Information Fusion 119 (2025) 103037

Contents lists available at ScienceDirect

Information Fusion

journal homepage: www.elsevier.com/locate/inffus

Multi-objective math problem generation using large language model through an adaptive multi-level retrieval augmentation framework

Jianwen Sun [a,b], Wangzi Shi [a,b], Xiaoxuan Shen [a,b], Shengyingjie Liu [a,b], Luona Wei [c], Qian Wan [a,b,*]

[a] National Engineering Research Center of Educational Big Data, Central China Normal University, Wuhan 430079, China
[b] Faculty of Artificial Intelligence in Education, Central China Normal University, Wuhan 430079, China
[c] College of Electronics and Information Engineering, South-Central Minzu University, Wuhan, 430074, China

ARTICLE INFO

Keywords:
Math problem generation
Large language model
Retrieval-augmented generation
Educational application
Generative artificial intelligence

ABSTRACT

Math problems are an important knowledge carrier and evaluation means in personalized teaching. Their high cost of manual compilation promotes the research of math problem generation. Many previous studies have focused on the generation of math word problems, which are difficult to meet the real teaching needs due to the single task-objective orientation and small differences in generation results. By fusing external knowledge through retrieval-augmented generation (RAG), large language model (LLM) can generate a variety of math problems, but the generated results still have limitations such as poor knowledge consistency, uncontrollability, and high computational cost. In this paper, we propose the task of multi-objective math problem generation (MMPG). This task introduces the triple objectives of generation including "question type, knowledge point and difficulty" in respond to teaching needs in real scene. To the best of our knowledge, this is the first study considering multiple objectives on the process of math problem generation. Based on this, we further design an adaptive multi-level retrieval augmentation framework (AMRAF) for LLM to generate multi-objective math problems. This plug-and-play framework can effectively improve the generation performance without parameter tuning of the target model due to the fine-grained information retrieval and fusion. To verify the effectiveness of the proposed framework and provide a benchmark for subsequent research, we construct an MMPG dataset containing 9,000 samples. Experimental results demonstrate the superiority and effectiveness of our framework.

1. Introduction

Math problem aims to provide teachers with coherent teaching materials and student performance feedback to aid in dynamic adjustments to teaching strategies, while also providing correction guidance for self-assessment and learning methods of students. As an important knowledge transfer carrier and learning assessment means in the teaching process, math problem is an indispensable educational resource [1–4]. However, the compilation of high-quality math problems always requires a significant investment of time and effort by experienced teachers. The manual compilation of math problems is not only costly, but also inefficient, making it hard to meet the increasingly large and diverse demand for teaching or assessment needs [5].

In view of unique educational value and application potential, math problem generation (MPG) has attracted widespread attention from researchers and has made some achievements [2,6–12]. Specifically, early research is carried out based on the design of rules and templates. Through the manual abstraction of existing problems, machine generates problem randomly according to obtained feature template [2,8]. Subsequently, inspired by the widespread success of recurrent neural networks (RNN) in the field of natural language processing, temporal neural networks such as long short-term memory and gated recurrent units are introduced into the field of MPG. These methods make full use of sequential feature and generate the problems that are closer to human expression habits in the form of sequence-to-sequence (Seq2Seq) [10,13]. In response to the issue of low generation efficiency caused by the difficulty of parallel inference in RNN, the attention mechanism has been widely used in MPG algorithm due to its superior long-sequence modeling capability and parallel inference efficiency. The introduction of attention mechanism on Seq2Seq model can better capture the semantic correlation between topics and expressions, and generate more coherent and accurate math problems [11,12]. The

* Corresponding author at: National Engineering Research Center of Educational Big Data, Central China Normal University, Wuhan 430079, China.
E-mail address: wanq8228@ccnu.edu.cn (Q. Wan).

https://doi.org/10.1016/j.inffus.2025.103037
Received 22 November 2024; Received in revised form 24 January 2025; Accepted 17 February 2025
Available online 25 February 2025

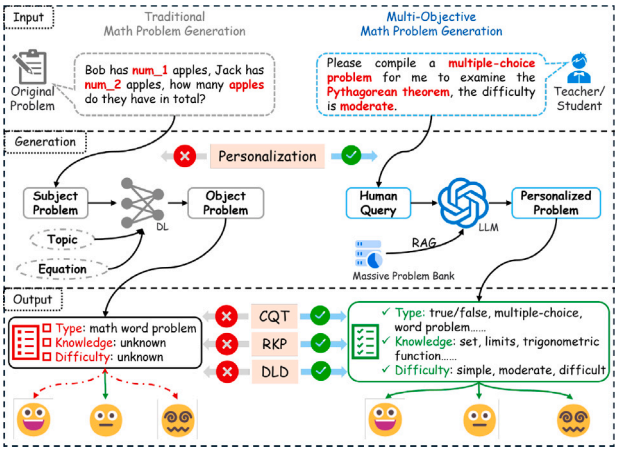

Fig. 1. Differences between MPG and MMPG, in which CQT means Controllable Question Types, RKP means Response Knowledge Points and DLD means Difficulty Level Differentiation. MMPG allows for the input of personalized user querys, and guides the generation process of math problem from multiple dimensions, which is more in line with the logic of teaching application.

above research demonstrates that the development of deep learning technology has effectively promoted the progress of MPG research.

Recently, the intelligent emergence of large language models (LLMs) has brought a new technical path for MPG research. Through training on massive texts, LLMs initially have the ability to understand and generate natural language, and acquire rich world knowledge. Techniques such as in-context learning (ICL) and retrieval-augmented generation (RAG) can supplement domain knowledge, guide generation patterns, and prompt content creation in the target domain [14,15]. Many researchers have been exploring the possibilities of combining LLMs with teaching process [3,16]. Compared to traditional methods, LLMs are more likely to generate high-quality, diverse, and personalized problems, but there are still two limitations to its deep enabling of MPG.

In terms of application, the requirement of teaching practice determines that high-quality math problems should meet the multiple objectives. **Controllability of question type**. Different learning objectives and teaching activities have complex and changeable requirements for question types. Only the flexible combination of multiple types of problems can fully examine the cognitive abilities of learners at different levels. Current mainstream research is aimed at generating math work problems and lacks consideration for other question types. **Responsiveness of knowledge point**. Math problems are required to accurately reflect course objectives and highly related knowledge points to endow them with training and assessment functions. MPG dataset with accurately annotated knowledge points is relatively scarce, which limits the usability of MPG in educational scenarios. **Difficulty level differentiation**. The learning progress and abilities of different learners vary widely. MPG should consider gradient variations in difficulty to support personalized learning. Previous research does not pay attention to the change of difficulty level, and the practice of rewriting simple problems cannot meet the needs of personalized teaching. As shown in Fig. 1, previous MPG studies fail to fully consider the application requirements of different dimensions at the beginning of formulating target tasks, so there are different degrees of application limitations.

In terms of technology, although LLMs based methods have made progress in the quality and diversity of MPG, there are still problems such as lack of domain knowledge, poor controllability, and high training costs. Based on accurate retrieval of external knowledge, RAG utilizes ICL ability to enhance the accuracy and consistency of generated content without modifying LLM parameters, which is an effective technical approach to deal with the above problems. However, naive RAG also faces the following challenges while enhancing the LLMs performance. Firstly, most RAG frameworks lack a gating mechanism, performing retrieval operations regardless of the difficulty of the query, which fails to strike a balance between generation quality and response speed. It not only increases the computation loss but also reduces the real-time performance of system, and may even be self-defeating in the face of simple queries. Secondly, math problems have characteristics of fewer characters, containing symbols, and a large amount of information, the traditional retrieval methods oriented towards large-scale documents are not adequately adapted. The relevance and diversity of retrieval results significantly influence the output quality of LLM, it is necessary to explore a fine-grained retrieval algorithm specialized for math problems.

To address the above two aspects of issues, this paper proposes a novel MPG task, named multi-objective math problem generation (MMPG). This task responds to the application needs in real teaching scenarios by introducing generation objectives including "question type-knowledge point-difficulty". To tackle the challenge of finer-grained controllability and knowledge consistency, we propose an adaptive multi-level retrieval augmentation framework (AMRAF) for LLM to generate multi-objective math problems. Specifically, we first fine-tuning a mentor model through domain-specific data, achieving adaptive simple task filtering and retrieval function activation during generation. Secondly, a multi-level retrieval mechanism and a fine-grained feature reordering algorithm are designed from structured and unstructured perspectives. Finally, the framework combines the suggestions of mentor model with retrieved results to build prompt to enhance MMPG performance for LLM. To evaluate the effectiveness of proposed framework, a Chinese MMPG dataset containing 9000 samples is constructed. Experimental results indicate that the proposed framework significantly improves the objective response, language fluency, and mathematical logic of the generated results, and has lower application costs.

The contributions of this paper can be summarized as follows:

- We propose multi-objective math problem generation considering question type, knowledge point, and difficulty. To the best of our knowledge, this is the first study integrate multiple objectives into the MPG process.
- An adaptive multi-level retrieval augmentation framework is proposed to improve MMPG performance of general LLM. This plug-and-play framework works with any open or closed source LLM without parameter tuning of the target model.
- The mentor model and multi-level retrieval mechanism are constructed to support adaptive fine-grained information retrieval and fusion, improving the efficiency and quality of generation.
- An MMPG Chinese dataset containing 9000 samples is constructed, fully verifying the effectiveness of the proposed framework and providing a benchmark for future research.

2. Related work

MPG garners significant attention from researchers due to its potential to enhance educational technologies and promote personalized learning. With the development of RAG techniques, using LLMs for generating math problems becomes a mainstream research trend. This paper provides a literature review on math word problem generation (MWPG) and retrieval-augmented generation (RAG), offering foundational knowledge crucial for understanding the proposed methods in this study.

2.1. Math word problem generation

Current research on MPG predominantly focuses on the generation of Math Word Problem (MWP). Early methods in this area primarily rely on the design of feature templates [2,17] or rewriting mechanisms [1,18]. For instance, Deane et al. [19] developed a prototype system for automatic MWP generation by using a semantics

frame to represent linguistic content. Wang et al. [7] used binary expression trees to structure the narratives of MWP and recursively generated natural language stories through a bottom-up traversal of these trees. Koncel-Kedziorski et al. [1] proposed a rewriting algorithm that constructs new texts by replacing words and phrases with thematically appropriate alternatives. Template-based methods are often constrained by the need for extensive and limited feature engineering, while rewriting-based methods, though more flexible, are restricted by existing formulas and lack creativity.

Neural network-based methods generate MWP in an end-to-end manner from equations and topics [10–12,20]. Zhou et al. [10] designed a neural network with two encoders to fuse equations and topics for generating relevant MWP. Liyanage et al. [11] generated MWP using a LSTM network enhanced with input features such as characters, words, and part-of-speech tags. Cao et al. [20] introduced a topic selector and controller to mitigate topic drift, along with a pre-training phase and a commonsense reasoning mechanism to avoid commonsense violations, achieving notable improvements. Wu et al. [12] utilized a topic-expression co-attention mechanism to capture the relationship between topics and expressions, optimizing the model using reinforcement learning, which significantly improved both the relevance and solvability of the generated problems.

Recent attempts have primarily focus on leveraging the vast knowledge and powerful generalization capabilities of LLM to generate MWP. Droria et al. [21] and Zong et al. [22] employed prompt learning and few-shot learning approaches to prompt the GPT-3, thereby automatically generating math problems on relevant topics. Their work demonstrated that GPT-3 can produce reasonable problems and solution steps for students to practice. Christ et al. [23] fine-tuned a LLM, MATHWELL, by manually constructing math problems that meet the standards of solvability, accuracy, and appropriateness, enabling the automatic generation of math word problems suitable for K-8 students. Although the research of Tang et al. [24] and Mitra et al. [25] did not directly focus on the automatic generation of new math problems, it is closely related. Their primary goal was to further train and fine-tune open-source models by utilizing new problems generated by state-of-the-art LLMs to enhance the ability of LLMs to solve math problems.

2.2. Retrieval-augmented generation

In recent years, LLMs have integrated into our life and work scenarios due to their amazing versatility and intelligence. However, in practice, LLMs still face challenges such as lack of domain knowledge and creating misleading illusions. To cope with these challenges, RAG techniques emerged. Early RAG framework effectively combines external knowledge sources with language models by integrating the three core components of indexing, retrieval, and generation, which significantly compensates for the shortcomings of LLMs in terms of knowledge reserves [26,27]. This technical architecture shows remarkable potential in diverse application scenarios, and gradually becomes a key technical route to improve the performance of LLMs. Especially after ChatGPT triggered the global AI technological change, RAG has gained widespread attention for its unique advantages, and has been widely used in downstream tasks such as code generation [28–30], Q&A systems [31,32] and creative writing [33]. Notably, RAG eliminates the need to update model parameters that may lead to high cost or instability, providing a cost-effective external knowledge interaction mechanism for LLMs [34].

However, the native RAG approach, while performing well in knowledge-intensive tasks, still suffers from several key issues. First, introducing retrieval can significantly impact the efficiency of the system. Second, not all tasks require retrieval. Lastly, when errors occur in retrieval, appropriate countermeasures are needed. The purpose of retrieval is to ensure that the generation model can obtain relevant and accurate knowledge. If the retrieved documents are irrelevant, not only will they not bring any benefit, but they may also exacerbate the factual errors in the language model. For this reason, some researchers proposed adaptive retrieval-augmented generation as a novel solution.

Considering that certain queries are not imperative retrieval and that faster and more accurate responses can be obtained without activating the retrieval mechanism, researchers proposed the concept of adaptive retrieval. Asai et al. [35] introduced Self-RAG to selectively retrieve knowledge and incorporated a critic model to decide whether to retrieve. Additionally, Feng et al. [36] and Ren et al. [37] took direct prompting of LLMs for retrieval decision-making, as some studies observed that LLMs can recognize their knowledge boundaries to some extent [38,39]. Mallen et al. [40] proposed determining the complexity level of queries based on the frequency of the queried entities and suggested utilizing the retrieval module only when the frequency falls below a certain threshold. Furthermore, in some multi-hop question-answering tasks, external knowledge is required more than once, necessitating attention to when and how often retrieval should be triggered. Jeong et al. [41] trained a classifier to predict the complexity of incoming queries, allowing the selection of the most appropriate retrieval strategy. However, the above methods are all designed based on open-domain question-answering tasks or multi-hop tasks, and they determine whether to perform a retrieval based on the input query. We argue that it is not reasonable for MPG task, as the knowledge contained within each generation model varies. Therefore, we believe that the generation model can be allowed to output its initial answer first, and then, based on this initial answer, decide whether retrieval is necessary, that is, to be result-oriented, and retrieval if it is likely to assist the model.

In summary, the generation of math word problems using deep learning techniques and LLMs achieves diversity and innovation in problem generation. However, these methods still face challenges in generating personalized, multi-objective math problems. This paper proposes a novel approach that combines the advantages of adaptive retrieval and LLM to generate math problems that satisfy multiple objectives. Our method optimizes the synergy between the retrieval and generation modules, achieving the generation of multi-objective math problems.

3. Method

In this section, we first define the MMPG task under RAG, followed by a detailed introduction to overall structure and functional modules of the proposed framework. Fig. 2 and Algorithm 1 illustrate the overall structure diagram and workflow of the framework, respectively.

3.1. Task definition

In the RAG-based MMPG task, the objective is to combine the generation model $\mathcal{G}$ with the retrieval module $\mathcal{R}$ to generate math problems that satisfy multiple constraints. Specifically, given a problem request q (which includes a set of generation objectives, such as question type T, knowledge point K, and difficulty D), as well as an accessible corpus C containing a large number of example problems, the retrieval module $\mathcal{R}$ aims to retrieve the top-k examples D_r related to the problem request q from the corpus C. The generator $\mathcal{G}$ based on the request q and the retrieved examples D_r, generates a math problem Q that meets the required objectives. Formally, the MMPG task under RAG is defined as follows:

$$Q = \mathcal{G}(q, D_r), where\ D_r = \mathcal{R}(q, C)\ and\ q \supset \{T, K, D\} \tag{1}$$

The above task definition exemplifies a native RAG-based approach to MMPG. However, this basic formulation can be enhanced through adaptive retrieval mechanisms that selectively engage external knowledge only when beneficial. Our proposed AMRAF framework improves upon this foundation by incorporating mentor model guidance to determine when retrieval is necessary, leading to a more flexible formulation as detailed in Section 3.5, Eq. (3).

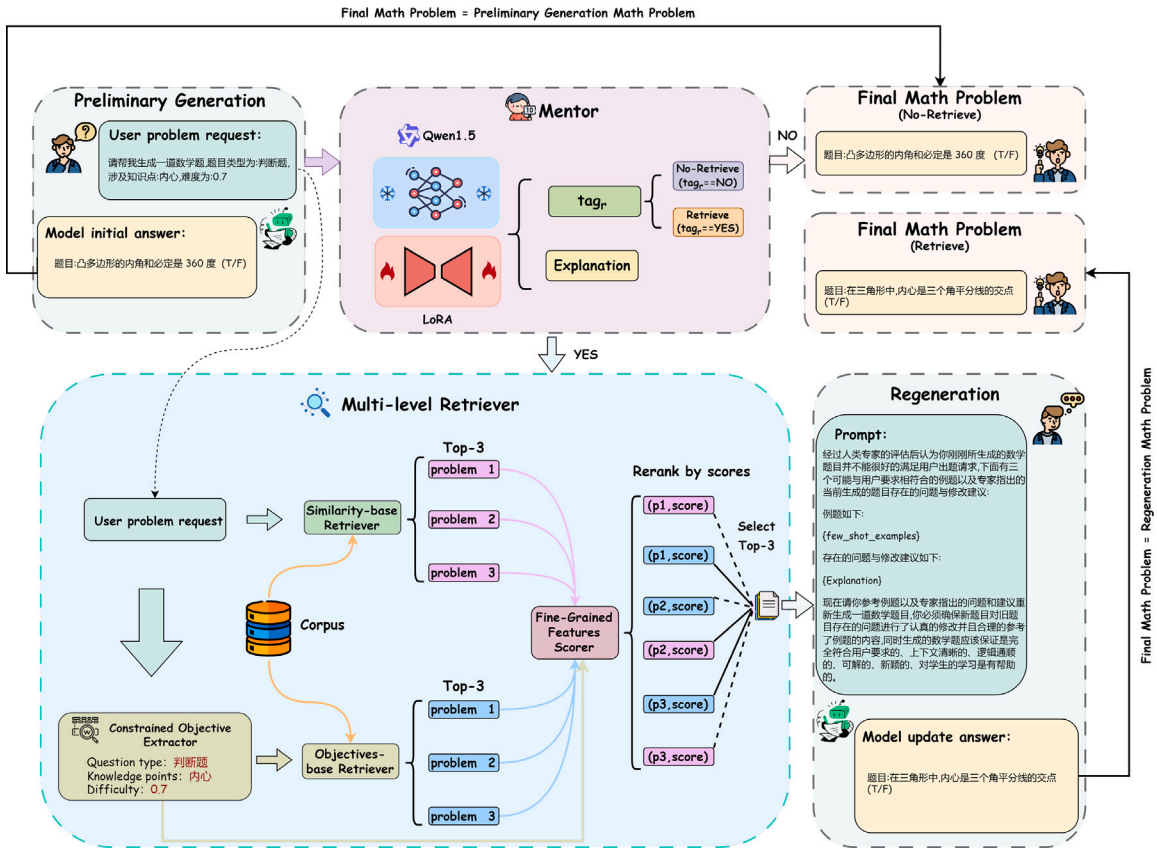

Fig. 2. Overview of AMRAF. The Chinese-English translation of the prompt template can be found in Appendix A.4.

3.2. Overview of AMRAF inference

Algorithm 1 shows an overview of the inference phase of AMRAF. In this study, we construct a mentor model $\mathcal{M}$ (see Section 3.3) and design a multi-level, fine-grained retrieval mechanism (see Section 3.4), which effectively improves the efficiency and quality of MPG. As shown in line 1 of Algorithm 1, given a problem request q, the generation model $\mathcal{G}$ first generates an initial math problem $\mathcal{Q}_{pre}$. Subsequently, as indicated in line 2 of Algorithm 1, a lightweight mentor model evaluates whether the current problem satisfies specific generation objectives. Based on the retrieval tag tag_r fed back by the mentor model, the system determines whether to perform the retrieval operation. If retrieval is triggered (see Algorithm 1 lines 3~10), the multi-level retriever $\mathcal{R}$ retrieves relevant examples from the problem corpus $\mathcal{C}$ (see Section 4.1.2). The retrieved examples $\mathcal{D}_r$ and the revision suggestions E from the mentor model are integrated as prompt, which is input into the generation model to regenerate a problem that satisfies the objectives. If retrieval is not triggered, the system directly outputs the initially generated problem. This process effectively optimizes both the quality and efficiency of problem generation.

3.3. Training mentor model

The timing of retrieval mechanism activation and the relevance of retrieval content are crucial in the process of problem generation. Previous studies demonstrated that blind retrieval augmentation could complicate simple problems, which might have negative impacts on certain tasks (for detailed analysis, see Appendix B).

As shown in the light purple module of Fig. 2, this paper designs a multifunctional mentor model that aims to provide intelligent supervision and assistance. The model takes the initial output of the generative model as a baseline. First, it evaluates whether the retrieval module needs to be activated to optimize resource utilization. Next, it analyzes the generated problems and offers detailed revision suggestions to enhance output quality. Additionally, the model also acts as a generation objective extractor (see Section 3.4.1), extracting key objective from problem requests. This multi-functional design significantly improves the efficiency and accuracy of the problem generation process. Specifically, we initialize the mentor model with Qwen1.5-14B-Chat [42] and fine-tune it using domain-specific data processed through both human and machine methods (see Section 4.1 and Appendix C for data processing details) to complete the necessary knowledge injection. To reduce computational costs and accelerate training, we use LoRA [43] for efficient fine-tuning of the mentor model. During training, the data is reformatted into alpaca-style [44] question-answer pairs for the instructional dataset. Through learning from this dataset, the mentor model can better judge the quality of the problem generated by the target LLM and adaptively activate subsequent retrieval modules when necessary. The adaptive approach based on the mentor model enhances context only when necessary to optimize performance, improving the robustness of model robustness in the MMPG task.

3.4. Multi-level retriever

This section details the design of the multi-level retriever (see Fig. 2 light blue module). As shown in line 3 of Algorithm 1, based on the retrieval tag $\text{tag}_r \in \{YES, NO\}$ provided by the mentor model, the framework can dynamically invoke the multi-level retriever. Specifically, if the retrieval tag $\text{tag}_r = YES$, the retrieval is triggered; otherwise, the retriever remains silent. The multi-level retriever primarily consists of three components: (1) generation objective extractor, (2) sentence-level recall, and (3) fine-grained features scorer.

3.4.1. Generation objective extractor

To retrieve the problems satisfying the generation objectives of user query from the problem corpus C (see Section 4.1.2) as much as possible, and improve the reliability of semantic vector matching. We design the generation objective extractor to locate objective fields precisely, providing key information support for subsequent retrieval components. Given the relatively low complexity of the question, we directly use the mentor model in Section 3.3 as the generation objective extractor and design a specialized prompt to meet the functional requirements. A demonstration of the prompt can be found in Appendix A.

3.4.2. Sentence-level recall

The sentence-level recall combines similarity-based as well as generation objectives-based retrieval methods.

Similarity – based Retriever. This method adopts an improved RAG architecture. Unlike traditional RAG, we segment the corpus based on complete problem, preserving the full semantics of each problem. The formal representation is as follows:, $C = \{c_1, c_2, \ldots, c_n\}$ represents the problem corpus, where each c_i represents a complete probelm. We use text-embedding-ada-002 model [45] from OpenAI to embed each c_i into a vector v_i : $v_i = Embed(c_i)$, and store all vectors in the Chromadb vector database. For a problem request q, we calculate its similarity score with the problem in the corpus: $score_{sim}(q, c_i) = cosine(Embed(q), v_i)$, and based on this, select the top-k example problems $d_{sim} = \{d_1, d_2, \ldots, d_k\}$.

Objectives – based Retriever. Although semantic similarity retrieval allows for multi-language and multi-modal searches and demonstrates strong robustness against spelling errors, it has limitations in capturing subtle differences between mathematical concepts, such as the confusion between "solutions of fractional equations" and "solving fractional equations". Additionally, this method relies on the quality of vector embeddings and is sensitive to out-of-domain terminology. To overcome these shortcomings, we introduce a generation objectives-based retrieval method. Each problem c_i in the corpus C is preprocessed by removing stop words and special symbols. The generation objective extractor $\mathcal{K}$ is applied to the problem request q to extract generation objectives: $\mathcal{W} = \mathcal{K}(q)$, which includes constraints on question type T, knowledge points K, and difficulty D. The BM25 [46] algorithm is then used to compute the relevance score: $score_{bm25}(q, c_i) = BM25(\mathcal{W}, c_i)$, selecting the top-k problems $d_{bm25} = \{d_1, d_2, \ldots, d_k\}$.

This hybrid strategy leverages the advantages of both semantic similarity and keyword matching, enhancing the accuracy and robustness of retrieval. However, optimizing the combination of different similarity scores remains a challenge. To address this issue, we introduce a fine-grained features scorer in Section 3.4.3.

3.4.3. Fine-grained features scorer

To optimize the hybrid retrieval results and control the number of examples injected into the prompt, we design a fine-grained features scorer. This scorer calculates a unified relevance score for each recalled example by considering fine-grained features such as question type, knowledge points, and difficulty.

We embed the values of the generation objectives $\mathcal{W}$ extracted by the Objectives-based Retriever into the embedding space using an embedding model. The problems retrieved by both methods, $D = \{d_{bm25}, d_{sim}\} = \{d_1, \ldots, d_{2k}\}$, are deduplicated to obtain the candidate set of problems D_r for reranking, in which the value of the corresponding generation objective of each problem is also converted to the embedding space using the same embedding model. By calculating the cosine similarity between the values of each generation objective, the similarity scores are weighted according to the importance of each generation objective. The formula to compute the relevance score for a given problem d_r is as follows:

$$score = \sum_{\substack{d_r \in D_r \\ x \in (T,K,D)}} \alpha_x f\left(E(\mathcal{W}(x)), E(d_r(x))\right) \tag{2}$$

where x represents the generation objectives T, K, and D, and α_x represents the corresponding weight coefficient for each generation objective. $\mathcal{W}(\cdot)$ refers to the values of the corresponding generation objectives extracted from the problem request, while $d_r(\cdot)$ represents the values of the corresponding generation objectives in the problems recalled by the retriever. E(·) is an embedding model, such as text-embedding-ada-002, and f(·) is a scoring function, such as cosine similarity.

This algorithm effectively reranks the set D_r, selecting the top-k problems as examples, which significantly enhances the quality of the final generated problems.

3.5. Multi-objective problem generation

AMRAF, as proposed in this study, adopts a more flexible strategy than the standard RAG, which performs retrieval operations indiscriminately in the MMPG task. AMRAF leverages feedback tag_r from the mentor model $\mathcal{M}$ to adaptively activate the subsequent retrieval module $\mathcal{R}$. When retrieval is triggered, the system retrieves relevant examples D_r and combines them with the revision suggestions E from mentor model, as external knowledge input for the generation model $\mathcal{G}$. This mechanism significantly improves the accuracy and consistency of regenerated problems. From a formal perspective, the MMPG task based on AMRAF can be defined as follows:

$$Q = \begin{cases} Q_{pre}, & tag_r = NO \\ \mathcal{G}(q, D_r, E, Q_{pre}), & tag_r = YES \end{cases} \tag{3}$$

where:

- $Q_{pre} = \mathcal{G}(q)$ represents the initial generation result of the generative model $\mathcal{G}$ for problem request q.
- $D_r = \mathcal{R}(q, C)$ represents the relevant examples retrieved by the retrieval module $\mathcal{R}$ based on problem request q and problems corpus C.
- $tag_r, E = \mathcal{M}(q, Q_{pre})$ denotes the predicted labels and suggested modifications given by the mentor model $\mathcal{M}$ based on the problem request q and the initial generated result Q_{pre}.

4. Experiment

4.1. MMPG dataset construction

4.1.1. Multi-objective math problems collection

The current mainstream math datasets (e.g., GSM8K [47] or MATH [48]) primarily focus on evaluating the math reasoning capabilities of LLMs. However, due to the lack of fine-grained annotations on problem characteristics, these datasets fail to meet the research requirements of the MMPG task. The MMPG task requires that the problem data contain explicit annotations that satisfy the generation objectives, such as question type, knowledge points, and difficulty.

In order to fill the benchmark gap in this area, we assemble a new MMPG dataset, with samples drawn from exam problems and in-class assignments ranging from elementary to middle school levels. As shown in Table 1, this dataset includes 9000 math problems, where math symbols are in LaTeX format. Each problem is manually annotated with three generation objectives: question type, knowledge points, and difficulty. Specifically, the dataset consists of 6 types of problems (true/false, proof, problem sets, multiple choice, problem solving, and fill-in-the-blank) and covers 3362 common knowledge points from grades 1 through 9 (combinations of single knowledge point is considered as new knowledge point). The difficulty of the problems (ranging from 0.1 to 1) is quantitatively measured based on five indicators: grade level, question type, readability, logical complexity, and required computation. Additionally, using text-embedding-ada-002 and uniform manifold approximation and projection (UMAP) [49], we visualize the knowledge points involved in some of the problems. The problems are embedded in a 2-dimensional space, allowing us to directly observe different clusters of problems based on varying knowledge points within the dataset. Relevant figures and analysis are provided in Appendix B.

Algorithm 1: Inference

Require: $\mathcal{R}$ (Retrieval Model),$\mathcal{M}$ (Mentor Model),$\mathcal{C}$ (Problems Corpus),$\mathcal{G}$ (Generative Model), $\mathcal{K}$ (Generation Objective Extractor),$\mathcal{S}$ (Fine-Grained Features Scorer)
Input: q (User problem request)
// request contains generation objectives on question type, knowledge points, and difficulty
Output: $\mathcal{Q}_{fin}$ (Generated response)
$\mathcal{G}$ generates preliminary answers $\mathcal{Q}_{pre}$ given q;
$\mathcal{M}$ predicts tag_r and **Explanation** given (q, $\mathcal{Q}_{pre}$);
if $tag_r == YES$ **then**
  $\mathcal{W}\leftarrow$ use $\mathcal{K}$ to extract objectives from q;
  $\mathcal{D}_r\leftarrow$Retrieve problems corpus $\mathcal{C}$ using $\mathcal{R}$ given (q, $\mathcal{W}$);
  for $d_r \in \mathcal{D}_r$ **do**
    use $\mathcal{S}$ to compute each d_r scores given $\mathcal{W}$;
  end
  $\mathcal{D}_r\leftarrow$Rerank $\mathcal{D}_r$ based on scores and select top-3;
  $\mathcal{G}$ regenerates answers $\mathcal{Q}_{fin} = \mathcal{G}(q, \mathcal{D}_r, E, \mathcal{Q}_{pre})$;
  /* E stand for Explanation */
  /* Detailed in Section 3 */
else
  Direct output $\mathcal{Q}_{fin} = \mathcal{Q}_{pre}$;
end

Table 1
The number of problems for different question types in the MMPG dataset, as well as the number of knowledge points covered by each question type. There is overlap in the knowledge points covered by different types of problems.

Question types	Problem count	Knowledge points count
True/False	1511	814
Proof	1310	569
Problem-Sets	145	112
Multiple-Choice	2576	1179
Problem-Solving	1839	972
Fill-in-the-Blank	1619	508
Total	9000	3362 (Deduplicated)

4.1.2. Data consistency check

To implement the training of the mentor model (see Section 3.3), at the same time, refine a more reliable retrieval corpus, we conduct a consistency check on the manually annotated MMPG dataset, verifying whether the problem aligns with the annotated generation objectives. We observe that SOTA LLM, represented by GPT-4 [50], can effectively simulate human experts in performing consistency checks [51]. Therefore, we first use reverse deduction to generate a problem request based on the three annotated generation objectives for each corresponding problem. This request, along with the corresponding problem, is input into GPT-4 as question-answer pairs for consistency checks. By using a carefully designed prompt (see Appendix A), the consistency check results generated by GPT-4 in this paper are highly consistent with the sampling comparison results of human experts.

The final consistency check results indicate that 10% of the problems in the MMPG dataset do not conform to the annotated generation objectives, while the remaining 90% are consistent with the annotations and are directly used as the external problem corpus for the framework proposed in this paper. Additionally, to alleviate the data imbalance issue during the training of the mentor model, we randomly select 7000 problems from the MMPG dataset and design generation objectives inconsistent with their labels to serve as negative sample examples. By now, the supervised data for the mentor model expands from 9000 to 16,000, with a roughly 1:1 ratio between positive and negative examples.

4.1.3. Data construction for evaluation

Based on the MMPG dataset constructed in Section 4.1.1, we build three fundamental seed sets: S_{types}, $S_{kpoints}$, and S_{diff}, which represent the sets of question types, knowledge points, and difficulty included in the dataset, respectively. First, we randomly sample a triplet $\langle t_i, k_i, d_i \rangle$ from these three basic seed sets. Then, based on the triplet, we construct a user problem request with multiple objectives on the question type, knowledge points, and difficulty (the syntax is as follows: "Generate a math question with question type: $\{t_i\}$, involving knowledge points: $\{k_i\}$, and with a difficulty of: $\{d_i\}$, without providing a solution process"). Constantly repeat the above construction process, as well as eliminate some unreasonable requests. Randomly, we generate an MMPG capability evaluation dataset containing 200 evaluation samples. This dataset includes a variety of problem generation requests covering 6 question types, 193 different knowledge points, and difficulty ranging from 0.1 to 1, which can effectively evaluate the multi-objective problem generation capabilities of the model. It is worth noting that the size of this evaluation sample set is close to the test set proposed by Christ et al. [23], and approximately 2 to 4 times larger than the manually evaluated samples used in other math word problem generation studies [1,10,22,52–55].

4.2. Baselines

In this paper, two main types of baseline models are selected for comparison: baselines without retrieval (which generate math problems without relying on an external corpus of example samples) and baselines with retrieval (which first retrieve relevant example problems from an external corpus and then generate math problems based on these examples).

Baselines without retrieval. We utilize the evaluation data constructed in Section 4.1.3 to assess the performance of existing open-source and closed-source models on the MMPG task. For closed-source models, we evaluate GPT-3.5-Turbo [56], developed by OpenAI, and Gemini-Pro [57], developed by Google. Both models excel at math reasoning, but to our knowledge, there has been no in-depth research into their controllable problem generation capabilities. For open-source models, we evaluate several high-performing pre-trained conversational language models, including ChatGLM3-6B [58], Baichuan2-7B-Chat, Baichuan2-13B-Chat [59], and Llama3-8B-Instruct [60].

Baselines with retrieval. We also evaluate the controllable problem generation capabilities of the above models under retrieval-augmented conditions. These models use the same retriever as our framework, and the retrieved example problems that are most relevant to the request from the user are placed at the top of the generation prompts to assist the model in completing the task more effectively.

The detailed description of each model is as follows:

- GPT-3.5-Turbo: developed by OpenAI with over 7B parametric count. Released in 2021, the model has been trained on a large amount of dialogue data and has strong math reasoning and natural language generation capabilities.
- Gemini-Pro: A large-scale language model, released by Google in 2023, boasts a participant count well in excess of 7B. The model is designed to handle a variety of natural language processing tasks such as text generation, translation, summarization and dialogue generation. It also excels at tasks such as math reasoning.
- ChatGLM3-6B: It is the third generation of dialogue pre-training model jointly released by Zhipu AI and KEG Lab of Tsinghua University. On the basis of retaining many excellent features such as smooth dialogue and low deployment threshold of the previous two generations of models, ChatGLM3-6B introduces a more powerful base model, more complete functional support and more comprehensive open source sequences.
- Llama-3-8B-Instruct: Meta AI's Meta Llama 3 series of 8B-parameter large language models, released in 2024, excel at complex tasks such as linguistic nuance, contextual understanding, code generation, and translation and dialogue generation.
- Baichuan2-7/13B-Chat: They are part of the Baichuan2 family of models, fine-tuned especially for chat and dialogue generation tasks. Both models are trained on a high-quality corpus of 2.6 trillion Tokens and have achieved excellent performance on several general-purpose domain Benchmarks in English, Chinese and multilingual.

For the above LLM, if publicly available, we use the official system prompts or instruction formats employed during training. The relevant problem generation prompt templates can be found in Appendix A.

4.3. Evaluation metrics

Recent studies showed that LLM exhibited performance in content evaluation that is highly consistent with human preferences, and could effectively simulate human judgment [61–63]. Based on this, we follow the approach of Balaguer et al. [64], using GPT-4 as the quality evaluator for model-generated content. In this study, GPT-4 is designated as a math problem quality review expert, scoring each generated problem on the following dimensions (on a scale of 10 points): whether it meets the generation objectives of user (including question type, knowledge points, and difficulty), the originality of the generated problem, readability, applicability, and usefulness. The average score across these dimensions is calculated as the evaluation metric. Additionally, we design detailed prompts (see Appendix A) to guide GPT-4 in conducting step-by-step analysis and assessing the solvability of the generated math problems.

In generation tasks, perplexity (PPL) is commonly used to measure the quality of LLM outputs, with lower values indicating higher output quality. Following previous studies [23,52], we use Llama3-70B-Instruct [60] to calculate the average perplexity of the problems generated by the model, which serves as one of the evaluation metrics.

Furthermore, we assessed the time each model required to complete the MMPG tasks in the evaluation data under different methods (measured in minutes), as an indicator of the efficiency of each approach.

5. Results and analysis

5.1. Main results

Table 2 shows the results of the proposed method and the baseline on the evaluation dataset for the MMPG task. Overall, the proposed method demonstrates outstanding performance in generating high-quality math problems with multiple objectives.

First, for open-source models, taking Baichuan2-7B-Chat as an example, compared to baseline without retrieval, the average score and solvability of the generated problems improve by 0.7 points and 7%, respectively. In comparison to the baseline with retrieval, these two metrics increase by 0.11 points and 1.5%, respectively. Similarly, the proposed method also demonstrates significant progress on the ChatGLM3-6B model. Compared to the baseline without retrieval, the average score increases by 0.44 points, and the solvability improves by 3%. Relative to the baseline with retrieval, the improvements are 0.29 points and 3%, respectively. For other open-source models (e.g., Llama-3-8B-Instruct, Baichuan2-13B-Chat), although the specific improvement magnitudes varied, the proposed method generally outperforms the baseline models in terms of problem quality and solvability, exhibiting positive gains in both average score and solvability rates.

On the closed-source model, using GPT-3.5-Turbo as the target LLM, the proposed method improves the average score of the generated problems by 0.34 points and the solvability by 1% compared to the baseline without retrieval. Compared to the baseline with retrieval, the proposed method improves the average score by 0.16 points and the solvability by 0.5%. On another closed-source model, Gemini-Pro, the average score and solvability improve by 0.35 points and 3.5%, respectively, compared to the baseline without retrieval, and by 0.11 points and 0.5%, respectively, compared to the baseline with retrieval. Notably, since closed-source models generally possess stronger performance and larger parameter sizes, the room for improvement is relatively limited. However, the proposed method still achieves certain improvements even on these high-performance models.

Secondly, as shown in Table 2, there is a significant upward trend in the PPL values of math problems generated using the full-retrieval method. For instance, the PPL values of Gemini-Pro and Baichuan2-7B-Chat using the full-retrieval method are 13.061 and 12.279, respectively, while through our method, the PPL values are reduced to 8.447 and 10.001, respectively. This trend is also evident in other models. The reason for this may be that the full-retrieval method provides external examples for all problem requests, which may introduce irrelevant or redundant information for some simpler tasks, thereby interfering with the generation process of LLM. In contrast, our method adaptively activates the retrieval strategy based on feedback from the mentor model, ensuring that external examples are provided to the generation model only when necessary. This approach effectively reduces the interference of irrelevant or noisy information, thereby lowering the perplexity of the generated content.

Lastly, the proposed retrieval framework demonstrates considerable competitiveness in terms of application efficiency. Although the proposed framework slightly exceeds the time consumption of non-retrieval method in all test models, it is significantly faster than the full-retrieval method. As shown in Table 2, using GPT-3.5-Turbo as the target LLM, the proposed method takes only 33 min to complete the global test, with an average generation time of 9.9 s per problem. In contrast, the full-retrieval generation method takes up to 100 min, with an average generation time of 30 s per problem. This advantage stems from the adaptive retrieval mechanism guided by the mentor model, which efficiently retrieves and integrates examples, reducing redundant steps and complexity in the generation process. While ensuring the quality of the generated content, this method greatly enhances generation efficiency, making it highly practical for real-world teaching scenarios.

5.2. Ablation study

To deeply analyze the contribution of each component in the proposed framework, we conduct an ablation study on GPT-3.5-Turbo (a closed-source model with well over 7B parameters) and ChatGLM3-6B (an open-source model with 6B parameters). Specifically, we remove one or more components from the framework, including the mentor model (MM), the objectives-base retriever (OR), and the similarity-base

Table 2
The overall experimental results of the six target LLMs on the evaluation dataset for the MMPG task, with bold numbers indicating the best performance among all methods.

LM	Parameters	Method	Score↑	Solvability↑	PPL↓	Time↓
GPT-3.5-Turbo	≫7B	w/o retrieval	7.35	97.5	**9.078**	**13**
		w/ retrieval	7.53	98	12.048	100
		Our	**7.69**	**98.5**	11.389	33
Gemini-Pro	≫7B	w/o retrieval	7.36	96	10.206	**18**
		w/ retrieval	7.60	99	13.061	118
		Our	**7.71**	**99.5**	**8.447**	34
ChatGLM3-6B	6B	w/o retrieval	6.67	92	10.195	**8**
		w/ retrieval	6.82	92	12.279	88
		Our	**7.11**	**95**	**10.001**	36
Llama-3-8B-Instruct	8B	w/o retrieval	7.19	93	**6.363**	**100**
		w/ retrieval	7.46	96	6.807	190
		Our	**7.57**	**97.5**	6.714	120
Baichuan2-7B-Chat	7B	w/o retrieval	6.68	90	**10.615**	9
		w/ retrieval	7.27	95.5	11.670	98
		Our	**7.38**	**97**	11.217	29
Baichuan2-13B-Chat	13B	w/o retrieval	7.11	97.5	14.002	**7**
		w/ retrieval	7.33	96	13.376	91
		Our	**7.46**	**98**	**13.006**	27

Table 3
We conduct an ablation study on the key components of our generation framework. "–" indicates the removal of the corresponding component from our framework, while MM, OR, and SR represent the Mentor Model, Objectives-base Retriever, and Similarity-base Retriever, respectively.

LM	Method	Score↑	Solvability↑	Time↓
GPT-3.5-Turbo	Our	7.69	98.5	33
	- MM	7.53	98	100
	- OR	7.53	98	25
	- SR	7.54	97.5	30
	- OR & MM	7.49	96.5	18
	- SR & MM	7.45	97.5	29
	- ALL	7.35	97.5	13
ChatGLM3-6B	Our	7.11	95	36
	- MM	6.82	92	88
	- OR	6.78	94	20
	- SR	6.81	93.5	23
	- OR & MM	6.63	92	10
	- SR & MM	6.71	94.5	20
	- ALL	6.67	92	8

retriever (SR) (fine-grained features scorer being removed along with the OR). We report performance metrics under various configurations, including generation quality score, solvability, and generation time. The results are shown in Table 3.

The results indicate that each component contributes significantly to the performance of the system. For instance, in the case of GPT-3.5-Turbo, removing the MM leads to a score drop from 7.69 to 7.53; removing the OR or SR causes the score to drop to 7.53 and 7.54, respectively. Removing multiple components further decreases the score to between 7.45 and 7.49. This indicates that each component plays a significant role in improving problem generation quality.

In terms of solvability, the removal of the MM in the ChatGLM3-6B model leads to a 3% drop in solvability, while removing the OR and SR results in a 1% and 1.5% decrease, respectively. This further highlights the importance of each component in ensuring the quality and solvability of the generated problems.

Finally, the time metric reflects the impact of each component on generation efficiency. The results show that the MM has the largest impact on generation efficiency in both models. OR and SR also influence efficiency to varying degrees. While removing these components reduces processing time, it also leads to a decline in overall performance. Therefore, properly configuring and optimizing these components is key to enhancing system performance and efficiency.

5.3. Quality distribution analysis of problems

To visually demonstrate the effectiveness of the proposed framework, we compare the score distribution of problems generated using the proposed framework with those generated directly by the language model based on the Score evaluation metric.

As shown in Figs. 3(c) and 3(d), the experimental results on ChatGLM3-6B indicate that among the problems generated by using our method (Fig. 3(c)), the proportion of 8-point problems is as high as 45.5% (91 questions), and 7-point problems 35.5% (71 questions), totaling 81%. Observations show that these problems perform well in satisfying the generation objectives in the user request for problems, with only a slight lack of innovativeness. On the contrary, in the direct generation method (Fig. 3(d)), the distribution of scores is more dispersed. Although problems scoring 8 (69 questions) and 7 (74 questions) are still predominant (accounting for 34.5% and 37%, respectively), the proportion of low-scoring problems increases significantly, e.g., the total proportion of problems with 1–5 points is 15%, while it is only 7.5% in our method. This indicates that the quality stability of directly generated problems needs improvement.

A similar trend is observed on GPT-3.5-Turbo. With our method (Fig. 3(a)), high-quality problems (scoring 7 or 8 points) dominate, with 52 and 143 problems, respectively, accounting for 97.5%. No problems score below 5 points. In contrast, problems generated directly by the language model (Fig. 3(b)) exhibit a more dispersed score distribution. While problems scoring 8 are the most (128 problems, accounting for 64%), there is still a notable proportion of lower-scoring problems (11 problems scoring 5 or below, accounting for 5.5%).

These results fully demonstrate the significant advantage of the proposed framework in improving the quality and stability of problem generation. Further experiments show that this framework exhibits similar applicability and effectiveness on other target language models.

5.4. Effects of mentor model

The mentor model, as a key component of this framework, is responsible for determining when to activate the retrieval mechanism, which is critical to both the quality and efficiency of the generated problems. In Section 3.3, we conduct fine-tuning of the Qwen1.5-14B-Chat model on a specific dataset to optimize its performance for this task.

To verify the training effect, we randomly divide a small test set (see Appendix C for details) from the supervised dataset in Section 4.1.2, and compare the performance of model before and after training using BLEU, ROUGE, and Exact Match (EM) metrics. The results are

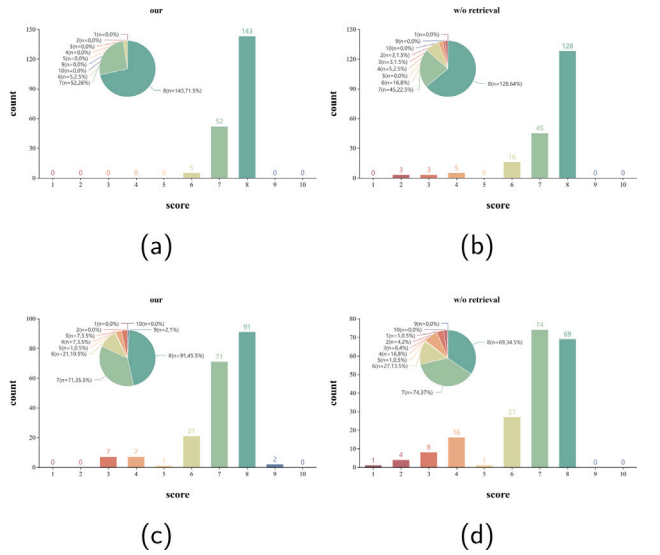

Fig. 3. (a) The score distribution of problems generated using our framework with GPT-3.5-Turbo as the target model; (b) The score distribution of problems generated directly by the GPT-3.5-Turbo model; (c) The score distribution of problems generated using our framework with ChatGLM3-6B as the target model; (d) The score distribution of problems generated directly by the ChatGLM3-6B model.

shown in Table 4. The data indicate that the trained mentor model outperforms the original Qwen1.5-14B-Chat model across all evaluation metrics. Specifically, the BLEU-4 score increases from 21.03 to 38.87. The ROUGE-1, ROUGE-2, and ROUGE-L scores improve from 49.28, 25.72, and 35.09 to 64.66, 48.09, and 53.10, respectively. The EM metric rises from 72% to 77%. The significant improvements in BLEU, ROUGE, and EM metrics reflect significant progress of the trained mentor model in determining whether the generated problems meet the generation objectives set by the user and in producing high-quality revision suggestions.

5.5. Similarity to the example problems

AMRAF adaptively activates the retrieval mechanism during the problem generation process. Upon activation, the retrieval module provides relevant example problems for the generation model to reference, and then regenerates a new problem. To verify that the regenerated problems by the generative model after referring to the example are not simple copies, we analyze the similarity between the generated problems and the examples. As shown in Table 5, the BERTScore F1 similarity between the generated problems by each model and the example problems is generally low, and the highest one, ChatGLM3-6B, is only 0.406. This indicates that, although the models referenced the example problems, the newly generated problems exhibit significant differences in both content and form, reflecting the ability of model to generate novel ideas and expressions when utilizing the examples.

In addition, we calculate the average quality score of the problems regenerated by the model after the retrieval mechanism is activated and provides with example problems (see Table 5), with all scores exceeding 7. This indicates that the overall quality of regenerated problems is high and can meet the multi-objective requirements of user. Combining the analysis of similarity and quality scores, we can conclude that introduction example problems effectively inspire the model to generate novel and high-quality math problems, rather than simply imitating the examples. This validates the rationality and effectiveness of the proposed method, helping to provide a broader range of practice problems and promoting comprehensive student learning and understanding.

Table 4
The performance changes of Qwen1.5-14B-Chat before and after training on the task of determining whether the problems generated by the model meet the user required objective. The EM metric represents the accuracy of consistency labels predicted by the model (i.e., whether labels predicted by the matches the ground truth consistency label in the test dataset).

LM	BLEU-4↑	ROUGE-1↑	ROUGE-2↑	ROUGE-L↑	EM↑
Qwen1.5-14B-Chat	21.03	49.28	25.72	35.09	72%
Mentor Model	38.87	64.66	48.09	53.10	77%

Table 5
The average score of math problems generated by the generative model after referencing example problems, as well as the similarity to the example problems. BF1 represents BERTScore F1.

LM	Score↑	BF1↓
GPT-3.5-Turbo	7.48	0.380
Gemini-Pro	7.50	0.401
ChatGLM3-6B	7.16	0.406
Llama-3-8B-Instruct	7.59	0.347
Baichuan2-7B-Chat	7.49	0.403
Baichuan2-13B-Chat	7.28	0.397

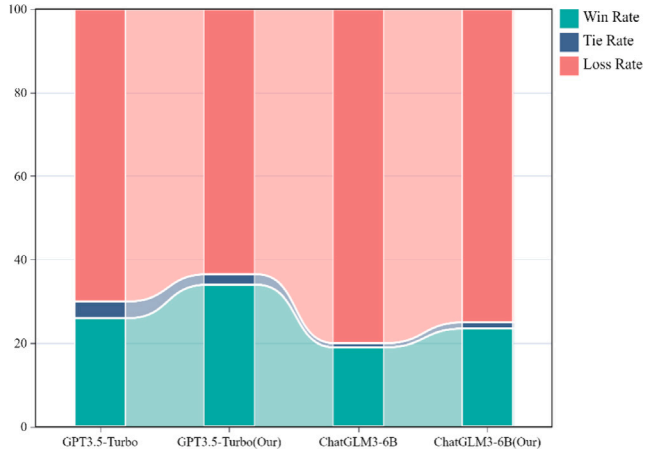

Fig. 4. The win rate, tie rate, and loss rate of GPT-3.5-Turbo and ChatGLM3-6B models against GPT-4, both before and after using our generation framework.

5.6. Battle against GPT-4

To more comprehensively evaluate the effectiveness of the MMPG framework proposed in this paper, we design an additional experiment: a competition between the target model and GPT-4 on the evaluation task. GPT-4o [65], which performs better in math tasks, is chosen as the judge to determine the quality of the problems generated by both models (see Appendix A for specific prompt templates). We select two representative models, GPT-3.5-Turbo and ChatGLM3-6B, and compare their performance before and after using the generation framework.

As shown in Fig. 4, GPT-3.5-Turbo win rate against GPT-4 is only 26% without using the framework, but it increases to 34% after applying the framework, indicating that the proposed framework significantly improves its ability to generate multi-objective math problems. Similarly, ChatGLM3-6B win rate increases from 19% to 23.5% after using the framework, showing a noticeable improvement.

Overall, the proposed framework effectively enhances the performance of the generation models in MMPG, allowing them to perform better when competing against stronger models.

5.7. Effect of number of examples

We analyze the impact of the number of examples on the performance of our framework. Specifically, we recall varying numbers of

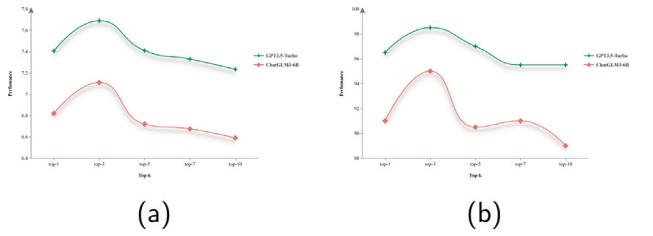

Fig. 5. (a) The impact of different numbers of examples on the quality of the problems generated by the model; (b) The impact on solvability.

example problems (from top-1 to top-10) and provid them to the target models (GPT-3.5-Turbo and ChatGLM3-6B) to evaluate the score and solvability of the generated problems.

Fig. 5(a) shows that as the number of examples increases from 1 to 3, the scores of both models rise significantly, reaching their peak (7.69 for GPT-3.5-Turbo and 7.11 for ChatGLM3-6B). However, when the number of examples exceeds 5, the scores decline, suggesting that too many examples introduce distracting information. Fig. 5(b) shows that solvability improves when the number of examples increases to 3 (GPT-3.5-Turbo from 96.5% to 98.5%, and ChatGLM3-6B from 91% to 95%), but further increases in examples lead to a decrease in solvability, likely due to information overload or overfitting.

In summary, a moderate number of examples (e.g., 3) helps to improve both generation quality and solvability, while too many examples may cause interference. We obtain similar results with other target models, providing valuable insights for selecting the optimal number of examples in similar tasks going forward.

5.8. Case study

The cases shown in Fig. 6 illustrate the performance of our framework when applied to GPT-3.5-Turbo (left), ChatGLM3-6B (center), and Baichuan2-7B-Chat (right) as the target models.

When generating multi-objective math problem, the initial problem generated by the model often deviate from the target attributes. For example, in the upper-left case of Fig. 6, the user requests the knowledge points of "sector area calculation, properties of tangents, and properties of isosceles triangles". However, the generated problem only cover "sector area calculation". Subsequently, the mentor model in the proposed framework effectively identifies and feeds back such errors, and proceeds to the example retrieval phase, retrieving three relevant examples from the problem corpus to serve as references for regenerating the problem. Eventually, the model adjusts and improves the problem by combining the mentor feedback and the examples, so that the generated problem are more consistent with the user request in terms of knowledge points.

5.9. Effect of similarity measures

To investigate the impact of different measures on retrieval performance in similarity-based retriever, we conduct comparative experiments using three common vector distance measuring methods: Cosine Similarity (CS), Euclidean Distance (ED), and Inner Product (IP). Table 6 presents the experimental results on GPT-3.5-Turbo and ChatGLM3-6B.

For GPT-3.5-Turbo, all three methods achieve competitive performance, with CS and ED showing particularly strong results. CS achieves the highest Score (7.69) and maintains robust Solvability (98.5%), while ED demonstrates slightly higher Solvability (99%) but a marginally lower Score (7.60). IP, while still effective, shows relatively lower performance in both metrics (Score: 7.37, Solvability: 96%). In terms of computational efficiency, the IP method is optimal and the other two methods are similar. For the ChatGLM3-6B model, having the same performance, CS consistently outperforms other methods, obtaining a Score of 7.11 and 95% solvability. Both ED (Score: 6.92, Solvability: 94%) and IP (Score: 6.79, Solvability: 93.5%) show noticeable degradation in performance. This suggests that the choice of similarity measure may have a more significant impact on smaller models.

Table 6
We compare the effects of using different similarity measures in similarity-based retriever.CS indicates the use of Cosine Similarity, ED indicates the use of Euclidean distance, and IP indicates the use of Inner Product.

LM	Method	Score↑	Solvability↑	Time↓
GPT-3.5-Turbo	CS	7.69	98.5	33
	ED	7.60	99	34
	IP	7.37	96	32
ChatGLM3-6B	CS	7.11	95	36
	ED	6.92	94	36
	IP	6.79	93.5	34

The superior performance of CS can be attributed to its focus on directional similarity rather than size, making it particularly suitable for semantic matching in math problem retrieval. Furthermore, CS demonstrates more stable performance across different model scales, suggesting it might be a more robust choice for various deployment scenarios. These findings support our choice of CS as the primary similarity measure in the proposed framework.

5.10. Mentor model computational cost analysis

In our framework, the mentor model serves as an assisted decision-making module designed to improve the accuracy and efficiency of multi-objective math problem generation. However, the introduction of the mentor model is also accompanied by additional computational overheads, especially in scenarios with large-scale generation tasks or highly concurrent requests. In order to fully understand its impact on computational resources, this section analyzes the main computational overheads introduced by the mentor model.

Training overhead: In order to improve the accuracy of the mentor model, we fine-tune it based on domain-specific data. This process requires additional computational resources, especially when the amount of data is large, and the overhead of the fine-tuning process increases significantly. For this reason, we use the LoRA technique for efficient fine-tuning, which significantly reduces the training cost. Compared with full fine-tuning, LoRA requires training only a small fraction of parameters (about 0.05%), which significantly reduces the computational and storage requirements. In addition, the moderate size of our training dataset (14,000 samples) allows the training process to be completed in less than 12 h on a single A100 GPU.

Reasoning overhead: The mentor model requires reasoning to determine whether a retrieval is triggered, a process that needs to be computed each time it is generated. In practice, the reasoning overhead of the mentor model is manageable, with an average additional reasoning time of only 2–3 s per question. This is an acceptable latency in practical application scenarios.

While the introduction of the mentor model improves the generation quality and control capability of the framework, it also imposes a certain computational overhead. To further improve the efficiency of the framework, future research could focus on optimizing the computational efficiency of the mentor model and exploring how to mitigate the computational overhead through model compression, knowledge distillation, or pruning techniques.

6. Conclusion

This paper conducts research on the technology and application of LLM in MPG. In response to the situation where the previous problem

Fig. 6. Case Study. Illustrative examples of the process of generating multi-objective math problems for GPT-3.5-Turbo (left), ChatGLM3-6B (middle) and Baichuan2-7B-Chat (right).

generation mode inadequately meets the requirements of personalized teaching, the MMPG task is first proposed, aiming to generate problem resources that meet multiple teaching objectives in a more controllable way. To address the challenges posed by this task regarding model controllability and knowledge consistency, we propose an adaptive multi-level retrieval augmentation framework for LLM to generate multi-objective math problems. This plug-and-play framework requires no parameters tuning of the target LLM. The construction of the mentor model enables adaptive activation of retrieval mechanism, offering effective contextual suggestions during the secondary generation process. The design of the multi-level retrieval mechanism facilitates the retrieval of more relevant and reliable example problems, significantly enhancing the MMPG performance of target LLM. Finally, a Chinese MMPG dataset containing 9000 samples is proposed, demonstrating the superiority of the proposed framework and providing a strong benchmark for future MMPG research.

CRediT authorship contribution statement

Jianwen Sun: Writing – original draft, Funding acquisition. **Wangzi Shi:** Writing – original draft, Investigation. **Xiaoxuan Shen:** Formal analysis, Data curation. **Shengyingjie Liu:** Software. **Luona Wei:** Writing – review & editing, Validation. **Qian Wan:** Writing – review & editing, Supervision, Funding acquisition.

Declaration of competing interest

The authors declare that they have no known competing financial interests or personal relationships that could have appeared to influence the work reported in this paper.

Acknowledgments

This work was financially supported by the National Key R&D Program of China (2023YFC3305704), National Natural Science Foundation of China (62437002, 62307015, 62293554), Hubei Provincial Natural Science Foundation of China (2024AFB169, 2023AFB295, 2023AFA020), China Postdoctoral Science Foundation, China (2023M741304), Fundamental Research Funds for the Central Universities of South-Central Minzu University, China (CZQ24006) and Knowledge Innovation Program of Wuhan-Shuguang Project, China (2023010201020390).

Appendix A. Prompt details

A.1. Prompts used in generation objective extract

In this section, we show the prompt template used to extract the corresponding generation objectives from user problem request, as shown in Fig. A.1.

A.2. Prompts used in data consistency check

In Fig. A.2, we show the prompt template used by GPT-4 to perform consistency label checks for the collected MMPG.

A.3. Prompts used in baselines without retrievals

In Fig. A.3, we show the prompt template used by the baseline model, without retrieval, for generating multi-objective math problems.

Orginal version-zh
你是一个自然语言处理模型，用于完成 NLP 基础任务。你现在需要完成一个属性抽取任务，定义的属性有：{properties question}，下面你将收到一个句子，请从请从给定的句子中抽取出规定的属性值，不要自行总结，如果句子中不存在该属性请将值置为 NULL。

以下是如何完成任务的示例：
{In context examples}

现在请提取：
{setence}

Translated version-en
You are a natural language processing model, used to complete NLP basic tasks. You now need to complete an attribute extraction task. The defined attributes are: {properties question}. Below you will receive a sentence. Please extract the specified attribute value from the given sentence. Do not summarize it yourself. If the attribute does not exist in the sentence, please set the value to NULL.

The following is an example of how to complete the task:
{In context examples}

Now please extract:
{setence}

Fig. A.1. Prompt template used for generation objective extraction. The {In context examples} represents an extraction example used for model learning, and {sentence} represents the user problem request.

Orginal version-zh
CONTEXT（上下文）
你是一名数学题审核老师，负责审核 AI 助手生成的数学题是否符合用户要求。用户通过 AI 助手生成了一道数学题目，想要判断这道数学题是否符合自己的要求。
用户要求如下:
{question}
AI 助手的回答如下:
{answer}

OBJECTIVE（目标）
请你严格的判断 AI 助手生成的数学题目是否满足了用户提出的题型、知识点、困难度这三个条件，并且你还需判断 AI 助手生成的题目是否是可解的。如果 AI 助手生成的题目完全满足用户要求的三个条件且可解，那么请你输出"[[YES]]"即可,如果有任何一点不满足请你输出"[[NO]]"并且你需要指出 AI 助手的错误然后针对每条错误给出解决方案。

STYLE（风格）
按照专业的数学题审核老师的风格。

Translated version-en
CONTEXT
You are a math problem reviewer, responsible for reviewing whether the math problems generated by the AI assistant meet the user's requirements. The user generates a math problem through the AI assistant and wants to determine whether the math problem meets his requirements.
The user's requirements are as follows:
{question}
The AI assistant's answer is as follows:
{answer}

OBJECTIVE
Please strictly judge whether the math problem generated by the AI assistant meets the three conditions of question type, knowledge points, and difficulty proposed by the user, and you also need to judge whether the problem generated by the AI assistant is solvable. If the problem generated by the AI assistant fully meets the three conditions required by the user and is solvable, please output "[[YES]]". If any of them are not satisfied, please output "[[NO]]" and you need to point out the mistakes of the AI assistant and provide solutions for each mistake.

STYLE
Follow the style of a professional math problem reviewer.

Fig. A.2. Prompt template for GPT-4 assisted consistency label checking. The {question} represents the user problem request containing multiple objectives generated by our model, and the {answer} represents the corresponding problem.

Orginal version-zh
你是一名 AI 数学老师，从事数学教育 30 年，精通设计各种数学题,请你按照用户要求出题并确保所生成的题目是可解的。其中题目困难度是对知识点和题型及其复杂程度、题目所需的计算量、题目的表述连贯性和逻辑性以及学生普遍学习水平综合考虑后的量化表示,困难度越大，题目越难。

现在请为用户出题,但是所生成的数学题应该保证是完全符合用户要求的、上下文清晰的、逻辑通顺的、可解的、新颖的、对该阶段学生的学习是有帮助的，用户具体要求如下:
{question}

Translated version-en
You are an AI math teacher who has been engaged in math education for 30 years and is proficient in designing various math problems. Please set problems according to user requirements and ensure that the generated problems are solvable. The difficulty of the problems is a quantitative expression of the knowledge points and question types and their complexity, the amount of calculation required for the problem, the coherence and logic of the problem's expression, and the general learning level of the students. The greater the difficulty, the harder the problem.

Now please give problems to users, but the generated math problems should be guaranteed to fully meet the user's requirements, have clear context, coherent logic, solvable, novel, and helpful for students at this stage of learning. The user's specific requirements are as follows:
{question}

Fig. A.3. Prompt template used by the baseline model for generating multi-objective math problems without retrieval. The {question} represents the user problem request containing multiple objectives.

A.4. Prompts used in our method

The prompt template used when generating preliminary math problems using our generative framework is the same as the one used when generating math problems using the baseline model without retrieval. When the generation model needs to reference external examples to regenerate problems, the prompt template shown in Fig. A.4 is applied.

A.5. Prompts used in baselines with retrievals

In Fig. A.5, we show the prompt template used by the baseline model with retrieval for generating multi-objective math problems.

A.6. Prompts used in evaluating performance

We use GPT-4 to evaluate the overall quality of the final generated problems, and Fig. A.6 shows the template content we use to prompt GPT-4.

A.7. Prompts used in determining the winner

In Section 5.6, we designate GPT-4o as the judge to determine which of the problems generated by the two models is superior and better meets the user problem request. The prompt template used to guide GPT-4o in completing this task is shown in Fig. A.7.

Appendix B. Dataset details

B.1. Is retrieval always helpful

Through experiments, we find that in the MMPG task, the retrieved contextual examples do not always improve the generation quality of the models. In Fig. B.1, we evaluate the performance changes of GPT-3.5-Turbo (a) and ChatGLM3-6B (b) on the evaluation dataset when generating problems without retrieval and with full retrieval (i.e., the change in the Score evaluation metric of the generated problems).

The results show that retrieval does not always provide benefits for MMPG. Specifically, retrieval improves the performance of the generation model in only 35% or fewer instances, while for over 40%

Orginal version-zh

经过人类专家的评估后认为你刚刚所生成的数学题目并不能很好的满足用户出题请求,下面有三个可能与用户要求相符合的例题以及专家指出的当前生成的题目存在的问题与修改建议：
例题如下：
例题 1:{example 1}
例题 2:{example 2}
例题 3:{example 3}
存在的问题与修改建议如下:
{explation}

现在请你参考例题以及专家指出的问题和建议重新生成一道数学题目,你必须确保新题目对旧题目存在的问题进行了认真的修改并且合理的参考了例题的内容,同时生成的数学题应该保证是完全符合用户要求的、上下文清晰的、逻辑通顺的、可解的、新颖的、对学生的学习是有帮助的。

Translated version-en

After evaluation by human experts, it is believed that the math problem you just generated does not meet the user's question request very well. The following are three examples that may meet the user's requirements and the shortcomings and modification suggestions of the currently generated problems pointed out by experts:
The examples are as follows:
Example 1: {example 1}
Example 2: {example 2}
Example 3: {example 3}
The existing shortcomings and modification suggestions are as follows:
{explation}.

Now please refer to the example and the shortcomings and suggestions pointed out by the experts to regenerate a math problem. You must ensure that the new problem has carefully modified the shortcomings of the old problem and reasonably refers to the content of the example. At the same time, the generated math problem should be guaranteed to fully meet the user's requirements, have clear context, coherent logic, be solvable, novel, and helpful for students' learning.

Fig. A.4. Prompt template used for regenerating problems with reference to examples.{example 1}, {example 2}, and {example 3} represent three math example problems retrieved by the retriever that are potentially relevant to the user problem request, and {explanation} represents the revision suggestions from the mentor model.

Orginal version-zh

你是一名 AI 数学老师，从事数学教育 30 年，精通设计各种数学题,请你按照用户要求出题并确保所生成的题目是可解的。其中题目困难度是对知识点和题型及其复杂程度、题目所需的计算量、题目的表述连贯性和逻辑性以及学生普遍学习水平综合考虑后的量化表示,困难度越大，题目越难。

以下有三个可能满足用户要求的数学题目作为例子参考:
例题一：{example 1}
例题二：{example 2}
例题三：{example 3}

现在请参考所给例子的出题风格为用户出题,但是所出题目不能与例题完全相同!同时生成的数学题应该保证是完全符合用户要求的、上下文清晰的、逻辑通顺的、可解的、新颖的、对该阶段学生的学习是有帮助的，具体要求如下:
{question}

Translated version-en

You are an AI math teacher who has been engaged in math education for 30 years and is proficient in designing various math problems. Please set problems according to user requirements and ensure that the generated problems are solvable. The difficulty of the problems is a quantitative expression of the knowledge points and question types and their complexity, the amount of calculation required for the problem, the coherence and logic of the problem's expression, and the general learning level of the students. The greater the difficulty, the harder the problem.

Here are three math problems that may meet the user's requirements as examples for reference:
Example 1: {example_1}
Example 2: {example_2}
Example 3: {example_3}

Now please refer to the problem-setting style of the given example to set problems for the user, but the problems cannot be exactly the same as the example problems! At the same time, the generated math problems should be guaranteed to fully meet the user's requirements, have clear context, coherent logic, solvable, novel, and helpful for students at this stage of learning. The specific requirements are as follows:
{question}

Fig. A.5. Prompt template used by the baseline model with retrieval for generating multi-objective math problems. The {question} represents the user problem request containing multiple objectives, while {example 1}, {example 2}, and {example 3} represent three math example problems retrieved by the retriever that are potentially relevant to the user problem request.

or more of the instances, retrieval has no impact on performance. Furthermore, in the remaining 25% of cases, retrieval actually degrades performance. This observed trend applies to both small (6B∼13B) and large (≫7B) generation models. In summary, these findings highlight the suboptimal nature of always using retrieval, which motivate us to propose adaptive retrieval.

B.2. Display and analysis based on knowledge points embedding

From Fig. B.2, we can observe that most problems related to the same knowledge point are clustered together in the embedding space, forming distinct groups. This indicates that the OpenAI embedding model we used effectively captures the semantic similarity of problems at the knowledge point level, indirectly validating the overall high quality of our knowledge point annotations.

Specifically, the clustering effect of the problems related to the knowledge points "Tea Brewing Problem" and "Pattern of Number Changes", locate on the left side, is particularly prominent. They form compact and distinct clusters in the embedding space with almost no overlap with problems from other knowledge points. This indicates that the problems for these two knowledge points are highly similar semantically, while being significantly distinct from other knowledge points. This reflects the accuracy of our annotation for these problems, with strong consistency in the annotation results.

Looking at the top right corner, we can see that the problems corresponding to knowledge points such as "Chicken and Rabbit in the Same Cage Problem", "Recognizing Time", and "Abstracting a Linear Equation from a Real-World Problem" show some spatial overlap. However, the clustering boundaries are still relatively clear overall. This overlap may be due to some similarities in the way problems are formulated across these knowledge points, resulting in closer distances in the embedding space. Nevertheless, we are still able to differentiate these knowledge points fairly well.

However, a few points in the figure are located far from the main cluster of their corresponding knowledge points, becoming outliers. This may be due to two possible reasons: first, these problems differ significantly from other problems of the same knowledge point in terms of phrasing, difficulty, or other aspects; second, there may be errors or inconsistencies in our annotation process for these problems. For the former, we can further analyze the uniqueness of these problems to determine whether the granularity of the knowledge point classification needs adjustment. For the latter, we need to re-review and correct the knowledge point annotations of the relevant problems to improve the accuracy and consistency of the annotations.

Appendix C. Finetuning details

C.1. Implementation details

For the fine-tuning of the mentor model, we use the AdamW optimizer for training, with a gradient clipping of 1.0, a cosine learning rate scheduling strategy with warm-up, a total of 3 epochs of training, a maximum learning rate set to 2e−5, a warm-up rate of 3%, and a maximum token length set to 2048. model training is carried out on a single A100 GPU, with mixed-precision training enabled.

Orginal version-zh

CONTEXT（上下文）
你是一名数学题审核老师，负责审核 AI 助手生成的数学题是否符合用户要求。请以公正的裁判身份评估下方展示的用户请求所得到的 AI 助手的回答质量。在你评分过程中不考虑 AI 助手有没有提供解题步骤,同时用户要求中题目困难度是对知识点和题型及其复杂程度、题目所需的计算量、题目的表述连贯性和逻辑性以及学生普遍学习水平综合考虑后的量化表示,困难度越大，题目越难。
用户要求如下:
{question}
AI 助手的回答如下:
{answer}

OBJECTIVE（目标）
请你严格的判断 AI 助手生成的数学题目是否满足了用户要求的题型、知识点、困难度这三个条件，在此基础上你还需判断 AI 助手生成的题目的创新性、可读性、适用性和帮助性。综合考虑以上各个指标后请你按照以下格式对题目质量进行严格评分，评分范围为 1 到 10:"[[rating]]"，例如:"评分:[[5]]",并且你需要对你的该评分做出解释。另外你需要单独考虑 AI 助手生成的题目是否可解，如果该题目是可解的那么请你输出"[[Y]]",如果该题目是不可解的那么请你输出"[[N]]".

STYLE（风格）
按照专业的数学题审核老师的风格，必须客观公正。

Translated version-en

CONTEXT
You are a math problem reviewer, responsible for reviewing whether the math problems generated by the AI assistant meet the user's requirements. Please evaluate the quality of the AI assistant's answers to the user's requirements shown below as an impartial judge. In your scoring process, do not consider whether the AI assistant provides the steps to solve the problem. At the same time, the difficulty of the question in the user's requirements is a quantitative expression of the knowledge points and question types and their complexity, the amount of calculation required for the problem, the coherence and logic of the problem's expression, and the general learning level of the students. The greater the difficulty, the harder the problem.
The user's requirements are as follows:
{question}
The answer of the AI assistant is as follows:
{answer}

OBJECTIVE
Please strictly judge whether the math problems generated by the AI assistant meet the three conditions of question type, knowledge points, and difficulty required by the user. On this basis, you also need to judge the innovation, readability, applicability, and helpfulness of the problems generated by the AI assistant. After comprehensively considering the above indicators, please strictly score the quality of the problems in the following format, with a score range of 1 to 10: "[[rating]]", for example: "Rating: [[5]]", and you need to explain your score. In addition, you need to consider separately whether the problems generated by the AI assistant are solvable. If the problem is solvable, please input "[[Y]]", if the problem is not solvable, please input "[[N]]".

STYLE
The style of the professional math problem reviewer must be objective and fair.

Fig. A.6. Prompt template for using GPT-4 to evaluate the quality of model generated multi-objective math problems.{question} represents the user problem request containing multiple objectives, and {answer} represents the final problem generated by the model.

Orginal version-zh

CONTEXT（上下文）
你是一名数学题审核老师，负责判断各种数学题的质量。请作为一名公正的法官，评估两名人工智能助手对下面展示的用户问题所做回答的质量。在你评估过程中不考虑 AI 助手有没有提供解题步骤，同时用户要求中题目困难度是对知识点和题型及其复杂程度、题目所需的计算量、题目的表述连贯性和逻辑性以及学生普遍学习水平综合考虑后的量化表示，困难度越大，题目越难。
用户要求如下:
{question}
助手 A 的回答如下:
{answer a}
助手 B 的回答如下:
{answer b}

OBJECTIVE（目标）
你应该选择遵循用户要求并更好地回答用户问题的助手。您的评估应该考虑他们生成的数学题目是否满足了用户提出的题型、知识点、困难度这三个条件，在此基础上你还需判断他们生成的题目的创新性、可读性、适用性和帮助性等因素。通过比较两个回答并提供简短解释来开始评估。避免任何立场偏见，并确保回答的顺序不会影响您的决定。不要让回答的长度影响您的评估。不要对某些助手的名字存在偏见,尽可能客观。在提供解释后，严格按照以下格式输出您的最终裁决:如果助手 A 的回答更好请输出[[A]]，如果助手 B 的回答更好请输出[[B]]，以及输出[[C]]表示平局。

STYLE（风格）
按照专业法官的分格，必须客观公正。

Translated version-en

CONTEXT
You are a math problem reviewer, responsible for judging the quality of various math problems. Please be a fair judge and evaluate the quality of the answers given by the two AI assistants to the user questions shown below. In your evaluation process, do not consider whether the AI assistant provides the steps to solve the problem. At the same time, the difficulty of the question in the user's requirements is a quantitative expression of the knowledge points and question types and their complexity, the amount of calculation required for the problem, the coherence and logic of the problem's expression, and the general learning level of the students. The greater the difficulty, the harder the problem.
The user's requirements are as follows:
{question}
Assistant A's answer is as follows:
{answer_a}
Assistant B's answer is as follows:
{answer_b}

OBJECTIVE
You should choose the assistant that follows the user's requirements and answers the user's questions better. Your evaluation should consider whether the math problems they generate meet the three conditions of question type, knowledge points, and difficulty proposed by the user. On this basis, you also need to judge the innovation, readability, applicability, and helpfulness of the problems they generate. Start the evaluation by comparing the two answers and providing a brief explanation. Avoid any bias and make sure that the order of the answers does not affect your decision. Do not let the length of the answer affect your evaluation. Do not be biased against the names of certain assistants, and be as objective as possible. After providing an explanation, output your final verdict strictly in the following format: if assistant A's answer is better, please output [[A]], if assistant B's answer is better, please output [[B]], and output [[C]] for a tie.

STYLE
In the style of a professional judge, you must be objective and fair.

Fig. A.7. Prompt template for GPT-4o as a judge to determine the winner between two models.{question} represents the user problem request, {answer a} represents the problem generated either by the baseline model without retrieval or by our generation framework, and {answer b} represents the problem generated by the GPT-4 model.

C.2. Data structure

We used 16,000 question-answer pairs in total (detailed in Section 4.1.2), with a 7–1 split for training, and testing. Data is provided in the instruction format shown in Figs. C.1, C.2 shows a specific example of training data.

Appendix D. Failure case study

To better understand the limitations of the proposed framework and to provide ideas for future improvement, we analyze in depth the common failures of the proposed framework in the generation process, focus on the limitations of the mentor model, the retrieval module, and the generation model itself, and propose possible directions for improvement.

First, the mentor model may give wrong retrieval tags when facing complex constraint combinations. For example, in Fig. D.1 (left), in the request “Generate a multiple-choice problem involving the knowledge points isosceles trapezoid, approximate numbers and significant digits (Difficulty: 0.4)”, the model incorrectly predicts “NO”, which results in the generated problem involving only isosceles trapezoid and fails to effectively incorporate the knowledge point of approximate numbers and significant digits. This suggests that the ability of model to make judgments when dealing with combinations of multiple knowledge points still needs to be strengthened.

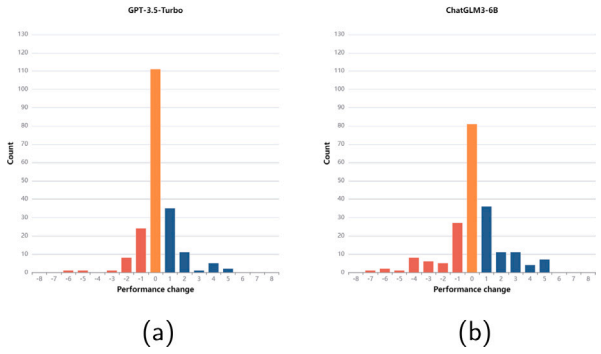

Fig. B.1. The performance gains achieved by full retrieval compared to not using retrieval are illustrated. The X-axis represents the change in Score (i.e., performance gain), where 1 indicates a 1-point increase in the Score of the generated problems after using retrieval, and −1 indicates a 1-point decrease in the Score. Other values follow this pattern. The Y-axis represents the number of instances corresponding to the current score change. (a) GPT-3.5-Turbo. (b) ChatGLM3-6B.

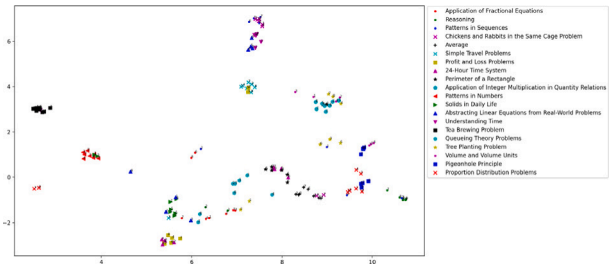

Fig. B.2. Visualization of different knowledge points embedded.

Instruction:{user instruction}
Output:{model response}
System:{system prompt}

Fig. C.1. Data follow alpaca format.

Orginal version-zh

Instruction:

CONTEXT（上下文）

用户通过 AI 助手生成了一道数学题目,想要判断这道数学题是否符合自己的要求。

用户要求如下:请帮我出一道题型为:解答题,涉及知识点:[乘法的应用]且困难度为:0.8 的数学题，不需要给出解答过程。

AI 助手的回答如下:{'题型': '解答题', '知识点': '[乘法的应用]', '困难度': '0.5', '题目': '学校买了 4 箱水果，每箱 25 千克，每千克水果 2 元钱。问学校买水果一共用去多少元钱？'}

OBJECTIVE（目标）

请你严格的判断 AI 助手生成的数学题目是否满足了用户提出的题型、知识点、困难度这三个条件，并且你还需判断 AI 助手生成的题目是否是可解的。如果 AI 助手生成的题目完全满足用户要求的三个条件且可解，那么请你输出[[NO]]即可,如果有任何一点不满足请你输出[[YES]]并且你需要指出 AI 助手的错误然后针对每条错误给出解决方案。

STYLE（风格）

按照专业的数学题审核老师的风格。

Output:

AI 助手生成的题目满足了用户的要求，题型为解答题，知识点为乘法的应用，困难度为 0.5，且题目是可解的。因此，我的判断是：[[NO]]。

System:

你是一名数学题审核老师，负责审核 AI 助手生成的数学题是否符合用户要求。

Translated version-en

Instruction:

CONTEXT

The user generates a math problem through the AI assistant, and wants to judge whether this math problem meets his requirements.

The user's request is as follows: Please help me to generate a math problem with the type of question: problem-solving, the knowledge point involved: application of multiplication and the difficulty level: 0.8, without giving the solution process.

The AI assistant's answer is as follows:{'Question Type': 'problem-solving', 'Knowledge Point': '[application of multiplication]', 'Difficulty': '0.5', 'Problem': 'The school bought 4 boxes of fruits, each box is 25kg, and each box is 2 yuan per kilogram of fruit. Ask how many dollars the school spent on fruit.'}

OBJECTIVE

Please strictly judge whether the math problem generated by the AI assistant satisfies the three conditions of question type, knowledge point and difficulty proposed by the user, and you also need to judge whether the problem generated by the AI assistant is solvable. If the problem generated by the AI assistant fully satisfies the three conditions requested by the user and is solvable, then please output [[NO]], if there is any point that does not satisfy the requirements, please output [[YES]] and you need to point out the errors of the AI assistant and then give a solution for each error.

STYLE

Follow the style of a professional math problem reviewer.

Output:

The problem generated by the AI assistant fulfils the user's requirements, the question type is an problem-solving problem, the knowledge point is the application of multiplication, the difficulty level is 0.5 and the problem is solvable. Therefore, my judgement is: [[NO]].

System:

You are a math problem reviewer, responsible for reviewing the math problem generated by the AI Assistant for user compliance.

Fig. C.2. A sample data.

Second, the retrieval module mainly faces the challenge of corpus coverage limitation. As shown in Fig. D.1 (center), in the case of "Generate a proof problem involving the knowledge points compound statistical tables and the volume of a cone", due to the lack of similar combinatorial samples in the corpus, the retrieval results only contain problems on a single knowledge point, which affects the quality of the final generation.

Finally, as shown in Fig. D.1 (right), even when the mentor guidance is correct and relevant samples are retrieved, the generative model may still perform poorly in terms of mathematical consistency, difficulty control, and knowledge integration. This problem mainly stems from the capacity limitations of the generative model, especially its inadequacy in handling complex mathematical logic and multidimensional constraints.

Based on the above analysis, we propose two future directions for improvement: (1) enhancing the performance of the mentor model by expanding the training data with complex constraint combinations, and (2) developing specialized math content embedding methods and hierarchical retrieval strategies.

Data availability

Data will be made available on request.

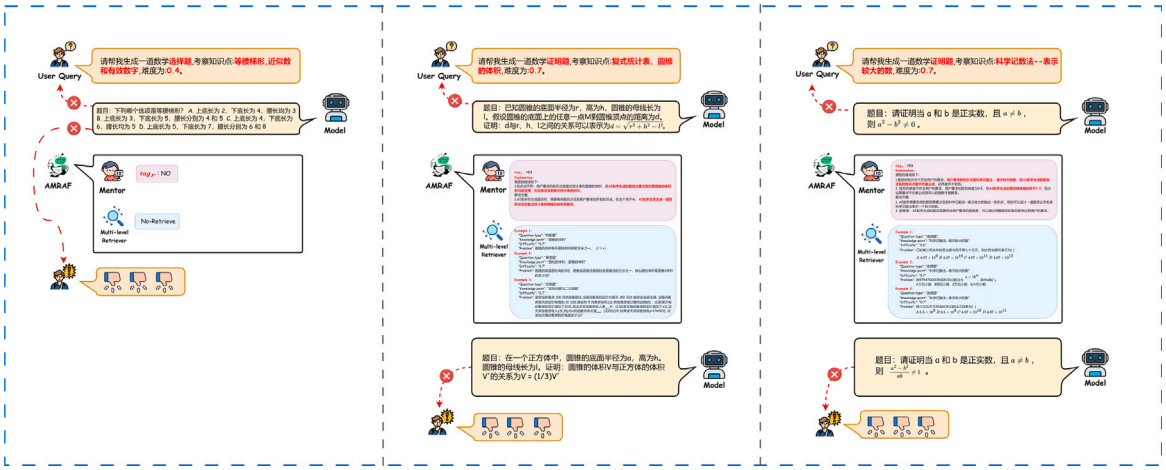

Fig. D.1. Failure case study.

References

[1] Rik Koncel-Kedziorski, Ioannis Konstas, Luke Zettlemoyer, Hannaneh Hajishirzi, A theme-rewriting approach for generating algebra word problems, 2016, arXiv preprint arXiv:1610.06210.

[2] Oleksandr Polozov, Eleanor O'Rourke, Adam M Smith, Luke Zettlemoyer, Sumit Gulwani, Zoran Popović, Personalized mathematical word problem generation, in: Twenty-Fourth International Joint Conference on Artificial Intelligence, 2015.

[3] Sannyuya Liu, Jintian Feng, Zongkai Yang, Yawei Luo, Qian Wan, Xiaoxuan Shen, Jianwen Sun, Comet:" cone of experience" enhanced large multimodal model for mathematical problem generation, 2024, arXiv preprint arXiv:2407.11315.

[4] Shaojie Ma, Yawei Luo, Yi Yang, Personas-based student grouping using reinforcement learning and linear programming, Knowl.-Based Syst. 281 (2023) 111071.

[5] Ghader Kurdi, Jared Leo, Bijan Parsia, Uli Sattler, Salam Al-Emari, A systematic review of automatic question generation for educational purposes, Int. J. Artif. Intell. Educ. 30 (2020) 121–204.

[6] Nabila Ahmed Khodeir, Hanan Elazhary, Nayer Wanas, Generating story problems via controlled parameters in a web-based intelligent tutoring system, Int. J. Inf. Learn. Technol. 35 (3) (2018) 199–216.

[7] Ke Wang, Zhendong Su, Dimensionally guided synthesis of mathematical word problems, in: IJCAI, 2016, pp. 2661–2668.

[8] Andinet Assefa Bekele, Automatic generation of amharic math word problem and equation, J. Comput. Commun. 8 (8) (2020) 59–77.

[9] Yawei Luo, Ping Liu, Yi Yang, Kill two birds with one stone: Domain generalization for semantic segmentation via network pruning, Int. J. Comput. Vis. (2024) 1–18.

[10] Qingyu Zhou, Danqing Huang, Towards generating math word problems from equations and topics, in: Proceedings of the 12th International Conference on Natural Language Generation, 2019, pp. 494–503.

[11] Vijini Liyanage, Surangika Ranathunga, Multi-lingual mathematical word problem generation using long short term memory networks with enhanced input features, in: Proceedings of the Twelfth Language Resources and Evaluation Conference, 2020, pp. 4709–4716.

[12] Qinzhuo Wu, Qi Zhang, Xuanjing Huang, Automatic math word problem generation with topic-expression co-attention mechanism and reinforcement learning, IEEE/ACM Trans. Audio Speech Lang. Process. 30 (2022) 1061–1072.

[13] Yawei Luo, Liang Zheng, Tao Guan, Junqing Yu, Yi Yang, Taking a closer look at domain shift: Category-level adversaries for semantics consistent domain adaptation, in: Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition, 2019, pp. 2507–2516.

[14] Qika Lin, Rui Mao, Jun Liu, Fangzhi Xu, Erik Cambria, Fusing topology contexts and logical rules in language models for knowledge graph completion, Inf. Fusion 90 (2023) 253–264.

[15] Hui Huang, Bing Xu, Xinnian Liang, Kehai Chen, Muyun Yang, Tiejun Zhao, Conghui Zhu, Multi-view fusion for instruction mining of large language model, Inf. Fusion 110 (2024) 102480.

[16] Yawei Luo, Yi Yang, Large language model and domain-specific model collaboration for smart education, Front. Inf. Technol. Electron. Eng. 25 (3) (2024) 333–341.

[17] Sandra Williams, Generating mathematical word problems, in: 2011 AAAI Fall Symposium Series, 2011.

[18] DeKita G. Moon-Rembert, Juan E. Gilbert, Illmatics: A web-based math word problem generator for students' distal and proximal interests, in: E-Learn: World Conference on E-Learning in Corporate, Government, Healthcare, and Higher Education, Association for the Advancement of Computing in Education (AACE), 2019, pp. 842–848.

[19] Paul Deane, Kathleen Sheehan, Automatic item generation via frame semantics: Natural language generation of math word problems, 2003.

[20] Tianyang Cao, Shuang Zeng, Songge Zhao, Mairgup Mansur, Baobao Chang, Generating math word problems from equations with topic consistency maintaining and commonsense enforcement, in: Artificial Neural Networks and Machine Learning–ICANN 2021: 30th International Conference on Artificial Neural Networks, Bratislava, Slovakia, September 14–17, 2021, Proceedings, Part III 30, Springer, 2021, pp. 66–79.

[21] Iddo Drori, Sarah Zhang, Reece Shuttleworth, Leonard Tang, Albert Lu, Elizabeth Ke, Kevin Liu, Linda Chen, Sunny Tran, Newman Cheng, et al., A neural network solves, explains, and generates university math problems by program synthesis and few-shot learning at human level, Proc. Natl. Acad. Sci. 119 (32) (2022) e2123433119.

[22] Mingyu Zong, Bhaskar Krishnamachari, Solving math word problems concerning systems of equations with gpt-3, in: Proceedings of the AAAI Conference on Artificial Intelligence, Vol. 37, 2023, pp. 15972–15979.

[23] Bryan R. Christ, Jonathan Kropko, Thomas Hartvigsen, MATHWELL: Generating educational math word problems at scale, 2024, arXiv preprint arXiv:2402.15861.

[24] Zhengyang Tang, Xingxing Zhang, Benyou Wan, Furu Wei, Mathscale: Scaling instruction tuning for mathematical reasoning, 2024, arXiv preprint arXiv:2403.02884.

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

[26] Patrick Lewis, Ethan Perez, Aleksandra Piktus, Fabio Petroni, Vladimir Karpukhin, Naman Goyal, Heinrich Küttler, Mike Lewis, Wen-tau Yih, Tim Rocktäschel, et al., Retrieval-augmented generation for knowledge-intensive nlp tasks, Adv. Neural Inf. Process. Syst. 33 (2020) 9459–9474.

[27] Kelvin Guu, Kenton Lee, Zora Tung, Panupong Pasupat, Mingwei Chang, Retrieval augmented language model pre-training, in: International Conference on Machine Learning, PMLR, 2020, pp. 3929–3938.

[28] Shuai Lu, Nan Duan, Hojae Han, Daya Guo, Seung-won Hwang, Alexey Svyatkovskiy, Reacc: A retrieval-augmented code completion framework, 2022, arXiv preprint arXiv:2203.07722.

[29] Noor Nashid, Mifta Sintaha, Ali Mesbah, Retrieval-based prompt selection for code-related few-shot learning, in: 2023 IEEE/ACM 45th International Conference on Software Engineering, ICSE, IEEE, 2023, pp. 2450–2462.

[30] Shuyan Zhou, Uri Alon, Frank F Xu, Zhiruo Wang, Zhengbao Jiang, Graham Neubig, Docprompting: Generating code by retrieving the docs, 2022, arXiv preprint arXiv:2207.05987.

[31] Jinheon Baek, Alham Fikri Aji, Amir Saffari, Knowledge-augmented language model prompting for zero-shot knowledge graph question answering, 2023, arXiv preprint arXiv:2306.04136.

[32] Shamane Siriwardhana, Rivindu Weerasekera, Elliott Wen, Tharindu Kaluarachchi, Rajib Rana, Suranga Nanayakkara, Improving the domain adaptation of retrieval augmented generation (RAG) models for open domain question answering, Trans. Assoc. Comput. Linguist. 11 (2023) 1–17.

[33] Zhihua Wen, Zhiliang Tian, Wei Wu, Yuxin Yang, Yanqi Shi, Zhen Huang, Dongsheng Li, Grove: a retrieval-augmented complex story generation framework with a forest of evidence, 2023, arXiv preprint arXiv:2310.05388.

[34] Zixuan Ke, Haowei Lin, Yijia Shao, Hu Xu, Lei Shu, Bing Liu, Continual training of language models for few-shot learning, 2022, arXiv preprint arXiv:2210.05549.

[35] Akari Asai, Zeqiu Wu, Yizhong Wang, Avirup Sil, Hannaneh Hajishirzi, Self-rag: Learning to retrieve, generate, and critique through self-reflection, 2023, arXiv preprint arXiv:2310.11511.

[36] Shangbin Feng, Weijia Shi, Yuyang Bai, Vidhisha Balachandran, Tianxing He, Yulia Tsvetkov, Knowledge card: Filling LLMs' knowledge gaps with plug-in specialized language models, 2023, arXiv preprint arXiv:2305.09955.
[37] Ruiyang Ren, Yuhao Wang, Yingqi Qu, Wayne Xin Zhao, Jing Liu, Hao Tian, Hua Wu, Ji-Rong Wen, Haifeng Wang, Investigating the factual knowledge boundary of large language models with retrieval augmentation, 2023, arXiv preprint arXiv:2307.11019.
[38] Saurav Kadavath, Tom Conerly, Amanda Askell, Tom Henighan, Dawn Drain, Ethan Perez, Nicholas Schiefer, Zac Hatfield-Dodds, Nova DasSarma, Eli Tran-Johnson, et al., Language models (mostly) know what they know, 2022, arXiv preprint arXiv:2207.05221.
[39] Zhangyue Yin, Qiushi Sun, Qipeng Guo, Jiawen Wu, Xipeng Qiu, Xuanjing Huang, Do large language models know what they don't know? 2023, arXiv preprint arXiv:2305.18153.
[40] Alex Mallen, Akari Asai, Victor Zhong, Rajarshi Das, Daniel Khashabi, Hannaneh Hajishirzi, When not to trust language models: Investigating effectiveness of parametric and non-parametric memories, 2022, arXiv preprint arXiv:2212.10511.
[41] Soyeong Jeong, Jinheon Baek, Sukmin Cho, Sung Ju Hwang, Jong C Park, Adaptive-rag: Learning to adapt retrieval-augmented large language models through question complexity, 2024, arXiv preprint arXiv:2403.14403.
[42] Qwen Team, Introducing qwen1.5, 2024.
[43] Edward J Hu, Yelong Shen, Phillip Wallis, Zeyuan Allen-Zhu, Yuanzhi Li, Shean Wang, Lu Wang, Weizhu Chen, Lora: Low-rank adaptation of large language models, 2021, arXiv preprint arXiv:2106.09685.
[44] Rohan Taori, Ishaan Gulrajani, Tianyi Zhang, Yann Dubois, Xuechen Li, Carlos Guestrin, Percy Liang, Tatsunori B Hashimoto, Stanford alpaca: an instruction-following llama model (2023). 1 (9) (2023). URL https://github.com/tatsu-lab/stanford_alpaca.
[45] Arvind Neelakantan, Tao Xu, Raul Puri, Alec Radford, Jesse Michael Han, Jerry Tworek, Qiming Yuan, Nikolas Tezak, Jong Wook Kim, Chris Hallacy, et al., Text and code embeddings by contrastive pre-training, 2022, arXiv preprint arXiv:2201.10005.
[46] Stephen E. Robertson, Steve Walker, Some simple effective approximations to the 2-poisson model for probabilistic weighted retrieval, in: SIGIR94: Proceedings of the Seventeenth Annual International ACM-SIGIR Conference on Research and Development in Information Retrieval, Organised By Dublin City University, Springer, 1994, pp. 232–241.
[47] Karl Cobbe, Vineet Kosaraju, Mohammad Bavarian, Mark Chen, Heewoo Jun, Lukasz Kaiser, Matthias Plappert, Jerry Tworek, Jacob Hilton, Reiichiro Nakano, et al., Training verifiers to solve math word problems, 2021, arXiv preprint arXiv:2110.14168.
[48] Dan Hendrycks, Collin Burns, Saurav Kadavath, Akul Arora, Steven Basart, Eric Tang, Dawn Song, Jacob Steinhardt, Measuring mathematical problem solving with the math dataset, 2021, arXiv preprint arXiv:2103.03874.
[49] Leland McInnes, John Healy, James Melville, Umap: Uniform manifold approximation and projection for dimension reduction, 2018, arXiv preprint arXiv:1802.03426.
[50] Josh Achiam, Steven Adler, Sandhini Agarwal, Lama Ahmad, Ilge Akkaya, Florencia Leoni Aleman, Diogo Almeida, Janko Altenschmidt, Sam Altman, Shyamal Anadkat, et al., Gpt-4 technical report, 2023, arXiv preprint arXiv:2303.08774.
[51] Yang Liu, Dan Iter, Yichong Xu, Shuohang Wang, Ruochen Xu, Chenguang Zhu, G-eval: Nlg evaluation using gpt-4 with better human alignment, 2023, arXiv preprint arXiv:2303.16634.
[52] Ying Jiao, Kumar Shridhar, Peng Cui, Wangchunshu Zhou, Mrinmaya Sachan, Automatic educational question generation with difficulty level controls, in: International Conference on Artificial Intelligence in Education, Springer, 2023, pp. 476–488.
[53] Kashyapa Niyarepola, Dineth Athapaththu, Savindu Ekanayake, Surangika Ranathunga, Math word problem generation with multilingual language models, in: Proceedings of the 15th International Conference on Natural Language Generation, 2022, pp. 144–155.
[54] Wei Qin, Xiaowei Wang, Zhenzhen Hu, Lei Wang, Yunshi Lan, Richang Hong, Math word problem generation via disentangled memory retrieval, ACM Trans. Knowl. Discov. Data 18 (5) (2024) 1–21.
[55] Zihao Zhou, Maizhen Ning, Qiufeng Wang, Jie Yao, Wei Wang, Xiaowei Huang, Kaizhu Huang, Learning by analogy: Diverse questions generation in math word problem, 2023, arXiv preprint arXiv:2306.09064.
[56] Long Ouyang, Jeffrey Wu, Xu Jiang, Diogo Almeida, Carroll Wainwright, Pamela Mishkin, Chong Zhang, Sandhini Agarwal, Katarina Slama, Alex Ray, et al., Training language models to follow instructions with human feedback, Adv. Neural Inf. Process. Syst. 35 (2022) 27730–27744.
[57] Gemini Team, Rohan Anil, Sebastian Borgeaud, Yonghui Wu, Jean-Baptiste Alayrac, Jiahui Yu, Radu Soricut, Johan Schalkwyk, Andrew M Dai, Anja Hauth, et al., Gemini: a family of highly capable multimodal models, 2023, arXiv preprint arXiv:2312.11805.
[58] Team GLM, Aohan Zeng, Bin Xu, Bowen Wang, Chenhui Zhang, Da Yin, Diego Rojas, Guanyu Feng, Hanlin Zhao, Hanyu Lai, Hao Yu, Hongning Wang, Jiadai Sun, Jiajie Zhang, Jiale Cheng, Jiayi Gui, Jie Tang, Jing Zhang, Juanzi Li, Lei Zhao, Lindong Wu, Lucen Zhong, Mingdao Liu, Minlie Huang, Peng Zhang, Qinkai Zheng, Rui Lu, Shuaiqi Duan, Shudan Zhang, Shulin Cao, Shuxun Yang, Weng Lam Tam, Wenyi Zhao, Xiao Liu, Xiao Xia, Xiaohan Zhang, Xiaotao Gu, Xin Lv, Xinghan Liu, Xinyi Liu, Xinyue Yang, Xixuan Song, Xunkai Zhang, Yifan An, Yifan Xu, Yilin Niu, Yuantao Yang, Yueyan Li, Yushi Bai, Yuxiao Dong, Zehan Qi, Zhaoyu Wang, Zhen Yang, Zhengxiao Du, Zhenyu Hou, Zihan Wang, ChatGLM: A family of large language models from GLM-130B to GLM-4 all tools, 2024.
[59] Baichuan, Baichuan 2: Open large-scale language models, 2023, arXiv preprint arXiv:2309.10305.
[60] Abhimanyu Dubey, Abhinav Jauhri, Abhinav Pandey, Abhishek Kadian, Ahmad Al-Dahle, Aiesha Letman, Akhil Mathur, Alan Schelten, Amy Yang, Angela Fan, et al., The llama 3 herd of models, 2024, arXiv preprint arXiv:2407.21783.
[61] Cheng-Han Chiang, Hung-yi Lee, Can large language models be an alternative to human evaluations? 2023, arXiv preprint arXiv:2305.01937.
[62] Yann Dubois, Chen Xuechen Li, Rohan Taori, Tianyi Zhang, Ishaan Gulrajani, Jimmy Ba, Carlos Guestrin, Percy S Liang, Tatsunori B Hashimoto, Alpacafarm: A simulation framework for methods that learn from human feedback, Adv. Neural Inf. Process. Syst. 36 (2024).
[63] Ben Naismith, Phoebe Mulcaire, Jill Burstein, Automated evaluation of written discourse coherence using GPT-4, in: Proceedings of the 18th Workshop on Innovative Use of NLP for Building Educational Applications, BEA 2023, 2023, pp. 394–403.
[64] Aman Gupta, Anup Shirgaonkar, Angels de Luis Balaguer, Bruno Silva, Daniel Holstein, Dawei Li, Jennifer Marsman, Leonardo O Nunes, Mahsa Rouzbahman, Morris Sharp, et al., RAG vs fine-tuning: Pipelines, tradeoffs, and a case study on agriculture, 2024, arXiv preprint arXiv:2401.08406.
[65] OpenAI, GPT-4o system card, 2024, Accessed 09 July 2024.