	**68.1**	39.9	76.5	79.3	29.2	**61.1**
Mistral-7B-v0.3	ScaleQuest Refreshed	84.7	59.3	36.7	64.3	73.5	19.7	54.8
Mistral-7B-v0.3	FLAMES_large	**86.4**	**63.6**	37.0	73.8	76.3	24.1	**57.5**
Qwen2.5-14B	ScaleQuest Refreshed	93.1	75.6	44.6	79.5	83.5	34.2	66.2
Qwen2.5-14B	FLAMES_large	**93.3**	**76.7**	44.9	81.5	84.0	37.9	**67.4**

Table 10: Comparison of FLAMES_large and ScaleQuest across diverse student models. DeepSeek-7B refers to DeepSeek-Math-7B, Mistral-7B refers to Mistral-7B-v0.3, Qwen2.5M-7B refers to Qwen2.5-Math-7B, FLAMES (L) refers to FLAMES_large, ScaleQuest (R) refers to ScaleQuest problems with Qwen2.5-Math-7B-Instruct solutions. Both datasets contain 1M problems.

D Additional Experiments and Detailed Results

D.1 Comparison with Open-Source Synthetic Datasets

Full results from Section 5.3 are given in Table 9 and Section 5.4 are given in Table 10.

D.2 Combining Data Generation Agents

In this section, we discuss our study of mixing data from different agents (Table 11). First, we analyze if mixing Taxonomy Key Concepts agent (having best out-of-domain performance) can improve out-of-domain reasoning for the best complexity-enhancing Suggester & Editor agent. By mixing data from Suggester & Editor and Taxonomy Key Concepts agents in equal proportion (mixture-A), we observe improved CollegeMath score of 41.6 exceeding both the agents. However, we observe slight drop in other metrics as compared to the Suggester & Editor agent. Second, we investigate impact of different mixing proportions (mixture-A, mixture-B and mixture-C) for Suggester & Editor and Taxonomy Key Concept agent. We observe that mixture-C (75/25 split) yields a better trade-off[20] as there is big drop of In-Domain metrics for mixture-A and OOD metrics for mixture-B.

Third, we investigate whether enhancing di-

[20]We observe drop in Competition level metrics for mixture-C, but we recover that with our mixture-D

		In-Domain			OOD	Robustness		Competition	
	Agents	**GSM8K**	**MATH5K**	**Average**	**College Math**	**Distraction**	**GSMPlus**	**Olympiad Bench**	**Average**
	Distraction Insertion	83.3	59.4	71.3	39.6	72.4	73.5	24.7	56.1
	IQC	86.0	59.9	73.0	38.7	72.1	75.7	24.4	56.9
	Suggester-Editor	85.3	61.0	73.2	39.4	72.1	75.9	25.2	57.4
	Taxonomy Key Concepts	77.1	56.1	66.6	40.9	64.7	67.3	21.8	52.6
	QFT	79.0	57.5	68.3	40.5	67	68.6	23.6	53.8
Mixture ID	**Agent Mixture**								
A	Suggester-Editor (50%) Taxonomy Key Concepts (50%)	82.9	58.8	70.9	41.6	69.0	73.5	23.7	56.1
B	Suggester-Editor (90%) Taxonomy Key Concepts (10%)	84.8	60.0	72.4	41.1	70.7	74.8	24.4	57.0
C	Suggester-Editor (75%) Taxonomy Key Concepts (25%)	84.8	59.4	72.1	41.4	70.4	74.6	23.4	56.7
D	Suggester-Editor (50%) IQC (25%) Taxonomy Key Concepts (25%)	84.9	59.6	72.3	40.6	69.5	74.6	25.2	57.0
FLAMES Small	Suggester-Editor (50%) Distraction Insertion (10%) IQC (20%) Taxonomy Key Concepts (20%)	85.2	60.0	72.6	41.4	72.2	74.7	26.1	57.5

Table 11: Results of underlying DeepSeek-Math-7B student model after fine-tuning on 150K problems of various agent data mixtures, alongside results using 150K from individual agents in each mixture.

versity by introducing a different complexity-enhancing IDC agent further improves the performance. For mixture-D, we keep same 75/25 proportion of complexity-enhancing and out-of-domain enhancing, by reducing Suggester & Editor proportion to 50% and including 25% of IQC. With mixture-D, we recover OlympiadBench performance with slight drop in OOD metrics as compared to mixture-C. Fourth, we explore whether adding small proportion (10%) of robustness-focused agents improves overall performance. We observe that including Distraction Insertion (mixture FLAMES_Small) further boosts performance, leading to a mixture with best GSM8K (85.2), Math (60.0), OlympiadBench (26.1) and Average score (57.5). These findings establish that mixing our novel Distraction Insertion agent data can bring significant improvement to the student model's performance.

With our FLAMES_Small mixture, we observe balanced performance across the five evaluation datasets, with performance surpassing all 12 agents on CollegeMath and Olympiad-Bench.

D.3 Solvability Filtering

We find in Section 5.1 that the recently-proposed solvability filtering strategy (Ding et al., 2024) for quality control of synthetic math problems does not lead to performance gains from fine-tuned student models. We hypothesize that this is because the solvability filtering strategy also removes solvable problems from the synthetic data pool. To test this hypothesis we use the human-crafted MATH500 (Hendrycks et al., 2021) dataset, filtering for solvability. We observe in Table 12 that the solvability filter removes 30% of the real problems, with a higher percentage of more difficult problems being removed (e.g. 50% of level 5 problems). These results indicate that solvability filtering is unreliable.

Difficulty Level	Deemed Solvable	Total	% Solvable
1	37	43	86.00%
2	75	90	83.30%
3	81	105	77.10%
4	88	128	68.80%
5	68	134	50.70%
All	349	500	69.80%

Table 12: Number of real problems in the MATH500 evaluation dataset (Hendrycks et al., 2021) which were deemed unsolvable by the solvability filter proposed in (Ding et al., 2024). We show in Section 5.1 that filtering synthetic problems for solvability does not lead to performance gains compared with simpler quality control measures.

E Additional Information

Landscape of recent Math data synthesis: We include a larger version of Figure 2 for reference in Figure 5, further highlighting the need for the unified math data synthesis evaluation environment provided by the FLAMES framework.

Table of models and datasets used: We also include tables giving an overview of all models and datasets used, then a table of all experiments and the locations of their corresponding results. Table 13 details all models used in the FLAMES framework, as well as models used in our controlled experiments. Table 14 details all datasets used in the FLAMES framework. Table 15 details all con-

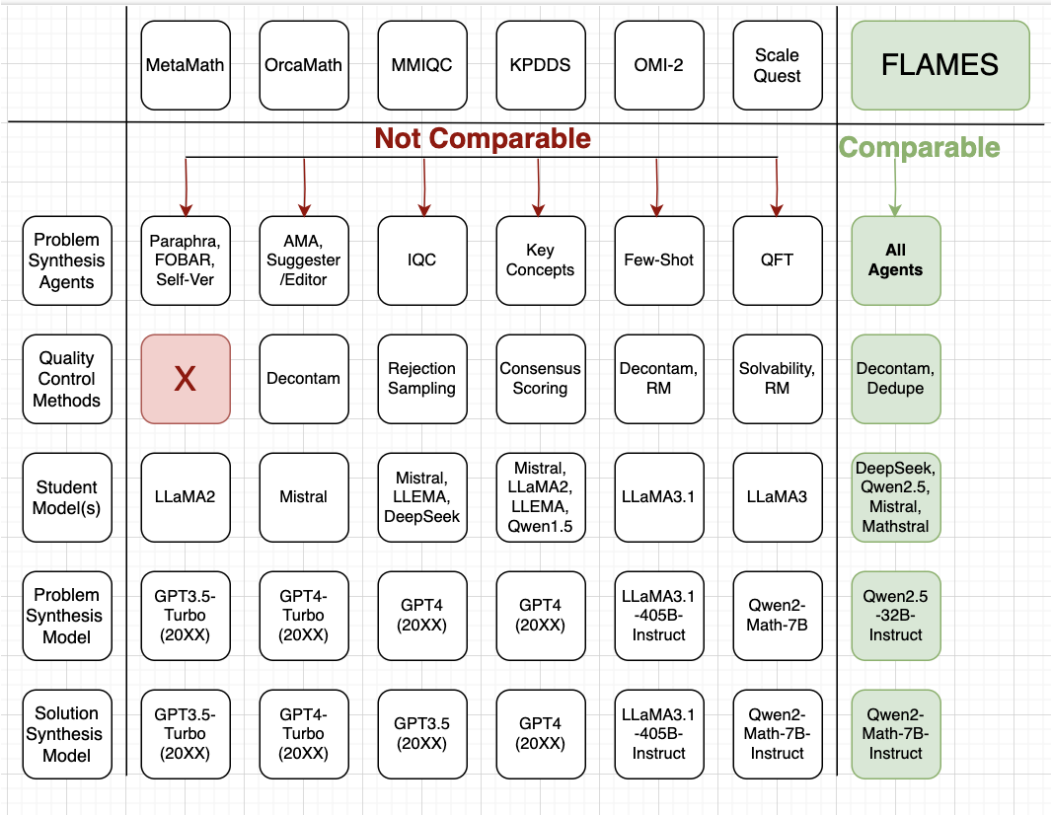

Figure 5: The full landscape of several recently-proposed math data synthesis works. Each work uses different synthesis models, student models, and quality control measures, making comparison of problem synthesis agents impractical.

Model	Use Case	Relevant Sections
Qwen2.5-32B-Instruct (Yang et al., 2024b)	FLAMES Problem Generation	Section 2
Qwen2.5-Math-7B-Instruct (Yang et al., 2024b)	FLAMES Solution Generation	Section 2
DeepSeek-Math-7B (Shao et al., 2024)	FLAMES Student Model	Section 2
DeepSeek-v2.5 (Liu et al., 2024a)	Ablation Problem Generation	Table 5
DeepSeek-Math-7B-RL (Shao et al., 2024)	Ablation Solution Generation	Table 5
Qwen2.5-Math-7B (Yang et al., 2024b)	Ablation Student Model	Table 4
Qwen2.5-14B (Yang et al., 2024b)	Ablation Student Model	Table 4
Mathstral-7B (Mistral AI, 2025)	Ablation Student Model	Table 4
Mistral-7B-v0.3[21]	Ablation Student Model	Table 4

Table 13: Overview of all models used in the FLAMES framework, as well as models used in all controlled experiments (ablations).

Dataset	Use Case	Relevant Sections
GSM8K (Cobbe et al., 2021)	FLAMES Seed Problems, Evalation	Section 2
MATH (Hendrycks et al., 2021)	FLAMES Seed Problems, Evalation	Section 2
GSMPlus (Li et al., 2024)	FLAMES Evaluation	Section 2
CollegeMath[22]	FLAMES Evaluation	Section 2
OlympiadBench[23]	FLAMES Evaluation	Section 2
NuminaMath (LI et al., 2024)	Open-Source Baseline	Table 3
MetaMathQA (Yu et al., 2023a)	Open-Source Baseline	Table 3
OrcaMath (Mitra et al., 2024)	Open-Source Baseline	Table 3
OpenMathInstruct2 (Toshniwal et al., 2024a)	Open-Source Baseline	Table 3
MMIQC (Liu et al., 2024b)	Open-Source Baseline	Table 3
ScaleQuest (Ding et al., 2024)	Open-Source Baseline	Table 3

Table 14: Overview of all datasets used in the FLAMES framework.

Experiment Description	Table(s)
Comparison of Synthetic Data Quality Control Strategies	Table 1
Comparison of Existing and Novel Data Agents	Table 2
Comparison of FLAMES Datasets with Existing Math Datasets	Table 3, Table 9
Comparison of FLAMES_large with Best Existing Dataset	Table 4
Comparison of Different Problem and Solution Generation Models	Table 5
Comparison of Different Data Agent Mixtures	Table 11
Solvability Filtering on MATH500 Test Set	Table 12

Table 15: Overview of all controlled experiments used to design the FLAMES framework.

trolled experiments run in this paper, along with their sections and results tables.

F Prompts

In this section, we include prompts for agents evaluated in the FLAMES Framework (if not referenced in Table 8).

Problem to Key Concepts Prompt: You are a Maths teacher. I will give you a Maths problem and its solution. Your task is to tell a key concept which a student need to understand to solve this problem. We will use this key concept to create similar problems so that students can learn to solve similar problems. Your response should contain only the key concept.
Make sure you follow below constraints:
1. Your response is about generic concept and does not use specifics like numbers or words or object from the problem, so that the key concept can be used to generate related but different problems.
2. Your response provide granular details so that this response can be independently used to create a similar problem for teaching this key concept.
Below are few examples :
Problem : In how many ways can 5 students be selected from a group of 6 students?
Solution : We can choose 5 students out of a group of 6 students without regard to order in binom{6}{5} = 6 ways.
Key Concept : Number of ways to select some fixed number of items from a group of large number of items
Problem : Express frac{3}{20} as a decimal.
Solution : frac{3}{20} = 0.15.
Key Concept : Fraction to decimal conversion
Problem : A bicycle is traveling at 20 feet per minute. What is the bicycle's speed expressed in inches per second?
Solution : There are 12 inches in a foot, so the bicycle is traveling at 12(20)=240 inches per minute. There are 60 seconds in a minute, so the bicycle is traveling at frac{240}{60}=4 inches per second.
Key Concept : A math problem expressed as word problem which requires converting from one unit to another.
Problem : problem
Solution : solution
Key Concept :

Figure 6: Prompt for extracting key concepts from existing GSM8K and MATH problems. Used in both Key Concepts and Seeded Key Concepts agents (Huang et al., 2024a).

Key Concepts Prompt: You are a Maths teacher. I will give you an existing problem and a key concept which a student need to understand to solve this problem. Your task is to create a new Math problem that checks if a student understand this key concept and can use it to solve a Math problem. Do not include the solution in your new Math problem. Your response should contain only the new problem.
Make sure you follow below constraints: 1. The generated problem is grammatically correct, does not make any incorrect assumptions, is self-sufficient and can be solved without any additional information.
Below are few examples:
Key concept : Number of ways to select some items from a pool of large number of items. New Problem : In how many ways can 4 students be selected from a group of 6 students?
Key Concept : Converting the units of a rate to new units New Problem : A person is running a distance of 300 meters in 30 seconds. What is the speed of the person in feet per minute?
Key Concept : Using the triangle inequality to reason about triangles New Problem : A stick 5 cm long, a stick 9 cm long, and a third stick n cm long form a triangle. What is the sum of all possible whole number values of n?
Key Concept : keyconcept New Problem :

Figure 7: Prompt for using key concepts to generate synthetic problem. Used by Key Concepts agent (Huang et al., 2024a).

Seeded Key Concepts Prompt: You are a Maths teacher. I will give you an existing problem and a key concept which a student need to understand to solve this problem. Your task is to create a new Math problem that checks if a student understand this key concept and can use it to solve a Math problem. Do not include the solution in your new Math problem. Your response should contain only the new problem.
Make sure you follow below constraints: 1. The generated problem is grammatically correct, does not make any incorrect assumptions, is self-sufficient and can be solved without any additional information. 2. The generated problem is of similar complexity as the existing problem.
Below are few examples:
Existing Problem : Determine the number of ways to arrange the letters of the word ELLIPSE. Key concept : Number of ways to select some items from a pool of large number of items. New Problem : In how many ways can 4 students be selected from a group of 6 students?
Existing Problem : A bicycle is traveling at 20 feet per minute. What is the bicycle's speed expressed in inches per second? Key Concept : Converting the units of a rate to new units New Problem : A person is running a distance of 300 meters in 30 seconds. What is the speed of the person in feet per minute?
Existing Problem : The lengths of the sides of a non-degenerate triangle are x, 13 and 37 units. How many integer values of x are possible? Key Concept : Using the triangle inequality to reason about triangles New Problem : A stick 5 cm long, a stick 9 cm long, and a third stick n cm long form a triangle. What is the sum of all possible whole number values of n?
Existing Problem : problem Key Concept : keyconcept New Problem :

Figure 8: Prompt for using problem and key concepts to generate synthetic problem. Used by Seeded Key Concepts agent (Huang et al., 2024a).

Distraction Insertion Prompt: Your goal is to hide a misleading detail in the given problem, such that it doesn't change the answer to the problem. The solution to the new problem should be the same as the solution to the original problem. Do not give a solution for the new problem. Do not give solution hints.
Here is an example:
Problem: Natalia sold clips to 48 of her friends in April, and then she sold half as many clips in May. How many clips did Natalia sell altogether in April and May?
Solution: Natalia sold altogether 72 clips in April and May.
New Problem: Natalia sold clips to 48 of her friends in April, and then she sold half as many clips in May. In June, she sold twice as many clips as in April. How many clips did Natalia sell altogether in April and May?
Now hide a misleading detail in the following problem.
Problem: problem
Solution: solution
New Problem:

Figure 9: Prompt used to generate synthetic problems using the novel Distraction Insertion agent (see Section 3).

Solvability Filtering Prompt: Please act as a professional math teacher. Your goal is to determine if the given problem is a valuable math problem. You need to consider two aspects:
1. The given problem is a math problem.
2. The given math problem can be solved based on the conditions provided in the problem (You can first try to solve it and then judge its solvability).
Please reason step by step and conclude with either 'Yes' or 'No'.
Given Problem: {problem}

Figure 10: Prompt used to filter unsolvable synthetic problems (Ding et al., 2024), see Section 4.