


• Supplementary File •


COMET : “Cone of experience” enhanced
**large multimodal model for mathematical problem
generation**

Sannyuya Liu^{1,2}, Jintian Feng^{1,2}, Zongkai Yang^{1,2}, Yawei Luo³,
Qian Wan^{1,2*}, Xiaoxuan Shen^{1,2*} & Jianwen Sun^{1,2*}

¹National Engineering Research Center of Educational Big Data, Central China Normal University, Wuhan 430079, China;

²Faculty of Artificial Intelligence in Education, Central China Normal University, Wuhan 430079, China;

³School of Software Technology, Zhejiang University, Hangzhou 310027, China

Appendix A Methodology

Figure 1 in the main manuscript contains a schematic diagram of the three-stage fine-tuning framework. The entire fine-tuning process is guided by the “Cone of Experience” educational theory, injecting symbolic experience, iconic experience, and direct experience. This section first defines the global fine-tuning goals and notations, decomposing the application requirements of the target domain into three subtasks for reinforcement. Then, the three-stage fine-tuning framework is expanded according to the target of injected experience, elaborating on the definitions, construction methods, and training methods.

Appendix A.1 Problem formulation

To effectively apply LMM in teaching scenarios, this work mainly enhances three capabilities of LMM during the domain fine-tuning process: **controllable generation** (CG), **analogy generation** (AG), and **fine-grained solving** (FS) for mathematical problems. Both CG and AG reflect the ability of LMM to generate problems, the difference being that the former generates the mind of design and original problem according to given requirements (such as problem type, knowledge point, difficulty level, etc.), while the latter understands and transforms the seed problems (such as changing topic and type, expanding knowledge point or adjusting difficulty level). The FS reflects the problem-solving capacity of LMM, emphasizing the importance of producing detailed solution steps similar to textbook references.

For LMM, the instructions for the above three tasks can be formally defined as follows:

1. Given the problem type t , knowledge point c , difficulty-level d and grade level g , the CG prompt is constructed as $q_c = F_c(t, c, d, g)$.
2. Given the seed problem $s \in S$, the AG prompt is constructed as $q_a = F_a(s)$.
3. Suppose a math problem is p , the prompt identifier of FS is $q_s = F_s(p)$.

The F_c, F_a, F_s can be flexibly designed according to the scene, and the settings in this work can be seen in Section 3.4. Please note that in this paper, \mathbf{x} represents a vector or a string, x represents a scalar or a single character, X represents a set, and \mathbf{X} represents a function.

The task requirements are defined as $q_{in} \in \{q_c, q_a, q_s\}$. This work can be defined as performing three-stage fine-tuning based on the general LMM $F_{lmm}^{(0)}$, combined with the “Cone of Experience” theory, to obtain an LMM $F_{lmm}^{(3)}$ that meets the application requirements of mathematical problem generation in the teaching scene, so as to maximize the following conditional probability:

$$P_{lmm}(\mathbf{m}|q_{in}; \theta^{(3)}) = \prod_{k=1}^{N_m} P_{lmm}(w_k^m | q_{in} \oplus \mathbf{w}_{<k}^m; \theta^{(3)}). \quad (A1)$$

$$P_{lmm}(\mathbf{a}|q_{in}, \mathbf{m}; \theta^{(3)}) = \prod_{k=1}^{N_a} P_{lmm}(w_k^a | q_{in} \oplus \mathbf{m} \oplus \mathbf{w}_{<k}^a; \theta^{(3)}). \quad (A2)$$

where $\theta^{(3)}$ is the parameters of LMM $F_{lmm}^{(3)}$, \oplus represents the string concatenation operation. $\mathbf{m} = \{w_1^m, w_2^m, \dots, w_{N_m}^m\}$ represents the mind of design or problem-solving steps generated by LMM, and $\mathbf{a} = \{w_1^a, w_2^a, \dots, w_{N_a}^a\}$ represents the original problem or final answer generated by the LMM.

* Corresponding author (email: wanq8228@ccnu.edu.cn, shenxiaoxuan@ccnu.edu.cn, sunjw@ccnu.edu.cn)

Appendix A.2 Symbolic experience: learning through abstractions

This paper defines symbolic experience as the background knowledge related to the target domain, or the prerequisite knowledge required to carry out the target task. Symbolic experience does not directly help the model solve specific tasks, but it provides strong support. For mathematical problem generation, we summarize symbolic experience into four types for production: book knowledge, graph knowledge, arithmetic knowledge, and general knowledge.

The data sources of **book knowledge** include textbooks, lecture notes, teachers' books, pedagogy, and psychology books, aiming to build teaching concepts and supplement subject knowledge. Through methods such as web crawling, OCR, and manual annotation, we complete data collection and pre-processing (de-duplication, noise reduction, etc.) via both online and offline channels. The number of book knowledge tokens sorted out in this work is approximately 140M.

We construct a large heterogeneous subject knowledge graph, where the node types include grade, knowledge points, concept descriptions, and example problems. This graph encompasses 1,225 knowledge points and related concepts from elementary to junior high school, providing approximately 18,000 example problems. To train LMM using structured data, we design a graph sampling method based on random walks to extract diversified and differentiated disciplinary information. Then GPT-4V is used to transform the edge information into a concatenated text, thereby generating **graph knowledge** for symbolic experiences. Specifically, the heterogeneous subject knowledge graph is represented as $G = \langle \{N_c, N_g, N_d, N_p\}, E \rangle$, where N_c, N_g, N_d, N_p represent the node sets of knowledge points, grade, concept descriptions, and related example problems. E is the set of edges between all nodes. The generation process of graph knowledge can be seen in Algorithm A1, which generates two types of training samples: a whole link learning sample (Sample_1) is formed as a four-tuple $\{\text{grade}, \text{knowledge point}, \text{concept description}, \text{example problem}\}$, and a relationship learning sample (Sample_2) formed by the concatenation of multiple adjacent knowledge points, function $\mathbf{GPT4V}(\cdot)$ were used to generate human-readable text, the graph knowledge result in totaling 220M tokens.

Algorithm A1 Graph Knowledge Generation

```

Input:  $G = \langle \{N_c, N_g, N_d, N_p\}, E \rangle$ ,
           $D_1 = \emptyset, D_2 = \emptyset$ 
Output: Sample_1, Sample_2
1:  $n_i = \mathbf{random}(N_c)$ 
2:  $D_1 = D_2 = \{n_i\}$ 
3:  $n_j = \mathbf{random}(N_g)$ 
4: if  $e_{ij}$  is not None then
5:    $D_1 = D_1 \cup \{n_j\}$ 
6: end if
7:  $n_j = \mathbf{random}(N_d)$ 
8: if  $e_{ij}$  is not None then
9:    $D_1 = D_1 \cup \{n_j\}$ 
10: end if
11: for  $k = 1$  to 3 do
12:    $n_j = \mathbf{random}(N_d)$ 
13:   if  $e_{ij}$  is not None then
14:      $D_2 = D_2 \cup \{n_j\}$ 
15:   end if
16: end for
17: for  $k = 1$  to 5 do
18:    $n_j = \mathbf{random}(N_c - \{n_i\})$ 
19:   if  $e_{ij}$  is not None then
20:      $D_2 = D_2 \cup \{n_j\}$ 
21:   end if
22: end for
23: Sample_1 =  $\mathbf{GPT4V}(D_1)$ 
24: Sample_2 =  $\mathbf{GPT4V}(D_2)$ 

```

The function of **arithmetic knowledge** is to compensate for the shortcomings of LMM in arithmetic, to reduce the probability of numerical errors occurring in the mathematical reasoning process. It is an equation consisting of pure numbers and mathematical operators. We directly use the arithmetic dataset proposed by Yang et al. [1]. This dataset is carefully designed, containing not only operations such as addition, subtraction, multiplication, division, and exponentiation, but also various numerical formats such as integers, decimals, percentages, fractions, and negative numbers. In this work, approximately 200M tokens are extracted as fine-tuning data for arithmetic knowledge.

We extracted approximately 220M tokens of generic data (including plain text, single-turn, and multi-turn Q&A) from open-source corpora, such as Wikipedia, SkyPile-150B, MOSS, and BELLE [2-5], as the **general knowledge** in symbolic experience. The main role of this knowledge is to slow down the forgetting phenomenon caused by continued pre-training.

This stage processes all the data associated with symbolic experience as pre-training form and infuses it into the LMM for learning, i.e., no masking of data content is undertaken. The backpropagation of model training computes loss from the first token of the input. Assuming the input sample is \mathbf{x} , the loss function at this stage is as follows:

$$\mathbf{Loss}_{1st}(\theta^{(0)}) = - \sum_{x_i \in \mathbf{x}} \log P(x_i | \mathbf{x}_{<i}; \theta^{(0)}). \quad (\text{A3})$$

Appendix A.3 Iconic experience: learning through observation

The iconic experience is defined as the data generated by the subject in the process of performing the target task, which includes not only human experts proficient in the target task but also the large model. Injecting the iconic experience aims to allow LMM to learn mathematical problem generation from humans and improve upon the failed reasoning data produced by other LMMs. This paper summarizes the iconic experience into three types of production: the experience of stem generation, problem-solving, and failure.

To construct **stem generation experience**, we first collect exercises and test items covering all grades from elementary to junior high school. Next, based on manual annotation methods, we extracted key information from math problems in several dimensions including educational grade, problem type, knowledge points, and difficulty, and deduced problem requirements in reverse. Finally, we constructed a query-problem pair, with manual writing examples of mind of design, and used GPT-4V for bulk supplementation of question-making ideas in a few-shot manner. The final data form is *problem requirement, mind of design, original problem*.

To construct **problem-solving experience**, we hire normal school students to write analyses and answers for the collected mathematical problems. However, due to differences in cognitive levels and writing styles between individuals, it is difficult to align the granularity of the analyses. To generate fine-grained analyses, we use GPT-4V to generate high-quality analyses with consistent writing styles based on manually parsed data. Three generation methods are proposed:

1. The task requires GPT-4V to directly solve the problem: $\{q\} \mapsto \{s\}$.
2. The task requires GPT-4V to fill in the middle process when both the problem and answer are given: $\{q, a\} \mapsto \{s\}$.
3. When the complete problem, analyses, and answer are given, GPT-4V is required to rewrite the analyses: $\{q, s, a\} \mapsto \{s'\}$.

We chose the second method as the data production method for this stage due to its stability. The final data form is $\{\textit{mathematical problem, mind of solution, final answer}\}$.

Failure experience is mainly generated by LMMs that have not been domain-adapted. First, a collaborative environment consisting only of LMMs is built. In this environment, we design several task instructions for the three capabilities required for mathematical problem generation, and prompt the LMMs to respond through multiple rounds of dialogue. Open-source LMMs (such as Qwen-VL-Chat, Yi-VL-6/34B, etc.) play the role of student responding to instructions, while GPT-4V acts as a teacher who has known the standard solution and final answer in advance, generating suggestions based on the student's responses.

Secondly, given an instruction, a student model that has been deployed is randomly selected to perform inference. The teacher model evaluates the completion based on the response and decides whether to terminate the current conversation. If the judgment is no, the teacher model generates instructive suggestions and initiates the next round of dialogue. It is worth noting that while each new round of dialogue retains historical information, the student model is re-randomized.

Finally, due to the limited capabilities of student models, the collaborative environment will generate a batch of multi-round dialogue data stacked by the multiple failure experiences. The merged process data format is $\{\textit{instruction, student}_1, \textit{teacher}_1, \textit{student}_2, \textit{teacher}_2, \dots, \textit{student}_n, \textit{teacher}_n\}$, in which the student response content gradually optimized according to the suggestions of teacher is the main learning object of the failure experience, that is, the instruction and suggestion of teacher are masked in the instruction tuning stage.

In this stage, the data pertaining to the iconic experience is learned by the LMM in the form of instruction Tuning. All data is arranged in a query-response pair, and a masking process is applied to the query part. The backward propagation of model training only starts calculating loss from the first token of the response. Suppose the query-response pairs are defined as $q : a$, the model input sequence is $x = \{q \oplus a\}$. The loss function at this stage is as follows:

$$\textit{Loss}_{2st}(\theta^{(1)}) = - \sum_{x_i \in a} \log P(x_i | x_{<i}; \theta^{(1)}). \quad (\text{A4})$$

Appendix A.4 Direct experience: learning by doing

The direct experience is defined as the procedural data generated when the fine-tuned object carries out the target task with results feedback. Such experience aims to correct the inference preference of the LMM with higher-order domain values, allowing it to embodied evolve during the practice.

Firstly, we design a set of task instructions for three subtasks (CG, AG, and FS). For CG, the focus of the prompt design is to highlight the controllable elements in the generation process. This paper mainly considers four controllable factors in the problem generation process: grade, problem type, knowledge point, and difficulty level, and requires giving out the mind of design. For AG, the prompt design focuses on asking the model to first understand the seed problem to initially judge the important elements such as problem type and knowledge points, and then give the mind of design and rewrite the problem in the form of chain of thought. For FS, the core concept of the prompt design is to clearly require the model to generate a detailed analysis process rather than just outputting an answer. All prompts designed in this work for the three tasks are shown in Figure A1.

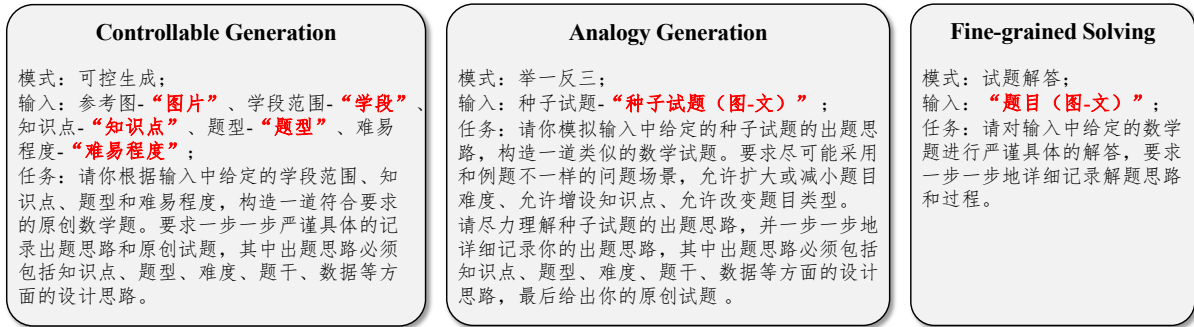


Figure A1 Prompt of the three tasks.

Secondly, the LMM can produce multiple different responses to the same query due to the randomness of its reasoning. We order the preferences of multiple responses corresponding to the same instruction. This paper utilizes human preferences (manual annotation) and model preferences (GPT-4V generation) during the preference ranking process. The final data form is $\{\textit{task instruction, high preference response, low preference response}\}$.

The fine-tuning stage uses the direct preference optimization [6] (DPO) algorithm to infuse direct experience into LMM, the loss function is as follows:

$$\textit{Loss}_{3st}(\theta^{(2)}, \theta^{(3)}) = -\mathbb{E}_{x, y_w, y_l} \sim D[\log \sigma(\beta \log \frac{P(y_w | x; \theta^{(3)})}{P(y_w | x; \theta^{(2)})} - \beta \log \frac{P(y_l | x; \theta^{(3)})}{P(y_l | x; \theta^{(2)})})]. \quad (\text{A5})$$

the $\theta^{(3)}$ uses $\theta^{(2)}$ as the initial solution, for the same input x , y_w and y_l represent the preferred solution and the non-preferred solution.

Appendix B Experiments

Appendix B.1 Implementation

We conduct the ‘‘Cone of Experience’’ enhanced three-stage fine-tuning based on the well-trained Qwen-VL-Chat [7] provided by Alibaba Cloud. For all fine-tuning stages, Adam with different learning rates is used as the optimizer, and the gradient truncation threshold is set to 0.5. We incorporate a warmup ratio of 0.05 and employ the batch size of 64. To control overfitting, we apply a weight decay of 0.1. The Max token is uniformly set to 2,048, and the data is spliced or truncated in the pre-processing stage to improve training efficiency or reduce information loss. In addition, we employ deepspeed with ZeRO-2 stage [8] to improve parallel efficiency for speed up training.

In the first and second stages, we use LoRA [9] to perform parameter-efficient fine-tuning, then set rank, alpha, and dropout to 16, 32, and 0.05. All linear layers (including the image encoder) of LMM except the head layer are designated to apply the LoRA adapter. Among them, the learning rate of the first stage is set to 2×10^{-5} , and halved in the second stage. We use the DPO [6] algorithm to inject direct experience for learning reasoning preferences in the third stage. The learning rate is 5×10^{-5} , the DPO smoothing value is 0.1. To ensure reproducibility, the random seed is set to 42 during the whole experiment. The three-stage fine-tuning is performed on 8 NVIDIA A800-80G. For one epoch, the three-stage fine-tuning takes about 200, 50, and 20 GPU hours. For evaluation, the inference parameters of LMM are uniformly set top_k to 20, top_p to 0.7, repetition_penalty to 1, and temperature to 0.3.

Appendix B.2 Dataset and baseline

Dataset: The datasets used are shown in Table B1. GSM8K [10] is an English single-modal math word problem (MWP) dataset, containing 7,473 training samples and 1,319 test samples. TAL-SCQ5K-CN¹⁾ is a Chinese single-modal multiple-choice problem (MCP) dataset for K12 math, including 3,000 training samples and 2,000 test samples.

In this work, we build a Chinese multi-modal math problem dataset CMM12K. The construction of CMM12K can be divided into two stages: data collection and annotation processing. In the data collection phase, we hired 3 teachers and 15 students to digitize the synchronized homework problems, examinations, and teacher autonomy questions in K12 teaching materials. Through the collaboration with HanWang Corporation, we used OCR technology to extract information such as grade level, question type, question stem, answers, etc. from paper materials, and stored this data in Word format. For the data annotation phase, under the premise of taking into account different school periods, we extracted 12,000 samples from the above data, and employed 20 normal university students to complete the annotation of knowledge points, solutions, and estimated solving time over a period of 3 months. Finally, we used the Mathpix API²⁾ to transfer the samples of CMM12K in LaTeX format, and further stored them in JSON files for easy LMMs training and optimization calls.

Table B1 Statistics of datasets.

Dataset	Language	Modal	Type	#Train.	#Dev.	#Test.
GSM8K	En → Zh	Single	MWP	7,473	-	1,319
TAL-SCQ5K-CN	Zh	Single	MCP	3,000	-	2,000
CMM12K	Zh	Multi	MCP, MFP, MWP, MPP	10,000	1,000	1,000

CMM12K includes 6,000 single-modal math problems and 6,000 multi-modal math problems, covering most of the knowledge points in K12 math from primary school to junior high school. This dataset contains four types of problems: MCP, math fill-in-the-blank problem (MFP), MWP, and math proof problem (MPP). The training set of CCM12K is divided into 10,000 samples, and the development set and test set are 1,000 samples each.

Figure B1 shows some examples in CMM12K. For each problem in CMM12K, We annotated fields such as the problem statement, problem image, correct answer, problem explanation, problem type, grade level, knowledge point, and estimated solving time.

题目: 如图, 三角形ABC为等腰三角形, $\angle BCA$ 为直角, D是AB的中点, $AB=20\text{dm}$, 圆弧GD、HD的圆心分别在A、B两点, 图中阴影部分的面积是多少?

题目图片:

正确答案: 107 平方分米

解析: 首先可以把右边的1/4圆旋转到左侧下方, 与另外一部分拼成一个大半圆。.....

题型: 解答题

年级: 六年级

知识点: 圆的面积、圆弧、圆心角

预估解题时间 (秒): 300

题目: 下面有一个11位数, 每3个相邻数字之和都是17, 你知道‘?’表示的数字是几吗?

A.6 B.7 C.8 D.9

题目图片:

8				?						6
---	--	--	--	---	--	--	--	--	--	---

正确答案: A

题型: 选择题

年级: 七年级

知识点: 简单周期现象中的规律

预估解题时间 (秒): 400

解析: 我们可以逐步推理, 首先第1位+第2位+第3位=17, 而第1位=8, 所以可以得到第2位+第3位=9, 而第2位+第3位+第4位=17, 因此可以推导出第4位=8。.....

Figure B1 Example of two problems in the CMM12K dataset.

Baselines: We selected a total of nine baselines, containing 3 closed-source models, including GPT-4o [11], Gemini-1.5-Pro [12], and Claude-3-Opus³⁾. Additionally, six open-source LMMs which cover different parameter levels was selected, including Qwen-VL-Chat(7B) [7], Yi-VL-6B/34B [13], LLaVA-1.6(7B) [14], CogVLM(17B) [15], and Math-LLaVA(13B) [16]. Among the open-source models, Math-LLaVA is a mathematics-specialized LMM, which was fine-tuned based on LLaVA-1.5-13B with 360K math problems.

It should be noted that in the three-stage fine-tuning process, all datasets are aligned with Chinese. The training sets of each dataset participate in the construction of iconic experience and the direct experience mainly depends on the development set. The test set is completely isolated from all stages.

1) <https://github.com/math-eval/TAL-SCQ5K>

2) <https://mathpix.com>

3) <https://www.anthropic.com/news/claude-3-family>

Appendix B.3 Metrics

In this work, we design three evaluation criteria for different LMM capabilities.

Scoring mode. Multiple scoring dimensions are designed for **controllable generation** (CG), **analogy generation** (AG), and **fine-grained solving** (FS). The scoring dimensions of CG include language fluency (LF) (both mathematical terms and formulas), logical correctness (LC), content completeness (CC) (both ideas and stems), knowledge point relevance (KR), difficulty appropriateness (DA) and type adaptability (TA). The scoring dimensions of AG include language fluency (LF), logical correctness (LC), content completeness (CC), reasoning rationality (RR), and seed relevance (SR). The scoring dimensions of FS include language fluency (LF), logical correctness (LC), analytical completeness (AC), and answer accuracy (AA). Human annotators and GPT-4V are asked to give a score in the range of 1 to 10 according to the dimension.

Arena mode. Considering the subjectivity of mathematical problem generation, We introduce a referee to comprehensively rule out different responses to the same query from aspects such as accuracy, fluency, and values. Specifically, we calculate a rating value for each LMM to represent the ability on a certain task. It is worth noting that in our experiment, we introduced both GPT-4V and human judges as referees. During the judging process by the referee, the ELO rating algorithm [17, 18] is used to update the rating value of the participating LMM. Assume the initial ELO rating value is 1,000. In M rounds of competition, two LMMs (called LMM- x and LMM- y) are randomly selected to reply to the same query each time. According to the ruling results of the referee, the rating value is calculated as follows:

$$E_x = \frac{1}{1 + 10^{(R_x - R_y)/400}}, \quad (\text{B1})$$

$$R'_i = R_i + K \cdot (\mathbf{PK}(i) - E_i). \quad (\text{B2})$$

where R_x and R_y respectively represent the rating values of LMM- x and LMM- y in the previous round, E_x and $E_y = 1 - E_x$ represent the current expected rating value. R'_i ($i \in \{x, y\}$) represents the updated rating value of LMM- i , $\mathbf{PK}(i) \in \{0, 1\}$ is a boolean function that identifies whether LMM- i win in this round. K represents the K-factor, which defaults to 4 and controls the change rate of rating.

Objective evaluation. For problems with clear answers (this work refers to MCP, MFP, and MWP), the accuracy (ACC) is returned through matching to report the solving performance of LMM. For the MPP, BLEU-1/2/L [19] and ROUGE-1/2/3/4 [20] are used to approximate testing.

Appendix C Result and discussion

Appendix C.1 Performance of controllable generation

Appendix C.1.1 Scoring mode

We evaluated the responses of LMMs to the CG task on the test set and subsequently reported the average scores across six dimensions. The closed-source models are indeed powerful, and GPT-4o achieves optimal or relatively high scores in almost all dimensions. However, as illustrated in Figure C1, COMET outperforms all other baselines in three of six dimensions (LF, KR, and DA), even compared to the powerful closed-source model GPT-4o, COMET outperform it in these dimensions.

Figure C1 shows that our model performs comparably to Yi-VL-34B across the six dimensions of CG tasks. Models with comparable parameter levels, such as Qwen-VL-Chat (7B), LLaVA-1.6 (7B), and Yi-VL-6B, demonstrate slightly inferior performance. For the math-specific model Math-LLaVA, our average score across all dimensions is higher than theirs.

For the largest open-source model in the baseline, Yi-VL-34B, although COMET slightly lags behind Yi-VL-34B in terms of content completeness (CC) and problem type adaptability (TA), it's important to note that our parameter count is approximately five times smaller than that of Yi-VL-34B. Therefore, this discrepancy may arise from limitations imposed by the scale of parameters, which hinder the comprehension and processing of contextual information.

These phenomena explain that through a three-stage fine-tuning process guided by the "Cone of Experience" theory, LMM can effectively focus on generating responses that prioritize controllable factors such as language fluency, logical correctness, knowledge point relevance, and difficulty appropriateness.

Appendix C.1.2 Arena mode

We conducted 6,000 competitions, where two LMMs were randomly selected for an anonymous battle, and the winner was determined by the referee. As shown in Figure C3(a), upon completing all competitions, our model's ELO rating surpassed the open-source baselines with a median value of 1,091 by GPT-4V judge and 1,105 by humans.

Figure C2 illustrates the win rates of battles between various LMMs, revealing that our model exhibits absolute superiority when battling against Qwen-VL-Chat, Yi-VL-6B, LLaVA-1.6, and Math-LLaVA. Moreover, even when pitted against Yi-VL-34B, our model maintains a slight edge with a 2%-3% advantage, and when compared with the powerful closed-source models GPT-4o and Gemini-1.5-Pro, the winning rate is approximately 50%, which shows that COMET has comparable performance to them.

This outcome suggests that while larger parameter sizes may enhance text comprehension and generation capabilities, our model, after undergoing multi-modal fine-tuning guided by the "Cone of Experience" theory, exceeds the performance of a 34B-parameter LMM and even GPT-4o in CG task without altering its parameter scale.

Appendix C.2 Performance of analogy generation

Appendix C.2.1 Scoring mode

In the AG task, COMET demonstrates strong capabilities and a clear advantage. As shown in Figure C1, first, COMET outperforms all open-source baselines across various metrics in AG, including the Yi-VL-34B with five times our parameter count, and the mathematics-specialized model Math-LLaVA. For the three powerful closed-source models, COMET lags behind GPT-4o only in logical correctness (LC).

Secondly, the backbone model Qwen-VL-Chat ranks lower among baselines, trailing behind models like Yi-VL-6B and LLaVA-1.6, which have the same parameter scale. However, after three-stage fine-tuning guided by the "Cone of Experience" theory, its AG capabilities saw a significant improvement, with an average score increase of approximately 60% across all dimensions.

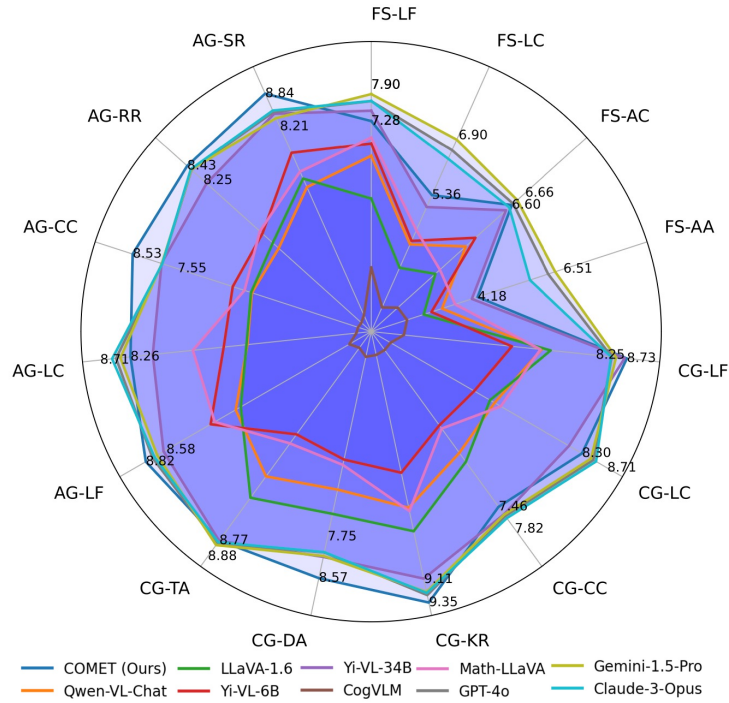


Figure C1 Performance (denotes by average score) of COMET and baselines on a broad range of problem-solving and generation fine-grained indicators of three tasks (CG, AG, and FS). Here, each evaluation indicator is expressed as ‘task-dimension’, and the number in the radar chart refers to the average score of models, we only present the scores of COMET and GPT-4o for visualization purposes. For instance, the label ‘CG-KR’ depicted at the bottom of the radar chart denotes the knowledge point relevance (KR) measure for the controllable generation (CG) task, where our model attains scores of 9.35, while GPT-4o obtains 9.11.

Appendix C.2.2 Arena mode

As illustrated in Figure C2(b), after undergoing 6,000 random competition rounds, COMET’s ELO Rating in AG tasks significantly surpasses all baselines. From the win rate heatmap, it’s evident that COMET exhibits absolute dominance when facing models with equivalent or double the parameter scale, defeating Qwen-VL-Chat, Yi-VL-6B, CogVLM, and LLaVA-1.6 with win rates of 97.5%, 93.5%, 98.5%, 92.0% respectively. Even when confronted with baselines five times larger in parameter size, COMET maintains a win rate of 73%. Compared to closed-source models such as GPT-4o, COMET all have a win rate of over 50%.

AG task hold educational value in demand for problem generation. Although our backbone model may lack training in this specific skill, our model successfully acquires it through the three-stage fine-tuning process. This outcome further validates the superiority of the proposed framework and model in this study.

Appendix C.3 Performance of fine-grained sovling

Appendix C.3.1 Scoring mode

We report the scores for 4 dimensions in the FS task of all LMMs on the test set. The average scores across various evaluation dimensions are presented in Figure C1, where in open-source models, our model achieved the state-of-the-art (SOTA) in 3 of 4 evaluation dimensions (LC, AC, and AA), falling slightly short only in language fluency (LF) compared to Yi-VL-34B. COMET maintains an absolute lead in most dimensions with a relatively smaller parameter size (7B). Moreover, by comparing with the backbone model Qwen-VL-Chat, we can conclude that the three-stage fine-tuning guided by the ‘‘Cone of Experience’’ theory significantly enhances its FS capabilities.

Appendix C.3.2 Arena mode

As illustrated in Figure C3(c), after 6,000 competition rounds, our model triumphs over all baselines with the highest ELO Rating. The average ELO rating leads the backbone model Qwen-VL-Chat by approximately 20%, and also maintains an advantage against Yi-VL-34B. However, analyzing from the perspective of win rates reflected in the heatmap, our model lags behind Yi-VL-34B with a slight disadvantage, as it has a win rate of only 49.5%.

Based on the scoring results of GPT-4V and the arena outcomes, it can be concluded that, in the FS task, our model demonstrates significant performance improvement compared to the backbone model Qwen-VL-Chat, surpassing baselines of the same parameter scale comprehensively, and even shows certain advantages compared to baselines approximately five times its size.

Appendix C.4 Result of objective evaluation

Appendix C.4.1 Performance of objective problems

As shown in Table C1, our model dramatically outperforms the backbone model Qwen-VL-Chat in terms of accuracy on GSM8K and TAL-SCQ5K-CN (+10.85%, +5.85%). The math-specific model Math-LLaVA achieved state-of-the-art (SOTA) results on TAL-SCQ5K-CN, but it lags behind us on other datasets. Although the larger model, including Yi-VL-34B, leads on GSM8K, its parameter size, which is 5 times larger than ours, implies greater training cost and time.

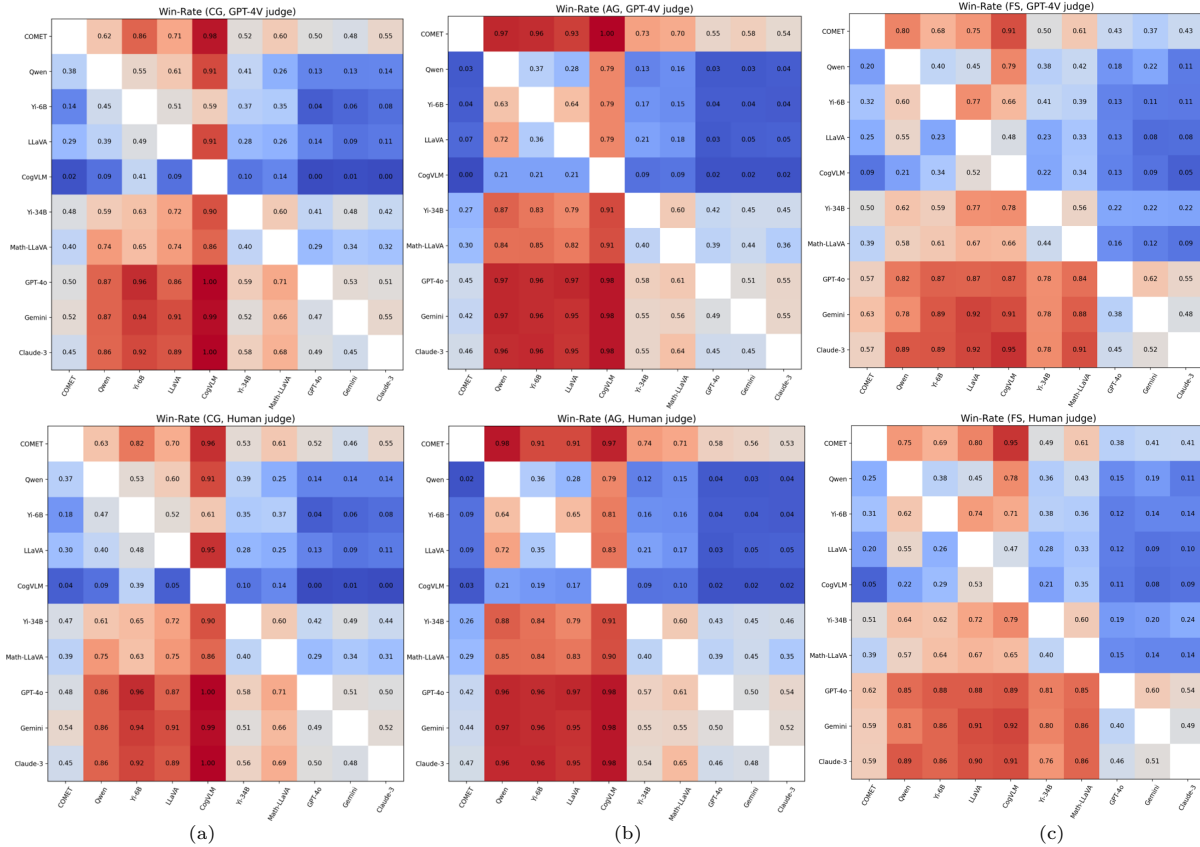


Figure C2 The statistics of win rate between models over 6,000 rounds. The three subfigures, (a), (b), and (c), respectively represent tasks CG, AG, and FS. For each subfigure, the upper section represents the win rate judged by GPT-4V, while the bottom section represents the win rate judged by humans.

On CMM12K, our model’s overall score of 33.89% remarkably exceeds all open-source baselines, with approximately a 9.33% performance advantage over the second-place Yi-VL-34B. Specifically, we conducted statistics on two modalities and three problem types, totaling $2 \times 3 = 6$ categories. When compared with closed-source models, COMET outperforms or matches the performance of GPT-4o on Multi-MCP, Multi-MPP, and Multi-MWP.

As the open-source models, COMET achieved SOTA in 5 of 6 categories, only slightly lagging behind Yi-VL-34B in single modal MFP by a small margin (-3%). Compared with the baseline of the same parameter size, our model leads in all types of problems. For CogVLM, which has twice the parameter size of ours, our model maintains a lead of more than 15% in all tasks. In summary, our model achieves relatively excellent problem-solving performance in all types of problems with a smaller parameter size (approximately 20% of Yi-VL-34B).

Appendix C.4.2 Performance of MPP

In regards to the MPP in the CMM12K dataset, we draw from the evaluation logic of machine translation, comparing each LMM’s response with the standard answer and calculating BLEU and ROUGE scores. The standard answer here is derived from manual annotation, encompassing not only the geometric elements and their relationships but also the complete proof process. By calculating BLEU and ROUGE, we can approximately determine whether the output of the LMM is in accordance with mathematical grammar and proof logic. Table C2 displays the response quality of our model and the baseline model on single-modal and multi-modal proof problems. The results indicate that our model is superior to all baselines as far as the response quality of the proof problems is concerned.

Appendix C.5 Performance in different difficulty levels

In this section, we divide the testset of the CMM12K dataset into three subsets based on difficulty: low, medium, and high. These subsets are used to evaluate the model performance at different difficulty levels. Specifically, we measure the difficulty of questions using the “estimated solving time” label in CMM12K. The longer the estimated solving time, the higher the difficulty.

The results in Table C3 show the performance of COMET and the untrained backbone model Qwen-VL-Chat across different difficulty levels. For the objective metric, namely solving accuracy, we observe that COMET outperforms the backbone model at all difficulty levels. Notably, on the highest difficulty questions, COMET achieves a 5.67% improvement in accuracy.

For the subjective evaluation results across the three types of tasks, COMET achieves improvements in all dimensions. On the low-difficulty subset, COMET even achieves near-perfect scores in some dimensions, such as the knowledge point relevance (KR) of the controllable generation (CG) task. As the difficulty increases, both COMET and Qwen-VL-Chat experience a decline in scores, which aligns with intuition. At the highest difficulty level, COMET demonstrates significant improvements across the three tasks. For example, it achieves score increases of 1.14, 2.22, and 3.84 in logical correctness (LC) of fine-grained solving (FS), LC of controllable generation (CG), and reasoning rationality (RR) of analogy generation (AG), respectively.

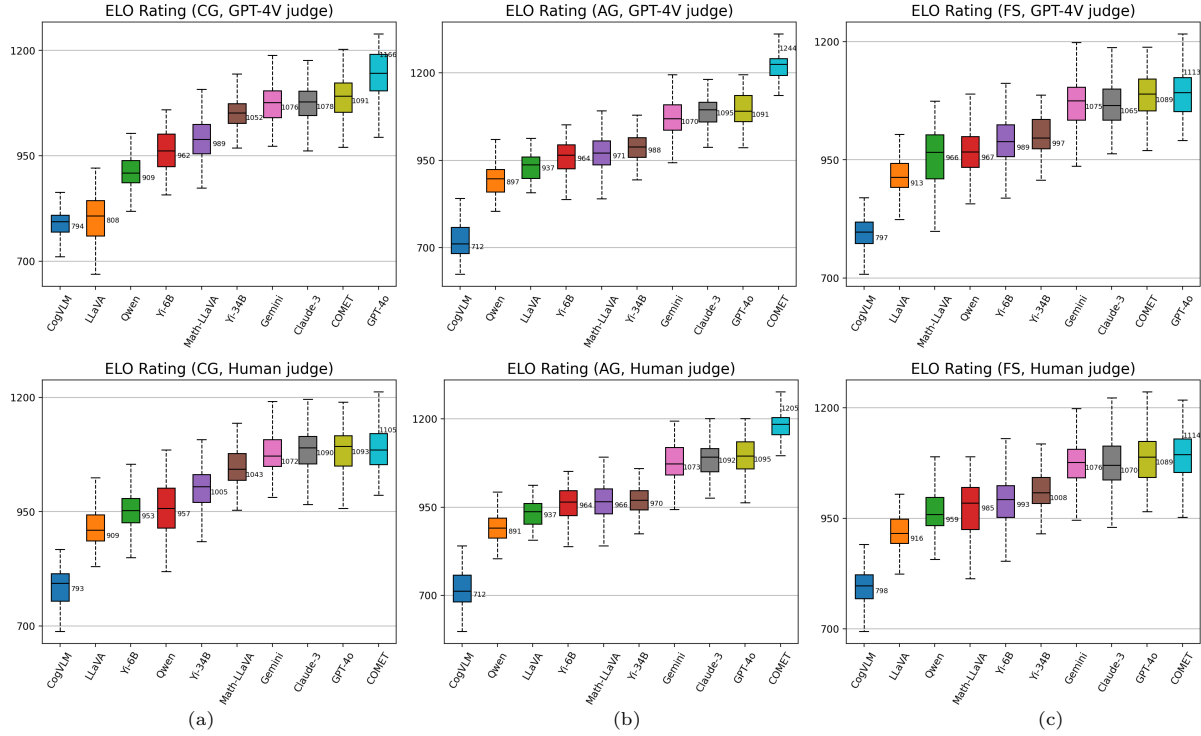


Figure C3 The statistics of the ELO rating over 6,000 rounds are presented. Similar to Figure C2, the three subfigures represent CG, AG, and FS, respectively. In each subfigure, the upper section illustrates the ELO rating as assessed by GPT-4V, while the bottom section depicts the ELO rating determined by humans.

The above analysis confirms that the proposed fine-tuning framework can generalize the improvement of model performance on math problems with different difficulty levels. Across tasks at all difficulty levels, COMET consistently outperforms the backbone model. This demonstrates the effectiveness of the three-stage fine-tuning framework based on the “Cone of Experience” theory.

Appendix C.6 Ablation study

To validate the effectiveness of the three-stage fine-tuning framework proposed in this paper, we designed the following ablation experiments. We refer to the model after the i^{th} stage of fine-tuning as S_i , where $i \in \{1, 2, 3\}$. We preserved all checkpoints of the fine-tuning stages: S_0 - the original backbone model Qwen-VL-Chat without any fine-tuning. S_1 - the model after the first stage of fine-tuning, injected with symbolic experience. S_2 - based on S_1 , the model after the second stage of fine-tuning, infused with iconic experience. S_3 - the model after the third stage of fine-tuning, incorporating direct experience.

We first obtained these four models’ responses on the test set regarding the CG, AG, and FS tasks, and scored them based on GPT-4V in 15 fine-grained dimensions. Figure C4 shows the score changes on 15 dimensions for the three types of tasks. The results indicate that with the deepening of the three-stage fine-tuning, the model’s scores in all dimensions show an increasing trend. The absolute performance improvements at each fine-tuning stage are shown in Table C4, these results show that on most capability dimensions, $\Delta_2 = \max\{\Delta_1, \Delta_2, \Delta_3\}$, meaning that the second stage contributed the most to the performance improvement in the three-stage fine-tuning framework based on the “Cone of Experience” educational theory.

We also observe some exceptions, including the four dimensions FS-AC, AG-CC, AG-RR, and AG-SR, whose common feature is that the original backbone model performs poorly, and the first stage of training plays a key role in improving the performance on these dimensions. In all dimensions of each task, further improvements in model performance can be achieved through continuous injecting of direct experience in the third stage of fine-tuning.

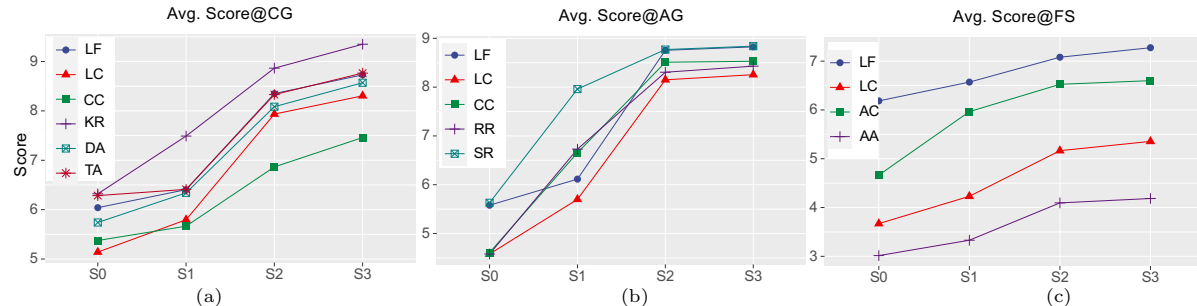


Figure C4 The average scores across various dimensions for COMET at each stage of training.

Table C1 Performances of LMMs on GSM8K, TAL-SCQ5K-CN, and CMM12K. Here Single/Multi indicates single modal/multi modal. All results are reported in terms of Acc (%), **bold** indicates optimal performance while underline indicates suboptimal.

Model	Size	GSM8K	TAL-SCQ5K-CN	CMM12K						
				Total	MCP		MFP		MWP	
					Single	Multi	Single	Multi	Single	Multi
<i>closed-source models</i>										
GPT-4o	-	78.99	65.50	41.89	54.00	32.67	<u>55.33</u>	<u>28.67</u>	<u>54.67</u>	<u>26.00</u>
Claude-3-Opus	-	85.74	57.25	34.56	<u>46.67</u>	19.33	56.67	10.00	60.67	14.00
Gemini-1.5-Pro	-	<u>81.88</u>	<u>62.75</u>	<u>41.56</u>	54.00	<u>30.00</u>	50.00	35.33	46.67	33.33
<i>open-source models</i>										
Yi-VL-6B	6B	22.52	6.55	21.11	14.00	18.67	32.00	18.67	28.67	14.67
Qwen-VL-Chat	7B	18.04	16.20	24.11	<u>27.33</u>	22.00	29.33	17.33	29.33	19.33
LLaVA-1.6	7B	18.65	10.05	19.78	17.33	<u>24.67</u>	27.33	11.33	22.67	13.33
Math-LLaVA	13B	8.57	27.75	13.22	21.33	18.00	4.00	16.67	5.33	14.00
CogVLM	17B	8.87	5.35	14.11	4.67	22.67	22.00	7.33	17.33	10.67
Yi-VL-34B	34B	38.66	7.60	<u>24.56</u>	16.67	16.67	38.67	<u>19.33</u>	<u>34.00</u>	<u>22.00</u>
COMET (ours)	7B	<u>28.89</u>	<u>22.05</u>	33.89	35.33	34.00	<u>36.00</u>	29.33	35.33	33.33

Table C2 Performances of LMMs on CMM12K with MPP, **bold** indicates optimal performance (higher is better for all metrics).

Model	Size	Total								Single				Multi								
		ROUGE				BLEU				ROUGE				BLEU								
		@1	@2	@L	@4	@1	@2	@3	@4	@1	@2	@L	@4	@1	@2	@L	@4					
<i>closed-source models</i>																						
GPT-4o	-	0.78	0.63	0.59	0.75	0.57	0.52	0.48	0.87	0.64	0.69	0.69	0.62	0.59	0.57	0.80	0.70	0.71	0.71	0.60	0.53	0.41
Claude-3-Opus	-	0.82	0.65	0.66	0.68	0.52	0.49	0.50	0.83	0.71	0.70	0.62	0.64	0.61	0.52	0.86	0.66	0.66	0.68	0.59	0.51	0.43
Gemini-1.5-Pro	-	0.76	0.70	0.72	0.72	0.56	0.48	0.49	0.80	0.68	0.70	0.61	0.60	0.62	0.54	0.81	0.64	0.64	0.67	0.63	0.52	0.45
<i>open-source models</i>																						
Yi-VL-6B	6B	0.32	0.20	0.27	0.32	0.15	0.10	0.07	0.32	0.18	0.26	0.29	0.14	0.09	0.06	0.33	0.20	0.28	0.33	0.15	0.10	0.07
Qwen-VL-Chat	7B	0.38	0.25	0.32	0.28	0.17	0.12	0.08	0.37	0.24	0.30	0.24	0.14	0.09	0.07	0.39	0.26	0.34	0.31	0.18	0.13	0.09
LLaVA-1.6	7B	0.35	0.21	0.23	0.27	0.11	0.07	0.05	0.37	0.21	0.22	0.25	0.10	0.06	0.04	0.33	0.20	0.25	0.29	0.13	0.08	0.05
Math-LLaVA	13B	0.39	0.21	0.22	0.29	0.17	0.22	0.12	0.34	0.26	0.30	0.26	0.16	0.10	0.08	0.29	0.26	0.34	0.32	0.20	0.13	0.10
CogVLM	17B	0.20	0.05	0.11	0.18	0.05	0.02	0.01	0.21	0.06	0.12	0.18	0.05	0.02	0.01	0.18	0.04	0.1	0.18	0.05	0.02	0.01
Yi-VL-34B	34B	0.38	0.20	0.27	0.34	0.16	0.11	0.07	0.37	0.20	0.25	0.30	0.14	0.09	0.06	0.40	0.23	0.31	0.38	0.19	0.13	0.09
COMET (ours)	7B	0.80	0.67	0.69	0.71	0.62	0.54	0.47	0.81	0.68	0.67	0.69	0.60	0.52	0.46	0.80	0.66	0.67	0.68	0.59	0.51	0.43

Appendix C.7 Case study

To demonstrate the strength of our model, we have selected some examples of CG, AG, and FS tasks respectively. Previous results, such as Figure C1, have already proven that Yi-VL-34B - a model with more than five times the number of parameters as ours, is a comparable competitor. Thus, for each task, we show the difference in response quality between our model, the backbone model Qwen-VL-Chat, and Yi-VL-34B when using the same prompt, the case details refer to Figure C5.

Case of CG In the example shown in Figure C5(a), LMM was asked to generate one problem based on the given planar geometric picture. Our model accurately captured the geometric elements in the picture and expressed the test problem using the correct mathematical language. In contrast, Qwen-VL-Chat failed to comprehend the content of the given picture, erroneously providing the condition ($\triangle ADC \sim \triangle AEC$, but both ADC and AEC are not triangles). For Yi-VL-34B, the problem it constructed was not based on the given image, hence not aligning with requirements.

Case of FS Given this problem, Qwen-VL-Chat correctly understood the elements in the picture, but its erroneous reasoning steps ($AD=AC=3\text{cm}$ is given, but actually $AD=2AC=6\text{cm}$) led to an incorrect final result. Yi-VL-34B made similar mistakes as Qwen-VL-Chat. However, our model first parsed the problem requirements, and then extracted the geometric elements of the given picture, and finally correctly reasoned step by step according to the problem to arrive at the correct answer ($DB=4\text{cm}$).

Case of AG We require LMMs to simulate the seed problem and construct a new problem. Each LMM first analyzes the ideas for the construction of the seed problem and then constructs a problem, and they are also asked to explain the thought process behind the constructed problem. Our model first understands the meaning of the problem, parses the content of the knowledge points tested (the sum of the interior angles of a triangle is 180°), and tests a similar knowledge point (determine the shape of a triangle based on its angles) by modifying the problem scenario. This demonstrates that the symbolic experience, especially graph knowledge injected in the first stage helps the model find similar knowledge points based on the given knowledge point.

For Qwen-VL-Chat, the problem it generates, there is a discrepancy between the idea of the problem and the content of the problem, and the generated problem has not been tested for data rationality and does not conform to logic (it is paradoxical that there are two angles of 40° and 65° in a right triangle). On the other hand, Yi-VL-34B correctly interprets the meaning of the seed problem, analyzes the knowledge points tested, and modifies the problem scenario by adding conditions. However, the quality of its problem can be further improved because it introduces an invalid condition (the length of BC is meaningless for solving $\angle B$). Although there is no logical issue with this constructed problem, compared to our model's response, the problem constructed by our model is more reasonable.

Table C3 Comparison of COMET and Qwen-VL-Chat at different difficulty levels, here diff represents difficulty, and acc indicates the model’s accuracy on the test subsets.

Diff	Model	ACC	FS				CG					AG					
			LF	LC	AC	AA	LF	LC	CC	KR	DA	TA	LF	LC	CC	RR	SR
Low	Qwen-VL-Chat	41.67	8.68	6.65	7.80	6.75	9.01	8.46	8.41	9.48	8.82	9.70	8.38	7.94	7.66	7.88	8.57
	COMET (ours)	54.67	8.72	8.01	8.63	7.80	9.43	9.43	8.91	9.99	9.23	9.63	9.67	9.51	9.58	9.56	9.93
Mid	Qwen-VL-Chat	20.33	7.51	4.17	5.98	2.48	7.75	6.43	6.85	8.11	7.36	7.96	7.00	5.45	5.74	5.47	7.05
	COMET (ours)	31.00	7.93	4.98	7.06	3.51	8.55	8.26	6.82	9.35	8.29	8.56	9.03	8.30	8.72	8.50	9.16
High	Qwen-VL-Chat	10.33	5.39	2.38	3.14	1.41	5.74	4.28	4.79	5.87	5.17	5.73	5.08	3.49	3.65	3.49	4.94
	COMET (ours)	16.00	5.80	3.52	4.21	2.32	7.22	6.50	5.24	7.65	7.05	6.97	8.38	7.01	7.79	7.33	8.11

Table C4 Performance improvements(Δ) of each evaluation dimension at each stage. Here, Δ_i represents the improvement in average score relative to the $(i-1)^{th}$ stage after the i^{th} stage training, the **bold** indicates $\max\{\Delta_1, \Delta_2, \Delta_3\}$.

Δ	FS				CG					AG					
	LF	LC	AC	AA	LF	LC	CC	KR	DA	TA	LF	LC	CC	RR	SR
Δ_1	0.385	0.560	1.300	0.315	0.365	0.655	0.290	1.170	0.600	0.125	0.535	1.120	2.045	2.150	2.330
Δ_2	0.510	0.935	0.560	0.765	1.945	2.140	1.200	1.375	1.745	1.920	2.640	2.450	1.855	1.575	0.810
Δ_3	0.195	0.190	0.075	0.090	0.380	0.370	0.595	0.485	0.485	0.435	0.070	0.105	0.020	0.125	0.070

CG 输入：学段范围-8年级、知识点-“相似三角形的判定”、题型-证明题、难易程度-简单、

任务：请你根据输入中给定的图片、学段范围、知识点、题型和难易程度，构造一道符合要求的原创数学证明题。

出题思路：本题考查相似三角形的判定定理，熟练掌握相似三角形的判定定理是解题的关键，我准备首先说明图中角、点、边的数值关系和位置关系，然后在此基础上考察证明2个指定的三角形相似。

原创试题：如图， $\triangle ABC$ 中，D、E分别是AB、AC上的点，且 $\angle AED = \angle B$ ，求证： $\triangle AED \sim \triangle ABC$ 。

出题思路：根据学段范围、知识点、题型和难易程度，构造一道证明题。

原创试题：证明：在图中， $AD=EC$ ， $\angle A = \angle E$ ，那么 $\triangle ADC$ 和 $\triangle ECB$ 是相似三角形。

首先，我们需要在图中表示出相似三角形。我们可以通过构造两个等边三角形来表示相似三角形。

原创试题：请证明图中的两个三角形是相似的。

(a) Case of Controllable Generation

FS 输入：如图所示，C、D是线段AB上两点，若 $AC=3\text{cm}$ ，C为AD中点且 $AB=10\text{cm}$ ，则 $DB=()$
A. 4cm B. 5cm C. 6cm D. 7cm

任务：你对输入中给定的数学题进行严谨具体的解答，要求一步一步地详细记录解题思路和过程。

详细解析步骤如下：这是一道关于线段长度的问题。根据题意，我们知道 $AC=3\text{cm}$ ，C为AD中点，所以 $AD=2AC=6\text{cm}$ 。又因为 $AB=10\text{cm}$ ，所以 $BD=AB-AD=10\text{cm}-6\text{cm}=4\text{cm}$ 。所以，答案是A。

详细解析步骤如下：根据题目，我们知道 $AC=3\text{cm}$ ，C为AD中点，所以 $AD=AC=3\text{cm}$ 。
1. 因为 $AB=10\text{cm}$ ，所以 $DB=AB-AD=10\text{cm}-3\text{cm}=7\text{cm}$ 。所以， $DB=7\text{cm}$ 。

根据题意， $AC=3\text{cm}$ ，C为AD中点，所以 $CD=2\text{cm}$ ，又因为 $AB=10\text{cm}$ ，所以 $DB=10\text{cm}-2\text{cm}=8\text{cm}$ 。因此答案是D 7cm。

(b) Case of Fine-grained Solving

AG 输入：一个缺角的三角形ABC残片如图，量得 $\angle A=40^\circ$ ， $\angle B=65^\circ$ ，则这个三角形残缺前的 $\angle C$ 的度数为()
A. 55° B. 65° C. 75° D. 85°

任务：请你模拟输入中给定的种子试题的出题思路，构造一道类似的数学试题。

种子题的出题思路：种子题的主要知识点是三角形的内角和定理，以及三角形的性质。题目中给出了三角形ABC的两个内角A和B的度数，要求求解三角形残缺前的角C的度数。这是一个典型的三角形内角和的问题，通过已知的两个内角的度数，可以求解出第三个内角的度数。题目的难度适中，主要考察学生对三角形内角和定理的理解和应用。

原创试题的出题思路：原创题的知识点仍然是三角形的内角和定理，以及三角形的性质。为了增加题目的难度，我打算先让学生通过已知的两个内角的度数，求解出第三个内角的度数，然后在此基础上考察该三角形的形状。题目的场景我打算设定为一个三角形的风筝，风筝的形状和三角形的性质相吻合，这样可以增加题目的趣味性。题目的数据我打算设置为 $\angle A=40^\circ$ ， $\angle B=70^\circ$ ，这样可以保证题目的解答过程是可行的。

原创试题：小明拾到了一个残缺的三角形的风筝，量得 $\angle A=40^\circ$ ， $\angle B=70^\circ$ ，则这个三角形风筝残缺前的形状为____三角形。

种子题的出题思路：这是一道关于三角形的题目，难度适中。题目中给出一个已知的三角形，要求计算出这个三角形的未知角的度数。这个题目可以考察学生对三角形内角和的掌握，以及根据已知条件计算未知角的能力。

原创试题的出题思路：这是一道关于相似三角形的题目，难度较大。题目中给出两个不同的三角形，要求判断这两个三角形是否相似，并给出判断的依据。这个题目可以考察学生对相似三角形的判断能力，以及根据图形的性质进行推理的能力。

原创试题：在直角三角形ABC中， $\angle A=40^\circ$ ， $\angle B=65^\circ$ ，求 $\angle C$ 的度数。

种子题的出题思路：该题主要考察的是三角形的内角和定理，即三角形三个内角的和为 180° 。通过给出一个残缺的三角形，要求计算残缺前的 $\angle C$ 的度数，从而考察学生对三角形内角和的理解和应用。

原创试题的出题思路：我决定构造一个类似的试题，但采用不同的场景。我选择了一个等腰三角形，并给出其顶角和底边长度，要求其计算其底角的度数。这样，我既保留了三角形内角和定理的考察，又改变了问题场景，提高了试题的新颖性。

原创试题：如图，等腰三角形ABC的顶角 $\angle A$ 为 60° ，底边BC的长度为4cm，求 $\angle B$ 的度数。

(c) Case of Analogy Generation

Figure C5 Case of the three tasks. The key correct parts of responses are highlighted in green and the incorrect ones in red.

Appendix D Annotation details

In this section, we list the annotation guidelines used for the parts of this work that involve human annotation.

- **Construction of iconic experience.** In Appendix A.3, we need to construct the experience of stem generation, problem-solving, and failure. For the stem generation experience, we use GPT-4V as the primary annotator, with human involvement for review. Specifically, we employ a rejection sampling strategy to exclude responses from GPT-4V that do not meet the required standards. Figure D1 (left and middle) lists the human annotation guidelines for reviewing. For the problem-solving experience, human annotators were asked to provide detailed explanations of the questions, Figure D1 (right) lists the guidelines.

- **Human preference ranking.** For the multiple responses generated by the LMM for the same query, we ask human annotators to rank them based on preference. Specifically, we expect the human annotators to identify the highest-quality response and the lowest-quality response. Figure D2 lists the annotation guidelines for annotators.

- **Human as referees.** We hired 20 students as annotators, each annotating 300 pieces of anonymized PK data, resulting in a total of 6,000 pieces of data, the annotators were asked to choose the better of the two given responses. For the small amount of data that could not be annotated by some annotators, we handled it separately at the end. Figure D3 shows the annotation guidelines.

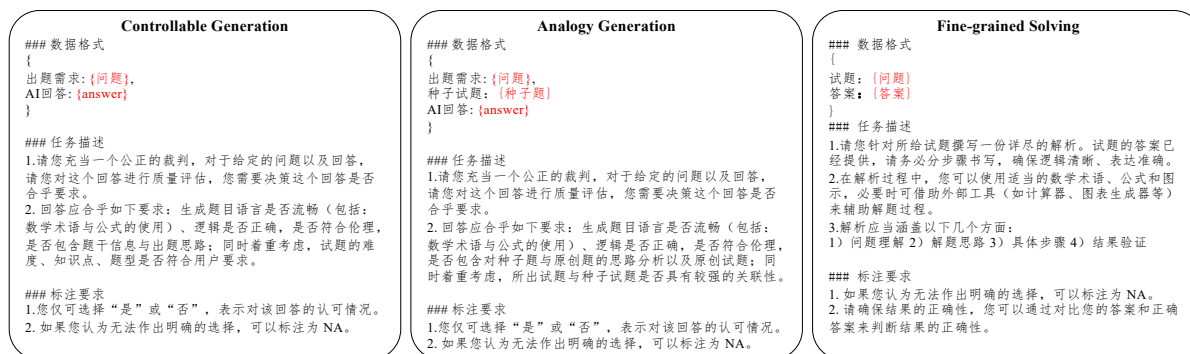


Figure D1 The annotation guidelines for data construction of iconic experience.

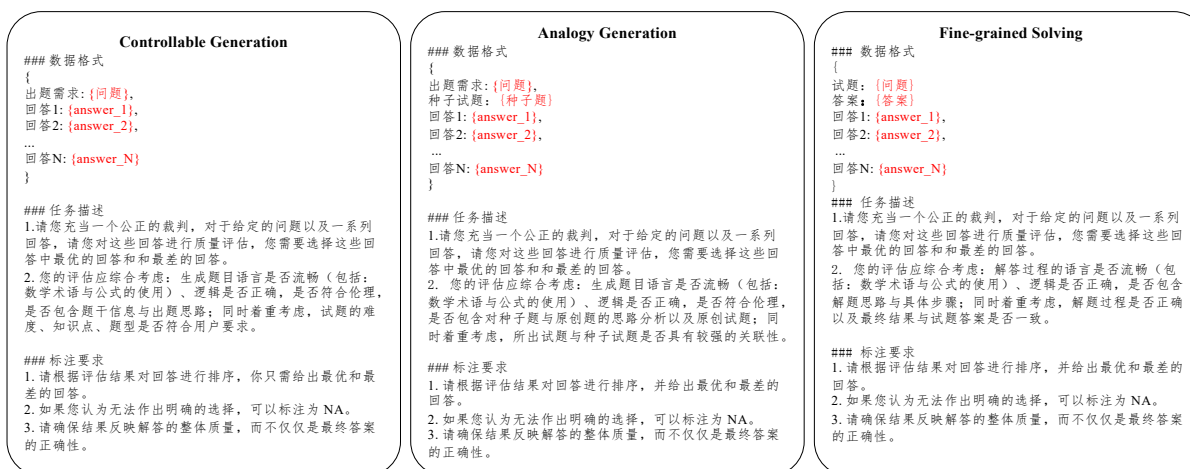


Figure D2 The annotation guidelines for human preference ranking.

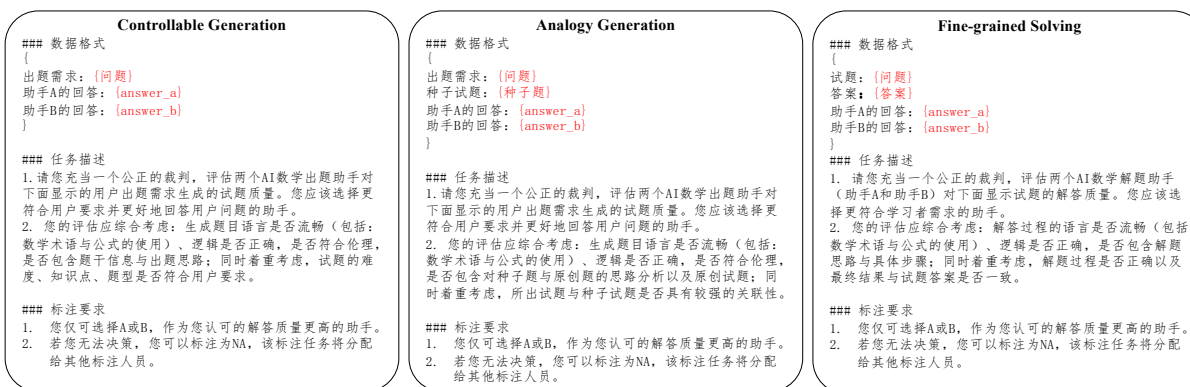


Figure D3 The annotation guidelines for human judge the better LMM's response in area mode.

References

- 1 Yang Z, Ding M, Lv Q S, et al. Gpt can solve mathematical problems without a calculator. 2023. ArXiv:2309.03241
- 2 Wei T W, Zhao L, Zhang L C, et al. Skywork: A more open bilingual foundation model. 2023. ArXiv:2310.19341
- 3 Sun T X, Zhang X T, He Z F, et al. Moss: Training conversational language models from synthetic data. 2023. ArXiv:2307.15020
- 4 Ji Y J, Deng Y, Gong Y, et al. Exploring the impact of instruction data scaling on large language models: An empirical study on real-world use cases. 2023. ArXiv:2303.14742
- 5 Ma S J, Luo Y W, Yang Y. Person-as-based Student Grouping using reinforcement learning and linear programming. *Knowledge-Based Syst*, 2023, 281: 111071
- 6 Rafailov R, Sharma A, Mitchell E, et al. Direct preference optimization: Your language model is secretly a reward model. In: *Proceedings of Advances in Neural Information Processing Systems*, 2024, 36
- 7 Bai J Z, Bai S, Yang S S, et al. Qwen-vl: A frontier large vision-language model with versatile abilities. 2023. ArXiv:2308.12966
- 8 Rajbhandari S, Rasley J, Ruwase O, et al. Zero: Memory optimizations toward training trillion parameter models. In: *SC20: International Conference for High Performance Computing, Networking, Storage and Analysis*. 2020. 1-16
- 9 Hu E J, Shen Y L, Wallis P, et al. Lora: Low-rank adaptation of large language models. 2021. ArXiv:2106.09685
- 10 Cobbe K, Kosaraju V, Bavarian M, et al. Training verifiers to solve math word problems. 2021. ArXiv:2110.14168
- 11 Achiam J, Adler S, Agarwal S, et al. Gpt-4 technical report. 2023. ArXiv:2303.08774
- 12 Gemini Team. Gemini 1.5: Unlocking multimodal understanding across millions of tokens of context. 2024. ArXiv:2403.05530
- 13 Young A, Chen B, Li C, et al. Yi: Open foundation models by 01.ai. 2024. ArXiv:2403.04652
- 14 Liu H T, Li C Y, Wu Q Y, et al. Visual instruction tuning. In: *Proceedings of Advances in Neural Information Processing Systems*, 2024. 36
- 15 Wang W H, Lv Q S, Yu W M, et al. Cogvlm: Visual expert for pretrained language models. 2023. ArXiv:2311.03079
- 16 Shi W H, Hu Z Q, Bin Y, et al. Math-LLaVA: Bootstrapping Mathematical Reasoning for Multimodal Large Language Models. 2024. ArXiv:2406.17294
- 17 Elo A E. The proposed uscf rating system, its development, theory, and applications. *Chess Life*, 1967, 22: 242-247
- 18 Zheng L M, Chiang W L, Sheng Y, et al. Judging llm-as-a-judge with mt-bench and chatbot arena. In: *Proceedings of Advances in Neural Information Processing Systems*, 2024. 36
- 19 Papineni K, Roukos S, Ward T, et al. Bleu: a method for automatic evaluation of machine translation. In: *Proceedings of the 40th Annual Meeting of the Association for Computational Linguistics*, 2002. 311-318
- 20 Rouge, Lin C Y. A package for automatic evaluation of summaries. In: *Proceedings of Workshop on Text Summarization of the Association for Computational Linguistics*, 2004. 74-81