

AgSynth: A Multi-Agent Reverse Synthesis Framework for Multimodal Mathematical Reasoning Data Generation

Yaxuan Wang[†], Lingfeng Zeng[†], Jiatu Hu, Guojie Lin,
Boyue Yang, Bing Zhao, Lin Qu, Hu Wei*

Alibaba Group

{kongwang}@alibaba-inc.com

Abstract

Multimodal large language models show significant potential in mathematical reasoning, yet they remain constrained by a scarcity of high-quality training data for visually grounded problems. Existing approaches commonly suffer from narrow coverage, image-text inconsistency, and heavy imitation of seed problems. To address these limitations, this paper proposes AgSynth, a multi-agent collaborative reverse synthesis framework. It first constructs solution logic programmatically and renders the corresponding images, then generates problem statements backward from the visual facts, thereby embedding answer correctness and image-text consistency directly into the generation pipeline. Meanwhile, AgSynth distills the core competencies tested in seed problems and performs divergent reconstruction, enabling synthesized problems to be substantially novel in both structure and visual presentation. Based on this framework, we construct the AgSynth-10K dataset, comprising 10,477 multimodal mathematical reasoning problems that have undergone two-tier quality control, spanning four major domains: geometry, algebra, combinatorics, and number theory. Experiments demonstrate that reinforcement learning training on AgSynth-10K significantly improves model performance on reasoning-intensive benchmarks (WeMath+7.42, LogicVista+9.15), validating the effectiveness of this traceable synthesis paradigm. The framework is publicly available at <https://github.com/Yaxuan7690/AgSynth>.

1 Introduction

The mathematical reasoning capabilities of Multimodal Large Language Models have attracted increasing attention; however, these models continue

to underperform on mathematical problems that require extracting information from visual elements such as geometric figures, function plots, and statistical charts, and performing rigorous reasoning thereupon. A growing body of research suggests that, beyond algorithmic design, the scarcity of high-quality multimodal training data has become one of the critical bottlenecks hindering further breakthroughs (Ahn et al., 2024; Huang et al., 2025; Liu et al., 2025; Sun et al., 2026a; Chen et al., 2021, 2022; Camposampiero et al., 2025; Chen et al., 2025).

The construction of existing multimodal mathematical data faces three core challenges: (1) Limited coverage of mathematical branches—existing synthesis efforts have concentrated on geometric scenarios, leaving multimodal data for branches such as functions, algebra, statistics, and combinatorial reasoning severely underrepresented (Sun et al., 2026b; Huang et al., 2025); (2) Difficulty in ensuring image-text consistency—directly using AI-generated images is prone to geometric distortions, while multi-stage pipelines built on multi-agent collaboration are susceptible to semantic drift across pipeline stages (Luo et al., 2026a; Liu et al., 2024); (3) Synthetic data tends toward superficial imitation—generated instances are often highly similar to seed problems, with only surface-level substitutions in numerical values or phrasing (Qiao et al., 2026; Lu et al., 2024).

To address these challenges, we propose AgSynth, a reverse synthesis framework based on multi-agent collaboration. Its core design proceeds as follows: the underlying examination concept is first extracted from a seed problem and subjected to divergent reconstruction; a deterministic image is then programmatically rendered via Python code; finally, the problem statement and answer are composed in reverse, conditioned on the already-determined image content. This image-first, problem-second design paradigm mechanisti-

*Corresponding author.

[†]These authors contributed equally to this work.

cally improves image-text consistency and answer correctness. Furthermore, the framework is not reliant on domain-specific templates, enabling broad coverage across multiple mathematical branches including geometry, functions, statistical charts, and logical reasoning (Yang et al., 2025; Zhuang et al., 2025; Meng et al., 2025; Deng et al., 2026).

The main contributions of this paper are as follows: (1) We propose AgSynth, a reverse synthesis framework based on multi-agent collaboration. The framework improves image-text consistency through a reverse construction paradigm, mitigates superficial imitation through divergent reconstruction, and achieves broad coverage across multiple mathematical branches. (2) We construct the AgSynth-10K dataset, comprising 10K high-quality multimodal mathematical training instances that have undergone multi-round consistency verification and dual-model veto-based filtering. The framework is publicly released.

2 Related Work

2.1 Multimodal Mathematical Datasets

Existing multimodal mathematical datasets encompass both evaluation benchmarks and training sets (Sun et al., 2026a; Shi et al., 2024; Qiao et al., 2025; Liu et al., 2025; Wang et al., 2024; Yu et al., 2024; Lu et al., 2023; Hendrycks et al., 2021; He et al., 2026; Zou et al., 2025; Shao et al., 2024; Roberts et al., 2025). While these efforts have effectively expanded the scale of available data, their construction paradigms rely primarily on resource aggregation, and the resulting visual diversity is fundamentally constrained by the original sources. Moreover, geometry problems dominate these datasets, whereas coverage of other branches such as functions, statistics, and logic remains markedly insufficient, making it difficult for models trained on them to generalize to non-geometric mathematical tasks.

2.2 Multimodal Mathematical Data Synthesis Methods

To overcome the bottleneck of the collection paradigm, researchers have explored various automated synthesis approaches. AutoGeo and GeoTikzBridge achieve large-scale data construction through geometric clause combination and TikZ code generation, respectively, yet both remain confined to the single branch of geometry. Regarding image-text consistency, mainstream methods

follow a forward pipeline—first determining the problem and then generating the image—where images and problems are produced independently across different stages, making the risk of semantic misalignment difficult to eliminate. In terms of generation diversity, MathGenie leverages back-translation and code verification to derive new problems from seed questions, and AgentMath validates the potential of multi-agent collaboration; however, both target plain-text scenarios and do not involve multimodal synthesis.

In summary, existing methods still face three structural contradictions: the decoupling of image and text generation leads to semantic misalignment, forward-oriented pipelines constrain the novelty of generated problems, and the dominance of a single branch suppresses cross-domain generalization. AgSynth is proposed precisely in response to these challenges.

3 AgSynth Framework

This section introduces the task definition, overall architecture, and module design of AgSynth.

3.1 Task Definition and Framework Overview

Task Definition. Let the seed problem be denoted as $S = (Q_s, I_s, A_s)$, where Q_s is the problem text, I_s is the associated image, and A_s is the ground-truth answer. The objective of AgSynth is to transform S into a new problem $S' = (Q', I', A')$ that satisfies three constraints: (1) *image-text consistency*— Q' and I' are strictly aligned in semantics; (2) *answer correctness*— A' can be deterministically derived from I' and Q' ; and (3) *novelty*— S' differs from S to a certain extent in either visual presentation or problem formulation. The framework realizes this transformation through the following mapping chain:

Extract the Examination Kernel

$$K = f_{\text{kernel}}(S)$$

↓

Generate Divergent Conceptions

$$P = f_{\text{plan}}(K)$$

↓

Programmatically Render the Image

$$I' = f_{\text{render}}(P)$$

↓

Reversely Generate the Problem and Answer

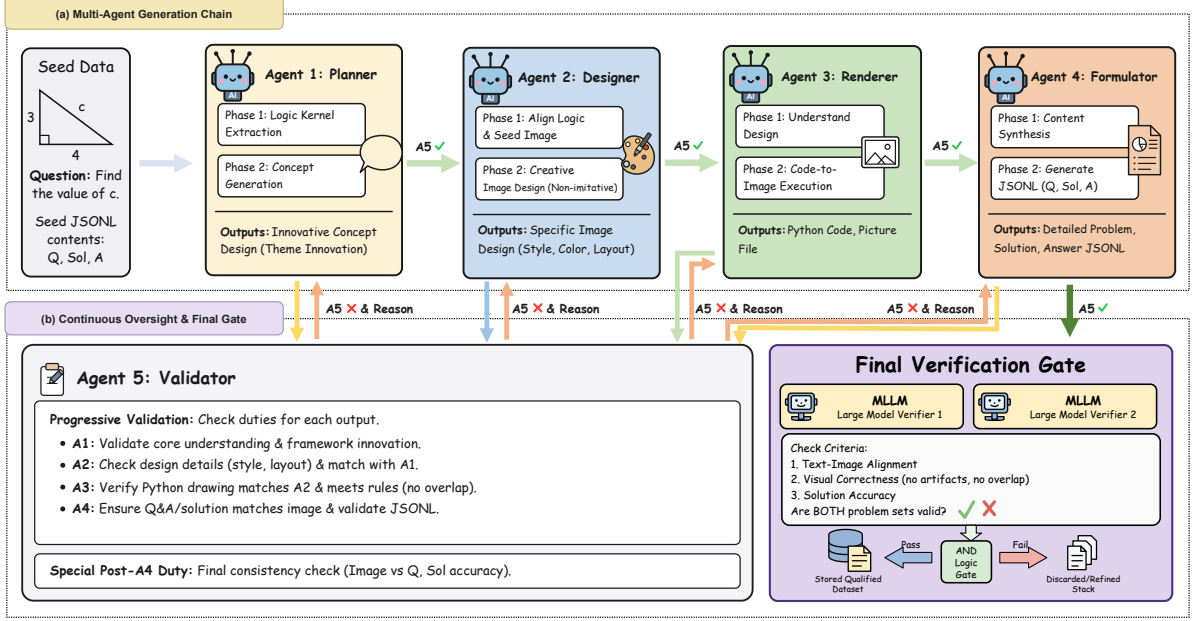


Figure 1: Overall Architecture of the AgSynth Framework

$$(Q', A') = f_{\text{reverse}}(I', K, P)$$

S' is accepted only if $f_{\text{validate}}(Q', I', A') = 1$, where 1 denotes “true” or “pass” in mathematical logic, and 0 denotes “false” or “fail.”

Framework Overview. The core philosophy of AgSynth can be summarized as:

Logical Decoupling $\rightarrow$ *Visual Instantiation* $\rightarrow$
Reverse Alignment $\rightarrow$ *Closed-loop Verification*

Unlike forward synthesis, which generates questions first and then matches images to them, our framework first extracts the kernel and engages in divergent ideation, subsequently renders deterministic images in a programmatic manner, and finally generates questions and answers in reverse based on the solidified visual facts. This design significantly reduces the risk of image-text semantic misalignment at the procedural level. The overall pipeline consists of five agents collaborating sequentially, connected via a structured information protocol, where the output of each stage may flow downstream only after passing verification.

3.2 Agent Design

These agents are connected via a structured information protocol, as illustrated in Figure 1.

Agent 1 (Planner, Logical Conception Agent).

Input: seed problem S .

Core functions: in the first stage, extract the examination kernel K (knowledge points, reasoning

structure, and competency dimensions); in the second stage, perform divergent ideation while preserving the kernel, with specific strategies including transforming the problem’s background scenario, swapping known and unknown quantities, altering the type of visual carrier (e.g., replacing geometric figures with statistical charts), and introducing distractor information.

Output: the examination kernel K and the innovative conception plan P .

Agent 2 (Designer, Visual Semantic Planning Agent). **Input:** conception plan P and seed problem image I_s .

Core function: transform the abstract conception into a structured visual design scheme that specifies the types of graphical elements, spatial layout, colors, and annotation methods.

Output: a structured visual design scheme. Agent 2 focuses on “what to draw” (visual semantic planning), whereas Agent 3 focuses on “how to draw” (executable visual implementation); decoupling the two allows the design intent and the code implementation to be independently optimized and reviewed.

Agent 3 (Renderer, Code Rendering Agent).

Input: the visual design scheme from Agent 2.

Core function: transform the design scheme into executable Python code and render it to produce the image I' . The advantage of programmatic ren-

dering is that the geometric relationships among elements in the image are determined by deterministic program logic, which substantially reduces the probability of geometric distortion and structural ambiguity compared with AI-based image generation. Such deterministic rule-based rendering compensates for the uncontrollability of diffusion models in generating mathematically precise geometric elements.

Output: a precisely rendered mathematical image I' .

Agent 4 (Formulator, Reverse Problem Generation Agent). **Input:** image I' , kernel K , and conception P .

Core function: reversely generate the problem Q' , solution steps, and answer A' based on the established visual facts. Because the problem is generated around the same set of known constraints, image–text consistency and answer correctness are structurally guaranteed.

Output: the complete problem text, solution process, and answer.

Agent 5 (Validator, Quality Audit Agent). **Input:** intermediate outputs and final deliverables from each stage.

Core function: conduct real-time auditing throughout the entire workflow and perform an end-to-end consistency check upon completion of Agent 4.

Output: audit approval or revision feedback. If the audit fails, the system feeds the identified issues back to the corresponding stage to trigger a revision loop (with a retry limit imposed to ensure convergence).

3.3 Quality Control Mechanism

AgSynth employs a two-tier quality control mechanism combining “stage-wise auditing with terminal dual-model veto.” During pipeline execution, Agent 5 performs real-time quality inspection on the output of each stage and triggers revision loops as needed. After all agents have completed their tasks, the system introduces two independent verification models to cross-validate the final output; the sample is discarded if either model identifies any issues. Formally, S' is accepted only if $f_{\text{validate}}(Q', I', A') = 1$. Here, f_{validate} consists of four sub-criteria: image quality inspection (e.g., element occlusion, overlap), image-text semantic correspondence inspection, problem-solving logical rigor inspection, and answer verifiability inspection. A sample is incorporated into the final

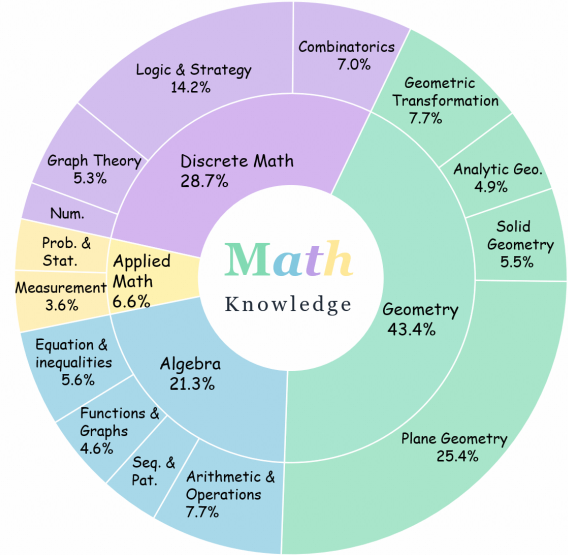


Figure 2: Distribution of Mathematical Topics in AgSynth-10K (Num. = Number Theory 2.2%, Seq. & Pat. = Sequence & Patterns 3.4%, Prob. & Stat. = Probability & Statistics 3.0%, Analytic Geo. = Analytic Geometry)

dataset only if it passes all four criteria.

This mechanism ensures that no single generation is blindly trusted; instead, the combination of deterministic rendering, reverse problem-solving, and multi-stage filtering selects only high-confidence, consistent results (Luo et al., 2026b).

4 AgSynth-10K Dataset

4.1 Dataset Overview

AgSynth-10K is a dataset designed for multimodal mathematical reasoning, comprising 10,477 samples. Each sample is structured in a unified JSON format, including problem images, text, standard answers, and detailed solution processes. Quality is ensured through dual-model cross-validation.

The dataset is constructed from two types of seed resources: 793 internally annotated problems sourced from competitions such as UKMT and AMC, and 500 externally selected problems from We-Math2.0-Standard. These seeds generated 7,074 and 3,403 samples, respectively, all adhering to unified generation protocols and quality control standards.

As shown in Figure 2, AgSynth-10K covers four mathematical domains and 14 knowledge categories. Geometry accounts for the largest proportion (43.4%), focusing on triangles, circles, etc.; Discrete Mathematics (28.7%) emphasizes logic and strategy; Algebra (21.3%) includes basic op-

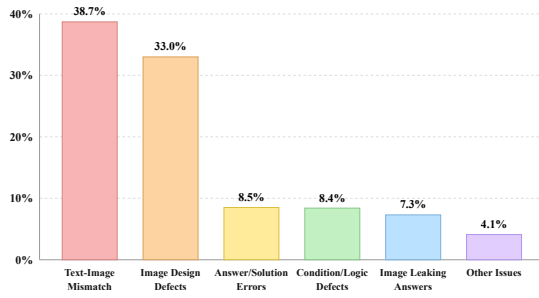


Figure 3: Error Attribution Distribution in Failed Samples from AgSynth-10K (Automated Classification via Gemini-3.1-Pro)

erations; and Applied Mathematics (6.6%) mainly involves probability and measurement. This ensures comprehensive topic coverage and diverse problem generation.

4.2 Quality Assurance

As described in Section 3.3, AgSynth employs a two-tier quality control mechanism consisting of “stage-wise auditing + terminal dual-model veto.” In the terminal veto stage, two independent verification models perform cross-validation on each question across four dimensions: image clarity, image accuracy, question completeness, and answer correctness. A sample is discarded if either model identifies any issue. Out of the 18,910 candidate questions in total, 10,477 passed the automated quality inspection, yielding a pass rate of approximately 55.4%.

To systematically analyze the underlying causes of errors in the filtered questions, this study employs gemini-3.1-pro to perform automated error attribution classification on the quality inspection feedback. The classification system is grounded in the free-clustering results of 50 sample questions during the pilot phase, and through manual induction and consolidation, an error classification framework comprising six major categories was ultimately established. The classification results are shown in the Figure 3. The error distribution exhibits markedly non-uniform characteristics. The results indicate that image generation and its semantic alignment with text constitute the primary bottlenecks of the current pipeline. In contrast, errors pertaining to the content correctness dimension account for a significantly smaller and more manageable proportion, which reflects that AgSynth has already attained a high level of reliability in the

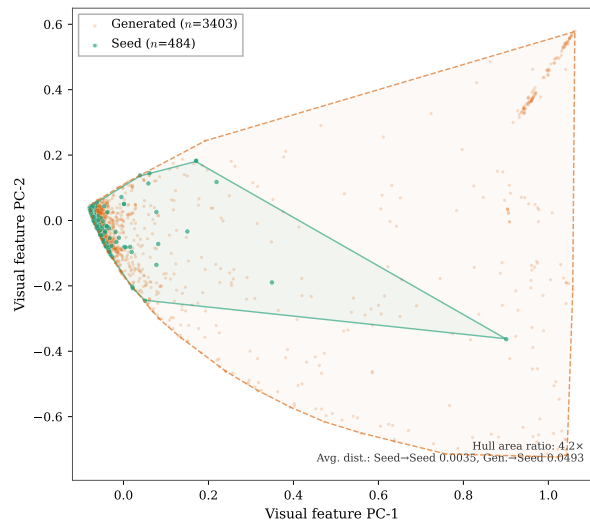


Figure 4: Quantitative Analysis of Visual Diversity via PCA Projection and Convex Hull Coverage

logical derivation and answer generation stages.

Beyond automated quality inspection, we further employed manual sampling to validate the final data quality. From the 10,477 questions that passed quality inspection, 500 were randomly selected for manual review, using “the question can be solved normally without obvious errors” as the passing criterion. The manual pass rate reached 96.8%, with the few failing samples primarily attributable to implicit omissions of problem-solving conditions and answer leakage caused by visual cues in the images.

4.3 Visual Diversity Analysis

This paper evaluates the image diversity of AgSynth-10K from two perspectives: qualitative case analysis and quantitative feature distribution. Due to space limitations, qualitative examples illustrating the visual diversity and evolution of synthetic samples derived from the same seed are provided in Appendix C (Figure 7). While maintaining mathematical traceability, these samples introduce diverse graphical carriers, node styles, path morphologies, and spatial layouts. This demonstrates that the generation process is a visual reconstruction centered on core mathematical constraints, rather than mere replication or local modification.

To quantify the extent of visual expansion, we extracted visual feature vectors—composed of color histograms and grayscale structural descriptors—from 484 seed problem images and 3,403 synthesized problem images, computed their pairwise cosine distances, and projected the results into a

two-dimensional space via PCA. As shown in Figure 4, the convex hull of the synthesized problems (orange) covers a significantly larger area than that of the seed problems (green). Specifically, the average nearest-neighbor distance among seed problems is 0.0035, whereas the average distance from each synthesized problem to its nearest seed problem is 0.0493—approximately 14 times the former. This gap indicates that the synthesized problems achieve substantial expansion in the visual feature space rather than merely imitating the seeds.

In summary, while inheriting the mathematical relationships of the seed problems, the synthesized problems in AgSynth-10K achieve controlled expansion in graphical structure and visual layout, thereby helping to enhance the generalization capability of multimodal mathematical reasoning models.

5 Experiments

5.1 Experimental Setup

Training Data. We use 10K multimodal mathematical training samples (AgSynth-10K), synthesized by the AgSynth framework, as the reinforcement learning training set.

Training Configuration. Building on Qwen3-VL-8B(Team, 2025) as the base model, we perform reinforcement learning fine-tuning with GRPO. Detailed prompt designs, hyperparameter settings, and other training details are provided in the Appendix.

Evaluation Benchmarks. We select four multimodal mathematical reasoning benchmarks that cover different capability dimensions: WeMath, MathVision(Wang et al., 2024), MathVerse-Vision(Zhang et al., 2025), and LogicVista(Xiao et al., 2024). All models are evaluated using identical prompts and inference parameters to ensure the comparability of results.

Reference Baselines. Our core controlled experiments compare the performance of the Qwen3-VL-8B-Instruct base model, Qwen3-VL-235B-A22B-Instruct, and the model trained on AgSynth-10K. For additional reference, we also include results from the closed-source models Gemini-2.5-Flash(Comanici et al., 2025) and GPT-5.2-Instant(OpenAI, 2025). All reported results are based on Pass@1 accuracy.

5.2 Validation of the Effectiveness of Data Synthesized by the AgSynth Framework

Table 1 presents the performance of each model across four multimodal mathematical reasoning benchmarks. The analysis reveals the following key findings:

Synthetic data-driven leap in reasoning capability. After training on AgSynth-10K, Qwen3-VL-8B achieves substantial improvements on reasoning-intensive benchmarks: WeMath increases from 54.29 to 61.71 (+7.42), and LogicVista rises from 54.91 to 64.06 (+9.15). With only 10K synthetic samples, an open-source 8B model performs on par with the closed-source large model Gemini 2.5 Flash on LogicVista (64.06 vs. 65.18), and substantially surpasses GPT-5.2-Instant on both benchmarks (WeMath 47.24, LogicVista 56.92). These results demonstrate that the construction paradigm of high-quality synthetic data can significantly narrow the reasoning capability gap between small open-source models and large closed-source models. Furthermore, we report the performance of Qwen3-VL-235B-A22B-Instruct as a reference upper bound. The results indicate that on the LogicVista benchmark, the absolute gap between the 8B model and this ultra-large-scale model is narrowed to only 4.04 points, demonstrating exceptional data efficiency.

Gains are concentrated in reasoning ability rather than visual perception. Although the visual encoder is updated synchronously with the language model during training, the model’s performance on MathVision (52.70 $\rightarrow$ 53.72) and MathVerse-Vision (56.62 $\rightarrow$ 56.50) remains essentially on par with the base model. This indicates that the gains from AgSynth-10K primarily stem from the construction of high-order logical chains and the internalization of mathematical reasoning capabilities, rather than from significant improvements in vision-related abilities. This phenomenon suggests that enhancements in reasoning ability can exist independently of improvements in visual perception, and it also points to a direction for future research: to further synergistically enhance perceptual capabilities, subsequent work could explore sample construction strategies targeted at visual perception abilities.

Table 1: Performance comparison on multimodal mathematical reasoning benchmarks. All scores are expressed in percentage (%). We evaluate the effectiveness of fine-tuning the Qwen3-VL-8B-Instruct base model with AgSynth-10K. Scores in **bold** indicate an improvement over the corresponding base model.

Method	WeMath	MathVision	MathVerse	LogicVista
<i>Closed-source Models</i>				
Gemini-2.5-Flash	69.50	49.41	61.02	65.18
GPT-5.2-Instant	47.24	49.05	64.26	56.92
<i>Open-source Models</i>				
Qwen3-VL-8B-Instruct (Base)	54.29	52.70	56.62	54.91
Qwen3-VL-235B-A22B-Instruct	69.00	70.00	70.30	68.10
+ AgSynth-10K (Ours)	61.71	53.72	56.50	64.06

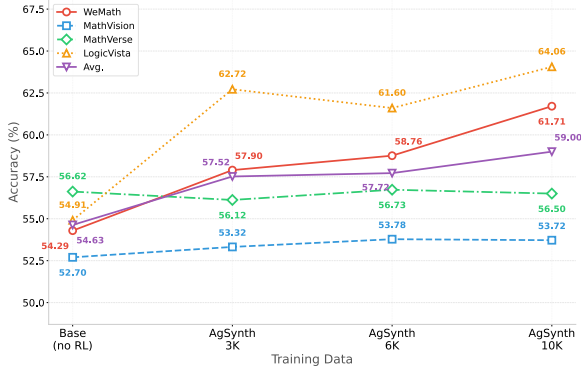


Figure 5: Impact of AgSynth-10K Training Data Scale on Multimodal Math Reasoning Performance

5.3 Analysis of the Impact of Synthetic Data Scale on Reasoning Performance

To investigate the effect of synthetic data scale, we conduct progressive scaling experiments on random subsets of AgSynth-10K (3K/6K/10K), with results shown in Figure 5. Overall, AgSynth data yields consistent gains on reasoning-intensive benchmarks, while performance on vision-perception benchmarks remains stable. Further analysis reveals the following observations:

Reasoning capabilities continue to improve with data scale. Both WeMath and LogicVista exhibit steady gains as the data scale expands (WeMath: +7.42; LogicVista: +9.15), validating that AgSynth-synthesized data effectively enhances multi-step reasoning and logical planning capabilities, with gains continuously accumulating as the scale grows.

Limited Gains on Perception-Oriented Benchmarks. Performance on MathVision and MathVerse remains largely stagnant across different

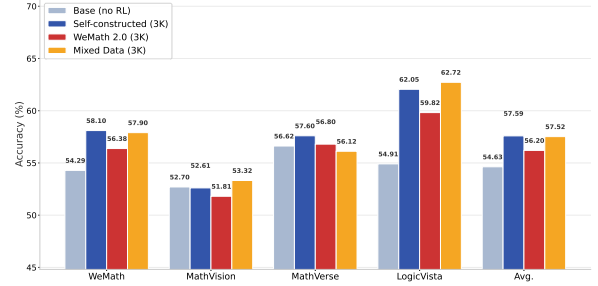


Figure 6: Impact of Seed Question Sources on Performance Across Multimodal Benchmarks

scales. This suggests that these benchmarks are highly sensitive to fine-grained visual parsing and extreme-resolution OCR, whereas the improvements from AgSynth-10K are primarily concentrated on logical reasoning chains. This observation indicates that reasoning-focused data alone may not fully overcome inherent perceptual bottlenecks, highlighting the necessity of integrating targeted perceptual enhancement strategies in the future.

High data efficiency: a small amount of data can significantly stimulate reasoning capabilities. With only 3K samples, it surpasses the base model by +3.61 (WeMath) and +7.81 (LogicVista), demonstrating that AgSynth can achieve substantial gains in reasoning capability under extremely low data overhead.

5.4 Analysis of Seed Question Sources

As the input to the AgSynth synthesis pipeline, the quality and domain distribution of seed questions directly influence the capability orientation of the generated data. To evaluate AgSynth’s adaptability to seed questions from different sources, we

Table 2: Ablation study results. Consistency: pass rate of image-text semantic matching; Novelty: level of innovation of synthesized problems relative to seed problems (1 = pure surface rewriting, 3 = moderate restructuring, 5 = conceptual-level innovation); Reliability: pass rate of problems being directly usable for teaching. "w/o DualVerify" denotes the full set of samples prior to filtering, while the remaining results are obtained after dual-model quality verification.

Configuration	Consistency (%)	Novelty (1-5)	Reliability (%)
AgSynth (Full)	66.0	3.14	64.0
w/o Reverse	52.0	1.57	46.0
w/o Divergent	62.0	1.10	38.0
w/o DualVerify	37.5	1.96	39.6

compared the training effects of two sources under the same data volume of 3K, with results shown in Figure 6. Notably, WeMath 2.0 Standard is an independent training question bank with no overlap with the WeMath evaluation set, ensuring no data leakage. Further analysis yields the following observations.

AgSynth produces significant positive gains across seed questions from different sources. After being processed by AgSynth, seed questions from both sources outperform the base model (self-constructed: Avg. +2.96; WeMath 2.0: Avg. +1.57), validating the framework’s general adaptability to heterogeneous seed question sources.

The quality and domain coverage of seed questions substantially affect synthesis performance. Self-constructed seed questions achieve superior results on MathVerse (+0.80) and LogicVista (+2.23), with an average gain that is 1.27 points higher. This indicates that seed questions with greater domain diversity and higher quality can more fully unleash the potential of the synthesis framework, thereby generating training data with stronger generalization capability.

Multi-source fusion achieves dual gains in broad coverage and domain depth through complementary strengths. While elevating the average to 57.52, the mixed data peaks in LogicVista and MathVision, proving that integration effectively balances domain coverage and performance stability across complex tasks.

5.5 Ablation Study

To verify the independent contribution of each core module, we designed three ablation variants (all using Claude Opus 4.7): w/o Reverse, w/o Divergent, and w/o DualVerify. Each variant generated 50 samples, blindly evaluated by 3 graduate students with LLM assistance across Consistency, Novelty, and Reliability (Table 2).

w/o Reverse: Consistency drops to 52.0% (−14.0 pp) and Novelty to 1.57, as the forward pipeline produces conservative problems to ease figure generation.

w/o Divergent: Novelty falls to 1.10 and Reliability to 38.0% (even below w/o Reverse’s 46.0%), suggesting that superficial rewriting disrupts original constraints, whereas deep reconstruction rebuilds the logical chain for stronger self-consistency.

w/o DualVerify: Both Consistency and Reliability reach their lowest values (37.5% and 39.6%), with gaps of +28.5 pp and +24.4 pp versus the full framework, confirming the effectiveness of terminal-state filtering.

6 Conclusion

This paper proposes AgSynth, a reverse synthesis framework based on multi-agent collaboration, which addresses the core deficiencies of existing multimodal mathematical data construction paradigms from three dimensions: it overcomes the coverage limitations of geometry-only scenarios through a domain-agnostic universal pipeline; it adopts a reverse construction process that first renders images programmatically and then derives problems from the images, improving image-text consistency at the paradigm level; and it drives synthesis innovation through examination kernel extraction and divergent reconstruction, effectively suppressing trivial imitation. Experiments demonstrate that AgSynth-synthesized data at the 10K scale delivers gains of +7.42 and +9.15 on WeMath and LogicVista respectively, confirming the critical role of the construction paradigm in stimulating model reasoning capabilities. The framework has been open-sourced.

Limitations

This section discusses the main current limitations of AgSynth and the corresponding directions for improvement.

Bottleneck in the Procedural Rendering Capability of Foundation Models. The synthesis quality of AgSynth is highly dependent on the procedural rendering capability of the generative model. Error attribution analysis reveals that, among the filtered samples, image-text information inconsistency (38.7%) and image rendering defects (33.0%) collectively account for 71.7%, indicating systematic deficiencies in the model’s ability to translate mathematical conceptions into precise rendering code. Introducing models with larger parameter scales can substantially improve the pass rate, but inference costs rise accordingly. Subsequent mitigation can proceed along two paths: employing Dynamic Batching and adaptive quantization to optimize the cost-effectiveness of computational resources, and enhancing procedural rendering capability through targeted fine-tuning or code self-repair mechanisms.

Residual Deficiencies in Image-Text Consistency. The reverse construction paradigm has substantially improved image-text consistency at the mechanistic level, with the proportion of errors along the content correctness dimension being significantly lower than that of the image generation dimension. Nevertheless, image-text information inconsistency still ranks first among all error categories at 38.7%, indicating that the limitations of procedural rendering precision constrain the full realization of the consistency advantages offered by reverse construction. The current approach relies on post-hoc filtering via a dual-model veto mechanism; future work could introduce fine-grained semantic constraints during the generation stage to further reduce the inconsistency rate at its source.

Potential Label Leakage Risk. Approximately 7.3% of the filtered samples exhibited cases where the image content contained answer-related information. After the two-tier quality inspection, such defects have been substantially filtered out (with a manual review pass rate of 96.8%). Further mitigation can be achieved by introducing prompt-engineering-based semantic decoupling or masking-isolation strategies during the generation phase.

The aforementioned limitations stem from the capability boundaries of current large language models in procedural rendering, rather than from inherent flaws in the framework design itself. As foundation model capabilities continue to evolve, the synthesis quality and pass rate of AgSynth are expected to improve correspondingly, with the framework serving as a general-purpose capability amplifier that will continuously benefit from advancements in the underlying models.

Ethical considerations

All mathematical problems, image materials, and textual annotations used in constructing AgSynth-10K are derived from publicly available sources or generated through our proposed synthesis framework. During dataset construction, we conduct multi-round image-text consistency verification and dual-model veto-based filtering to reduce hallucinated, misleading, or low-quality instances. This work is intended solely for advancing research on multimodal mathematical reasoning and does not involve personal, sensitive, or private information. Furthermore, we transparently acknowledge the use of AI assistants only for language polishing, code formatting, and auxiliary manuscript preparation, while all core research design, data construction protocols, experimental analysis, and conclusions remain under the authors’ responsibility.

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 *Proceedings of the 18th Conference of the European Chapter of the Association for Computational Linguistics: Student Research Workshop*, pages 225–237, St. Julian’s, Malta. Association for Computational Linguistics.
- Giacomo Camposampiero, Michael Hersche, Roger Wattenhofer, Abu Sebastian, and Abbas Rahimi. 2025. [I-RAVEN-x: Benchmarking generalization and robustness of analogical and mathematical reasoning in large language and reasoning models](#). In *The 5th Workshop on Mathematical Reasoning and AI at NeurIPS 2025*.
- Jiaqi Chen, Tong Li, Jinghui Qin, Pan Lu, Liang Lin, Chongyu Chen, and Xiaodan Liang. 2022. [UniGeo: Unifying geometry logical reasoning via reformulating mathematical expression](#). In *Proceedings of the 2022 Conference on Empirical Methods in Natural Language Processing*, pages 3313–3323, Abu Dhabi, United Arab Emirates. Association for Computational Linguistics.

- Jiaqi Chen, Jianheng Tang, Jinghui Qin, Xiaodan Liang, Lingbo Liu, Eric Xing, and Liang Lin. 2021. [GeoQA: A geometric question answering benchmark towards multimodal numerical reasoning](#). In *Findings of the Association for Computational Linguistics: ACL-IJCNLP 2021*, pages 513–523, Online. Association for Computational Linguistics.
- Zhe Chen, Weiyun Wang, Yue Cao, Yangzhou Liu, Zhangwei Gao, Erfei Cui, Jinguo Zhu, Shenglong Ye, Hao Tian, Zhaoyang Liu, Lixin Gu, Xuehui Wang, Qingyun Li, Yiming Ren, Zixuan Chen, Jiapeng Luo, Jiahao Wang, Tan Jiang, Bo Wang, and 23 others. 2025. [Expanding performance boundaries of open-source multimodal models with model, data, and test-time scaling](#). *Preprint*, arXiv:2412.05271.
- Gheorghe Comanici, Eric Bieber, Mike Schaekermann, Ice Pasupat, Naveen Sachdeva, Inderjit Dhillon, Marcel Blistein, Ori Ram, Dan Zhang, Evan Rosen, Luke Marris, Sam Petulla, Colin Gaffney, Asaf Aharoni, Nathan Lintz, Tiago Cardal Pais, Henrik Jacobsson, Idan Szpektor, Nan-Jiang Jiang, and 3416 others. 2025. [Gemini 2.5: Pushing the frontier with advanced reasoning, multimodality, long context, and next generation agentic capabilities](#). *Preprint*, arXiv:2507.06261.
- Yihe Deng, Hritik Bansal, Fan Yin, Nanyun Peng, Wei Wang, and Kai-Wei Chang. 2026. [OpenVL-Thinker: Complex vision-language reasoning via iterative SFT-RL cycles](#). In *The Thirty-ninth Annual Conference on Neural Information Processing Systems*.
- Zhiwei He, Tian Liang, Jiahao Xu, Qiuzhi Liu, Xingyu Chen, Yue Wang, Linfeng Song, Dian Yu, Zhenwen Liang, Wenxuan Wang, Zhuosheng Zhang, Rui Wang, Zhaopeng Tu, Haitao Mi, and Dong Yu. 2026. [Deepmath-103k: A large-scale, challenging, decontaminated, and verifiable mathematical dataset for advancing reasoning](#). In *The Fourteenth 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)*.
- Zihan Huang, Tao Wu, Wang Lin, Shengyu Zhang, Jingyuan Chen, and Fei Wu. 2025. [Autogeo: Automating geometric image dataset creation for enhanced geometry understanding](#). *IEEE Transactions on Multimedia*, 27:3105–3116.
- Fuxiao Liu, Kevin Lin, Linjie Li, Jianfeng Wang, Yaser Yacoob, and Lijuan Wang. 2024. [Mitigating hallucination in large multi-modal models via robust instruction tuning](#). In *The Twelfth International Conference on Learning Representations*.
- Wentao Liu, Qianjun Pan, Yi Zhang, Zhuo Liu, Ji Wu, Jie Zhou, Aimin Zhou, Qin Chen, Bo Jiang, and Liang He. 2025. [Cmm-math: A chinese multimodal math dataset to evaluate and enhance the mathematics reasoning of large multimodal models](#). In *Proceedings of the 33rd ACM International Conference on Multimedia, MM '25*, page 12585–12591, New York, NY, USA. Association for Computing Machinery.
- Pan Lu, Hritik Bansal, Tony Xia, Jiacheng Liu, Chunyuan Li, Hannaneh Hajishirzi, Hao Cheng, Kai-Wei Chang, Michel Galley, and Jianfeng Gao. 2023. [Mathvista: Evaluating mathematical reasoning of foundation models in visual contexts](#). In *The 3rd Workshop on Mathematical Reasoning and AI at NeurIPS'23*.
- Zimu Lu, Aojun Zhou, Houxing Ren, Ke Wang, Weikang Shi, Juntong 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, Bangkok, Thailand. Association for Computational Linguistics.
- Haipeng Luo, Huawen Feng, Qingfeng Sun, Can Xu, Kai Zheng, Yufei Wang, TAO YANG, Han Hu, and Yansong Tang. 2026a. [Agentmath: Empowering mathematical reasoning for large language models via tool-augmented agent](#). In *The Fourteenth International Conference on Learning Representations*.
- Ruilin Luo, Zhuofan Zheng, Lei Wang, Yifan Wang, Xinzhe Ni, Zicheng Lin, Songtao Jiang, Yiyao Yu, Chufan Shi, Ruihang Chu, Jin zeng, and Yujiu Yang. 2026b. [Unlocking multimodal mathematical reasoning via process reward model](#). In *The Thirty-ninth Annual Conference on Neural Information Processing Systems*.
- Fanqing Meng, Lingxiao Du, Zongkai Liu, Zhixiang Zhou, Quanfeng Lu, Daocheng Fu, Tiancheng Han, Botian Shi, Ping Luo, Kaipeng Zhang, Yu Qiao, Qiaosheng Zhang, and Wenqi Shao. 2025. [MM-eureka: Exploring the frontiers of multimodal reasoning with rule-based reinforcement learning](#).
- OpenAI. 2025. [Introducing GPT-5.2](#).
- Runqi Qiao, Qiuna Tan, Guanting Dong, MinhuiWu MinhuiWu, Chong Sun, Xiaoshuai Song, Jiapeng Wang, Zhuoma GongQue, Shanglin Lei, YiFan Zhang, Zhe Wei, Miaoxuan Zhang, Runfeng Qiao, Xiao Zong, Yida Xu, Peiqing Yang, Zhimin Bao, Muxi Diao, Chen Li, and Honggang Zhang. 2025. [We-math: Does your large multimodal model achieve human-like mathematical reasoning?](#) In *Proceedings of the 63rd Annual Meeting of the Association for Computational Linguistics (Volume 1: Long Papers)*, pages 20023–20070, Vienna, Austria. Association for Computational Linguistics.
- Runqi Qiao, Qiuna Tan, Peiqing Yang, Yanzi Wang, Xiaowan Wang, Enhui Wan, Guanting Dong, Shiqiang Lang, Sitong Zhou, Yida Xu, Yuchen Zeng, Jie Wang,

- Chong Sun, Chen Li, and Honggang Zhang. 2026. [We-math 2.0: A versatile mathbook system for incentivizing visual mathematical reasoning](#). In *The Fourteenth International Conference on Learning Representations*.
- Jonathan Roberts, Mohammad Reza Taesiri, Ansh Sharma, Akash Gupta, Samuel Roberts, Ioana Croitoru, Simion-Vlad Bogolin, Jialu Tang, Florian Langer, Vyas Raina, Vatsal Raina, Hanyi Xiong, Vishaal Udandara, Jingyi Lu, Shiyang Chen, Sam Purkis, Tianshuo Yan, Wenye Lin, Gyungin Shin, and 15 others. 2025. [Zerobench: An impossible visual benchmark for contemporary large multimodal models](#). *Preprint*, arXiv:2502.09696.
- Zhihong Shao, Peiyi Wang, Qihao Zhu, Runxin Xu, Junxiao Song, Mingchuan Zhang, Y. K. Li, Y. Wu, and Daya Guo. 2024. [Deepseekmath: Pushing the limits of mathematical reasoning in open language models](#). *CoRR*, abs/2402.03300.
- Wenhao Shi, Zhiqiang Hu, Yi Bin, Junhua Liu, Yang Yang, See-Kiong Ng, Lidong Bing, and Roy Ka-Wei Lee. 2024. [Math-LLaVA: Bootstrapping mathematical reasoning for multimodal large language models](#). In *Findings of the Association for Computational Linguistics: EMNLP 2024*, pages 4663–4680, Miami, Florida, USA. Association for Computational Linguistics.
- Haoxiang Sun, Lizhen Xu, Bing Zhao, Wotao Yin, Wei Wang, Boyu Yang, Rui Wang, and Hu Wei. 2026a. [Deepvision-103k: A visually diverse, broad-coverage, and verifiable mathematical dataset for multimodal reasoning](#). *Preprint*, arXiv:2602.16742.
- Jiayin Sun, Caixia Sun, Boyu Yang, Hailin Li, Xiao Chen, Yi Zhang, Errui Ding, Liang Li, Chao Deng, and Junlan Feng. 2026b. [Geotikzbridge: Advancing multimodal code generation for geometric perception and reasoning](#). *CoRR*, abs/2603.22687.
- Qwen Team. 2025. [Qwen3 technical report](#). *Preprint*, arXiv:2505.09388.
- Ke Wang, Juntao Pan, Weikang Shi, Zimu Lu, Houxing Ren, Aojun Zhou, Mingjie Zhan, and Hongsheng Li. 2024. [Measuring multimodal mathematical reasoning with MATH-vision dataset](#). In *The Thirty-eight Conference on Neural Information Processing Systems Datasets and Benchmarks Track*.
- Yijia Xiao, Edward Sun, and Wei Wang. 2024. [Logicvista: Multimodal LLM logical reasoning benchmark in visual contexts](#).
- Yi Yang, Xiaoxuan He, Hongkun Pan, Xiyan Jiang, Yan Deng, Xingtao Yang, Haoyu Lu, Dacheng Yin, Fengyun Rao, Minfeng Zhu, Bo Zhang, and Wei Chen. 2025. [R1-onevision: Advancing generalized multimodal reasoning through cross-modal formalization](#). In *2025 IEEE/CVF International Conference on Computer Vision (ICCV)*, pages 2376–2385.
- 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*.
- Renrui Zhang, Dongzhi Jiang, Yichi Zhang, Haokun Lin, Ziyu Guo, Pengshuo Qiu, Aojun Zhou, Pan Lu, Kai-Wei Chang, Yu Qiao, Peng Gao, and Hongsheng Li. 2025. [Mathverse: Does your multi-modal llm truly see the diagrams in visual math problems?](#) In *Computer Vision – ECCV 2024*, pages 169–186, Cham. Springer Nature Switzerland.
- Wenwen Zhuang, Xin Huang, Xiantao Zhang, and Jin Zeng. 2025. [Math-puma: Progressive upward multimodal alignment to enhance mathematical reasoning](#). In *Thirty-Ninth AAAI Conference on Artificial Intelligence, Thirty-Seventh Conference on Innovative Applications of Artificial Intelligence, Fifteenth Symposium on Educational Advances in Artificial Intelligence, AAAI 2025, Philadelphia, PA, USA, February 25 - March 4, 2025*, pages 26183–26191. AAAI Press.
- Chengke Zou, Xingang Guo, Rui Yang, Junyu Zhang, Bin Hu, and Huan Zhang. 2025. [Dynamath: A dynamic visual benchmark for evaluating mathematical reasoning robustness of vision language models](#). In *The Thirteenth International Conference on Learning Representations*.

A Multi-Agent System Prompts & Audit-Feedback Loop

A.1 Architecture Design Philosophy

The V7 pipeline follows a **4+1 Architecture** with embedded *Generate* $\rightarrow$ *Audit* $\rightarrow$ *Feedback* $\rightarrow$ *Retry* loops. Agent 5 serves as a universal middleware auditor. If any output fails its dedicated audit, Agent 5 generates corrective instructions, triggering a re-generation (max 3 attempts).

A.2 Detailed Agent Specifications

Agent 1: Concept Designer

Mission: Logic Kernel Extraction & Framework Design

- **Role:** Analyzes seed data to extract mathematical kernels and designs new scenarios with concrete values (e.g., 3 apples, 8kg).
- **Core Output:** JSON including `core_logic_kernel`, `concept_design`, and `expected_answer`.
- **Audit Loop (Agent 5):** Senior Mathematical Logic Expert. Validates logic closure and consistency via Python math tools.
- **Rejection Triggers:** Contradictory data or non-unique solutions.

Agent 2: Visual Specifier

Mission: Natural Language Visual Blueprinting

- **Role:** Receives logic and designs a complete visual specification (shapes, hex colors, exact coordinates) without using code.
- **Core Output:** Descriptive natural language (max 1500 chars).
- **Audit Loop (Agent 5):** Spatial Geometry Expert. Checks for data drift and ensures the visual form matches the mathematical logic.
- **Rejection Triggers:** Missing spatial details or data mismatches with Agent 1.

Agent 3: Painter

Mission: Executable Visualization Rendering

- **Role:** Translates blueprints into Matplotlib code.
- **Core Output:** Executable Python code.
- **Audit Loop (Agent 5):** Image QC Officer. Audits the **rendered image** for readability and technical compliance.
- **Critical Rule: Zero Tolerance** for overlapping elements or occlusion.
- **Feedback:** Specific line-by-line code corrections.

Agent 4: QA Generator

Mission: Multimodal Question Synthesis

- **Role:** Generates the final question text (min 50 chars) and step-by-step solutions based on the final image.
- **Core Output:** JSON with `question_text` and `solution_steps`.
- **Audit Loop (Agent 5):** Chief Editor. Final gate for text-image consistency.
- **Rejection Triggers:** Numerical mismatches between text and image.

Agent 5: Quality Rater / Auditor

Mission: Full-Process Quality Gate

- **Auditor Function:** Real-time feedback loop after each stage.
- **Scoring Function:** Hybrid (40% rule + 60% LLM) rating (1–10).
- **Strict Penalty:** Any overlap or logic error results in Score < 5.

A.3 Pipeline Control & Quality Gates

Stage 4.5: Text Masking Ensures figure dependency by stripping numerical values from the question.

- **Rewrite:** *gemini-3.1-pro-preview*.
- **Validate:** *claude-opus-4-6-20260205* (Max 5 retries).

Stage 6: Dual-Model Final Gate A unanimous pass from *gemini-3.1-pro* and *claude-sonnet* is required for final sample acceptance.

A.4 Execution Workflow Summary

Table 3: Closed-Loop Pipeline Execution Flow

Step	Process: Generate → Audit	Retries
1	Concept → Logic Closure	3
2	Visual Spec → Spatial Mapping	3
3	Code → Readability/Occlusion	3
4	QA Text → Text-Image Match	3
4.5	Masking → Figure Dependency	5
6	Dual-Model Quality Gate	Pass/Fail

B Training Hyperparameters and Configurations

This appendix provides the detailed configuration and hyperparameter settings used for the RL fine-tuning process. The experiment is conducted using the `veRL` (EasyR1) framework with the Group Relative Policy Optimization (GRPO) algorithm.

B.1 Model and Infrastructure

The training is performed on a single node with 8×NVIDIA GPUs, utilizing FSDP (Fully Sharded Data Parallel) for distributed strategy and vLLM as the rollout engine.

Table 4: Model and Framework Configurations

Parameter	Value
Base Model	Qwen3-VL-8B-Instruct
Framework	veRL (EasyR1)
Algorithm	GRPO
Vision Tower Status	Unfrozen (Trainable)
Gradient Checkpointing	Enabled
Mixed Precision	bf16 (Param), fp32 (Buffer/Reduce)
Distributed Strategy	FSDP
Rollout Engine	vLLM

B.2 Data and Pre-processing

The dataset comprises 10,477 samples specifically curated for multi-source mathematical reasoning.

Table 5: Data and Tokenization Settings

Parameter	Value
Training Samples Dataset	10,477 easyrl_math_reasoning_m ultisource_qita_10477
Max Prompt Length	4,096 tokens
Max Response Length	4,096 tokens
Image Resolution (Pixels)	Min: 262,144 Max: 4,194,304
Filter Overlong Prompts	True
Dynamic Batching	Enabled

B.3 Training and Optimizer Settings

The training process spans 2 epochs with a constant learning rate schedule.

Table 6: Optimization Hyperparameters

Parameter	Value
Total Epochs	2
Training Steps	84
Global Batch Size	64
Micro Batch Size (Update)	2 per device
Micro Batch Size (Rollout)	4 per device
Optimizer	AdamW
Actor Learning Rate	3e-6
Learning Rate Scheduler	Constant
Warmup Ratio	0
Weight Decay	0.01
Max Grad Norm	1.0

B.4 RL Algorithm (GRPO) and Rollout Details

Hyperparameters specific to the GRPO algorithm and the rollout generation process are detailed below.

Table 7: GRPO and Rollout Configurations

Parameter	Value
Advantage Estimator	GRPO
Group Size (G)	4 samples per prompt
KL Coefficient (β)	0.01
KL Penalty Type	low_var_kl
Clip Ratio (ϵ)	[0.2, 0.3]
Dual Clip Ratio	3.0
Reward Function	math_accuracy_only (Binary Score)
Rollout Temperature	1.0
Rollout Top-p	1.0
Tensor Parallel Size (vLLM)	2
GPU Memory Utilization	0.6

C Visual Diversity and Evolution from Seeds

This appendix demonstrates the visual diversity achieved by our multi-agent pipeline. Figure 7 illustrates the evolutionary process where the system

extracts the core mathematical logic from simple seed questions and manifests them into various complex visual forms.

C.1 Observations on Visual Mapping

As shown in Figure 7, the diversity is manifested in two primary directions:

- **Contextual Variation:** The left column illustrates a transition from a simple line-and-point seed to path-finding problems involving stylized icons (houses, schools) and varied geometric containers (hexagons).
- **Geometric Complexity:** The right column shows how a simple triangular or angle-based seed is expanded into multi-ray systems, circular constraints, and abstract node-link diagrams, significantly increasing the visual complexity while maintaining the underlying mathematical logic.

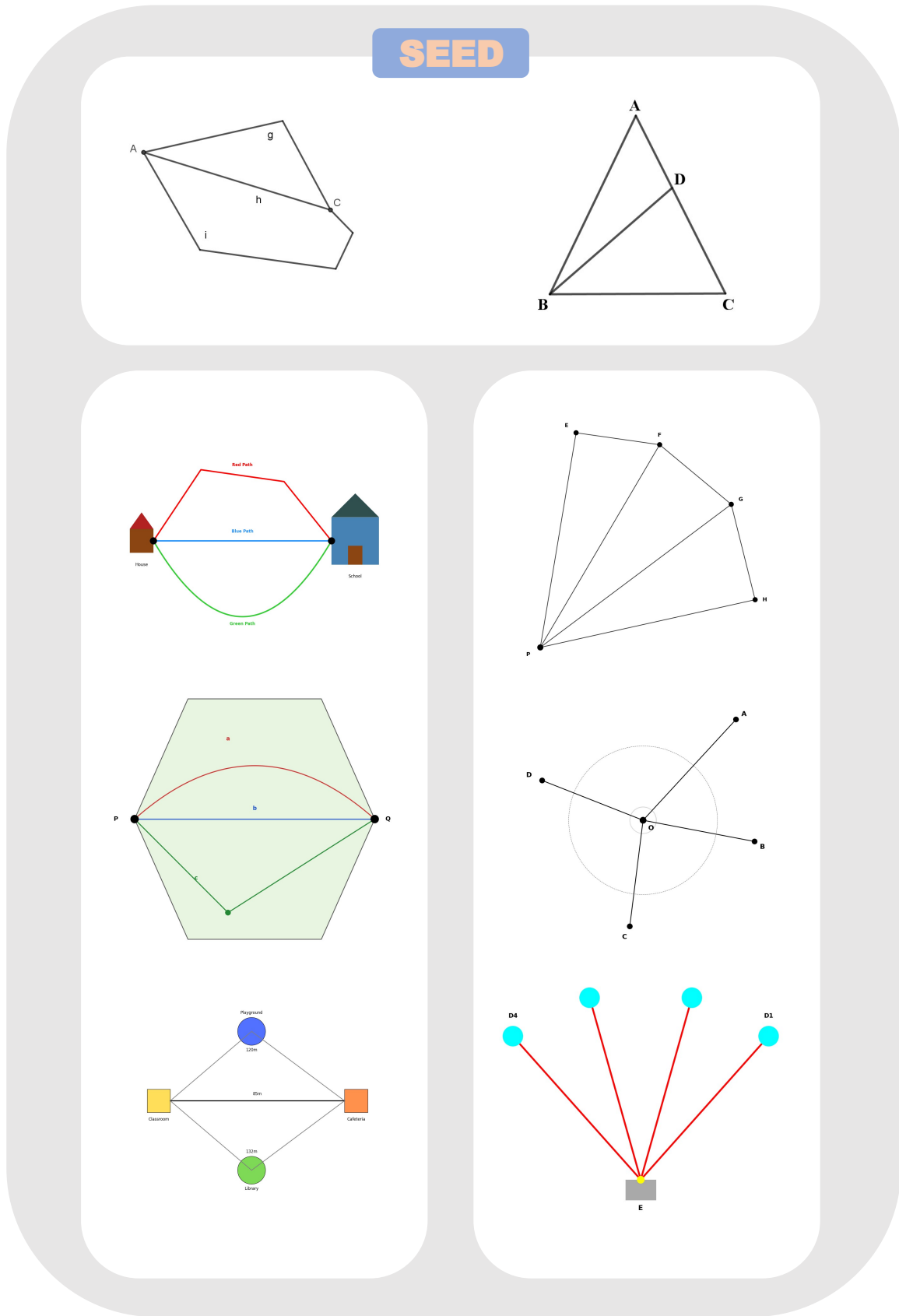


Figure 7: Comparison between seed questions and generated visual variations. The top panel (labeled “SEED”) shows the original geometric sketches. The lower panels showcase the diverse visual outputs generated by our pipeline. This evolution demonstrates Agent 1’s ability to reinterpret logical kernels (e.g., path selection, geometric relations) and Agent 2’s ability to map them into diverse contexts, ranging from stylized real-world scenarios to complex abstract geometric configurations.