FlipVQA-Miner: Cross-Page Visual Question–Answer Mining from Textbooks

Zhen Hao Wong * **Jingwen Deng** * **Hao Liang** **Runming He**
Chengyu Shen **Wentao Zhang**†
Peking University
zhenhao1141@stu.pku.edu.cn wentao.zhang@pku.edu.cn

Abstract

The development of Large Language Models (LLMs) increasingly depends on high-quality supervised data, yet existing instruction-tuning and RL datasets remain costly to curate and often rely on synthetic samples that introduce hallucination and limited diversity. At the same time, textbooks and exercise materials contain abundant, high-quality human-authored Question-Answer(QA) content that remains underexploited due to the difficulty of transforming raw PDFs into AI-ready supervision. Although modern OCR and vision–language models can accurately parse document structure, their outputs lack the semantic alignment required for training. We propose an automated pipeline that extracts well-formed QA and visual-QA (VQA) pairs from educational documents by combining layout-aware OCR with LLM-based semantic parsing. Experiments across diverse document types show that the method produces accurate, aligned, and low-noise QA/VQA pairs. This approach enables scalable use of real-world educational content and provides a practical alternative to synthetic data generation for improving reasoning-oriented LLM training. All code and data-processing pipelines are open-sourced at https://github.com/OpenDCAI/DataFlow.

1 Introduction

Large Language Models (LLMs) have achieved remarkable performance across a wide range of natural language and reasoning tasks. Their success, however, critically depends on access to large-scale, high-quality training data. Early pretraining corpora were primarily constructed through extensive web scraping, covering diverse textual domains. While this approach provided massive coverage and linguistic variety, it also introduced noise, duplication, and factual inconsistencies. As LLMs have grown in scale, improvements in downstream performance increasingly rely not only on model architecture or size but also on human-aligned, high-quality supervision data.

To refine pretrained models, the community widely adopts Supervised Fine-Tuning (SFT) and Reinforcement Learning (RL) with human feedback. These processes endow LLMs with instruction-following and reasoning capabilities beyond basic language modeling. Recent studies, including DeepSeek (Guo et al., 2025), OpenAI's O-series, and Claude 4, indicate that RL plays a particularly important role in achieving factual grounding and reasoning stability. However, expanding such high-quality datasets is both costly and labor-intensive, requiring extensive human annotation, iterative feedback collection, and expert verification. As models scale, the marginal cost of obtaining additional reliable supervision rises sharply, posing a bottleneck to further improvement.

To address this challenge, some approaches attempt to generate synthetic supervision using LLMs(Wang et al., 2023; Tan et al., 2024; Zweiger et al., 2025). Although self-generated data can increase training volume, it often suffers from hallucination, stylistic uniformity, and limited generalization (Shumailov et al., 2023). Simultaneously, vast quantities of authentic human-authored content—including textbooks, exercises, and exam materials—remain underutilized. These sources contain rich, structured question–answer knowledge that captures genuine human reasoning and pedagogical intent. Despite advances in OCR and vision-language models such as DocLayout-YOLO (Zhao et al., 2024), LayoutLMv3 (Huang et al., 2022), and MinerU 2.5 (Niu et al., 2025), existing methods mostly focus on document readability or layout reconstruction rather than producing structured, AI-ready data suitable for fine-tuning.

Human curation of such materials into training-ready QA datasets is costly and inconsistent, heav-

*Equal contribution.
†Corresponding author.

ily dependent on annotator expertise and scalability (Sylolypavan et al., 2023; Gil-Gonzalez et al., 2025). In response, we propose an automated pipeline that accurately extracts Question-Answer (QA) and Visual Question-Answer (VQA) pairs from educational and scientific documents. Our approach bridges the gap between document-level OCR outputs and instruction-level supervision data, enabling scalable and reliable extraction of authentic human knowledge for instruction and reinforcement learning in large language models.

2 Related Works

2.1 Vision Language Models for Structured Document Understanding

Recent advances in vision language models (VLMs) have significantly improved the understanding and digitization of complex documents. Traditional OCR tools, such as Tesseract (Pendlebury et al., 2019), convert scanned pages into text but often lose layout information crucial for preserving document semantics. Layout-aware models, including LayoutLMv3 (Huang et al., 2022), DocLayout-YOLO (Zhao et al., 2024), Pix2Struct (Lee et al., 2023), and more recently MinerU 2.5 (Niu et al., 2025), unify text recognition, layout parsing, and language modeling within a single framework. These approaches can reconstruct document contents from PDFs with impressive accuracy, recovering both textual and structural information such as tables, equations, and multi-column layouts. However, the outputs of these models are primarily optimized for readability rather than structured supervision. The recognized text lacks explicit semantic annotations, such as question–answer boundaries, reasoning steps, or task-oriented labels needed for effective LLM fine-tuning.

2.2 Authentic Data Mining for Instruction and Reinforcement Learning

Supervised fine-tuning (SFT) and reinforcement learning (RL) heavily depend on high-quality instruction–response datasets. Many existing corpora, such as Self-Instruct (Wang et al., 2023), Alpaca-Instruct (Chen et al., 2023), and the OpenAssistant Conversations Dataset (OASST1) (Köpf et al., 2023), primarily depend on synthetic data generated by LLMs or crowd-sourcing, which often introduces hallucination and stylistic homogeneity. Recent efforts have begun emphasizing authentic data mining, deriving supervision signals from real human-authored materials to improve factuality and diversity while reducing generation costs. In this context, educational documents represent a rich, underutilized resource containing naturally curated human reasoning. By extracting QA pairs from OCRed PDFs, our method enables scalable collection of genuine human knowledge, providing a low-cost and low-hallucination alternative to synthetic instruction data for advancing reasoning-oriented LLM training.

3 Method: Visual Question-Answer (VQA) Pair Extraction

We aim to extract high-quality VQA pairs from PDF documents that may contain complex layouts and formats, multiple languages, and both textual and visual content. Our approach supports both **interleaved** VQA pairs where questions and answers appear alternately, and **long-distance** VQA pairs where questions and answers appear across different sections or documents.

As shown in Figure 1, our approach integrates MinerU (Wang et al., 2024; Niu et al., 2025), a document parsing toolkit, with LLMs, for efficient and semantically accurate extraction. MinerU provides a structured representation of the document content, while the LLM performs block grouping, QA pairing and image inserting on the extracted elements. This hybrid design achieves both content precision, semantic accuracy, and affordable computation cost, enabling scalable and high-quality extraction of both interleaved and long-distance VQA pairs.

3.1 Structural Content Extraction via MinerU

MinerU first converts raw PDF files into a structured, machine-readable format (e.g., JSON). It detects and preserves layout information, text blocks, tables, and embedded images along with their spatial metadata. This step produces a fine-grained decomposition of the document into content blocks, which greatly reduces the input and output complexity for subsequent LLM processing. Although MinerU itself cannot infer semantic relationships, such as grouping blocks into a single question or aligning each question with its corresponding answer, it provides a clean structural foundation on which higher-level reasoning can operate.

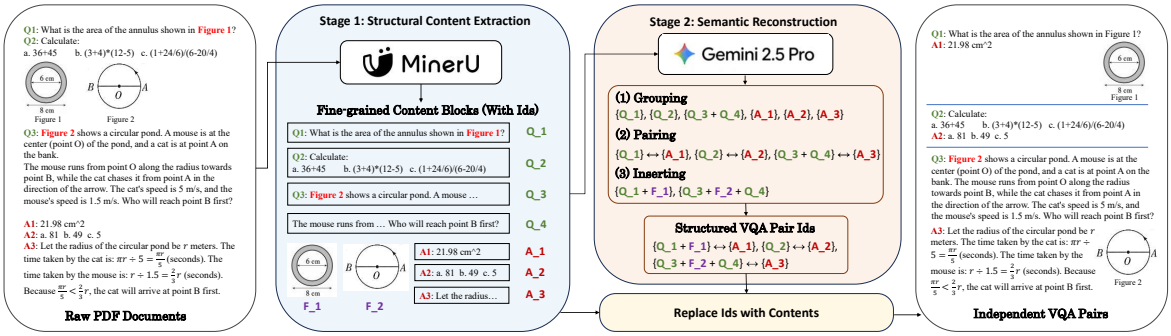


Figure 1: Our approach of VQA pair extraction from raw PDF documents. MinerU provides a structured representation of the document content, while the LLM performs block grouping, QA pairing and image inserting on the extracted elements.

3.2 Semantic Reconstruction via LLM

Directly prompting LLMs to read raw PDF text and extract VQA pairs is possible but computationally expensive[1] and less controllable. By first leveraging MinerU, we enable the LLM to output block identifiers instead of original content, thus reducing the output token length, and avoid text parsing errors from LLMs (MinerU is trained specifically for text parsing and is more accurate than general LLMs). In addition, the LLM will be able to reason over structured elements rather than unstructured PDF documents, producing VQA pairs with higher accuracy.

We assign each text or image block from MinerU a unique identifier, and prompt the LLM to perform three key reasoning operations: (1) **Grouping** — Merge fragmented text blocks that belong to the same question or answer. (2) **Pairing** — Extract hierarchical metadata (e.g., chapter titles, question numberings) to match question–answer pairs across pages or even across documents. (3) **Inserting** — Place relevant images near their associated question or answer to form coherent multimodal QA pairs. The prompt is provided in Section B.

Finally, we reconstruct each VQA pair by substituting the identifiers with the original text and images from MinerU's output.

4 Experiments

4.1 Experimental Settings

We evaluate our VQA pair extraction pipeline on three representative documents selected to reflect distinct structural challenges encountered in educational materials:

1. **Interleaved VQA pairs:** A student solution manual on complex analysis, in which questions and answers appear in short-range, tightly interwoven sequences.

2. **Long-distance VQA pairs:** A textbook on abstract algebra, where questions and their corresponding answers are separated by substantial amounts of intervening content, often across multiple pages.

3. **Multi-column, Chinese long-distance VQA pairs:** A Chinese middle-school mathematics exercise book that combines a double-column layout with long-distance question–answer relationships, thereby introducing both multilingual and multi-column parsing complexity.

These documents jointly capture the principal structural patterns of interleaving, long-distance separation, and multi-column formatting, and therefore provide a comprehensive basis for evaluating the robustness of VQA pair extraction.

For structural parsing, we employ **MinerU 2.5** (VLM backend) (Niu et al., 2025), which exhibits strong capabilities in layout recognition, reading order determination, and textual understanding. For question grouping, answer pairing, and image insertion, we utilize **Gemini-2.5-pro** (Comanici et al., 2025) as the LLM component, leveraging its robust instruction-following ability and proficiency in generating coherent QA pairings.

4.2 Evaluation Protocol

Extraction quality is assessed through manual evaluation along two modalities:

1. **Text:** Evaluation considers whether each extracted question correctly matches its corresponding answer, whether the QA pairs are

[1] Our method costs $0.012 per question, while the direct prompting method costs $0.046 per question, tested on a Chinese middle school math exercise book.

Document	Type	Layout	Modality	#Samples	Precision	Recall	F1
Contemporary Abstract Algebra (Joseph A. Gallian)	Long-distance	Single-column	Text	1281	0.9968	0.9766	0.9866
			Vision	27	1.0000	0.9259	0.9615
Solutions Manual for Complex Analysis (T. W. Gamelin)	Interleaved	Single-column	Text	390	0.9797	0.9897	0.9847
			Vision	294	1.0000	0.9456	0.9720
Chinese middle school math exercise book	Long-distance	Multi-column	Text	445	0.9911	0.9955	0.9933
			Vision	266	1.0000	0.9774	0.9886

Table 1: Extraction performance on different structural pattern documents. #Samples indicates the number of QA pairs for text and the number of images for vision.

complete and properly ordered, and whether no extra or hallucinated text is introduced. This assesses MinerU's text parsing ability as well as the grouping and pairing operations performed by the LLM.

2. **Vision:** Evaluation considers whether images are inserted at the correct locations relative to their associated questions or answers. This assesses both MinerU's image segmentation performance and the LLM's image-insertion step.

We focus on high-level structural errors, such as reading order or bounding box inaccuracies, while ignoring minor text-level parsing issues, for instance formula errors, which are difficult to identify manually. Notably, MinerU achieves a text F1 score above 94 on multiple benchmarks, indicating that such minor errors introduce minimal noise.

4.3 Experimental Results

4.3.1 Quantitative Results

We evaluate our FlipVQA-Miner extraction pipeline on three representative documents with distinct structural patterns, layouts and languages. Table 1 presents the quantitative results for both text and vision modalities.

Text extraction. The pipeline achieves F1 scores above 0.98 across all documents, indicating robust handling of interleaved, long-distance, and multi-column QA pairs, including non-English content.

Vision extraction. Image placement precision is consistently 1.0, with F1 scores ranging from 0.9615 to 0.9886, demonstrating stable performance across different layouts and languages.

These results indicate that our pipeline generalizes effectively across diverse document structures, languages, and presentation formats. It achieves consistently high extraction accuracy for interleaved sequences, long-distance QA pairs, multi-column layouts, and multilingual content, demonstrating its robustness and potential for broader applications in automated VQA pair extraction.

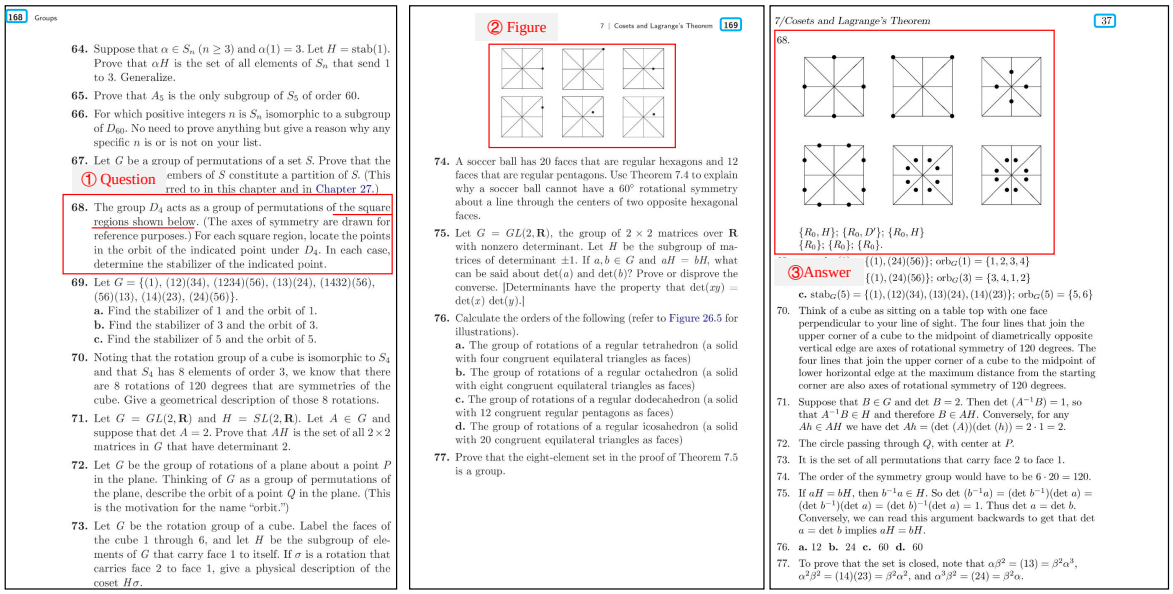
4.3.2 Qualitative Results

To demonstrate the effectiveness of FlipVQA-Miner, we present a representative example in which the components of a single question are distributed across distant pages. As shown in Figure 2, the question (①) refers to "the square regions shown below", yet the corresponding figure (②) appears on the next page and is neither captioned nor labeled. The answer (③) to this question is found not only far from the question itself but even located in a different companion book, rather than in the same volume as the question. This separation further increases the difficulty of linking the question with its corresponding figure and answer.

Figure 2a presents the extracted and grouped output generated by our pipeline for a case in Figure 2b. Although the case is quite challenging, FlipVQA-Miner successfully retrieves these elements and assembles them into single structured VQA instance, demonstrating its ability to resolve long-range and cross-document dependencies. More representative cases are provided in Appendix A.

5 Conclusion

Across three documents with diverse structural properties and modality configurations, our experiments demonstrate that the proposed extraction pipeline can reliably recover both textual QA relationships and their associated visual elements. Manual evaluation confirms that the system consistently identifies complete and correctly ordered QA pairs, while maintaining a low rate of hallucinated or extraneous content. Moreover, images are accurately localised and inserted relative to their corresponding textual components, indicating that the integration between MinerU's structural parsing

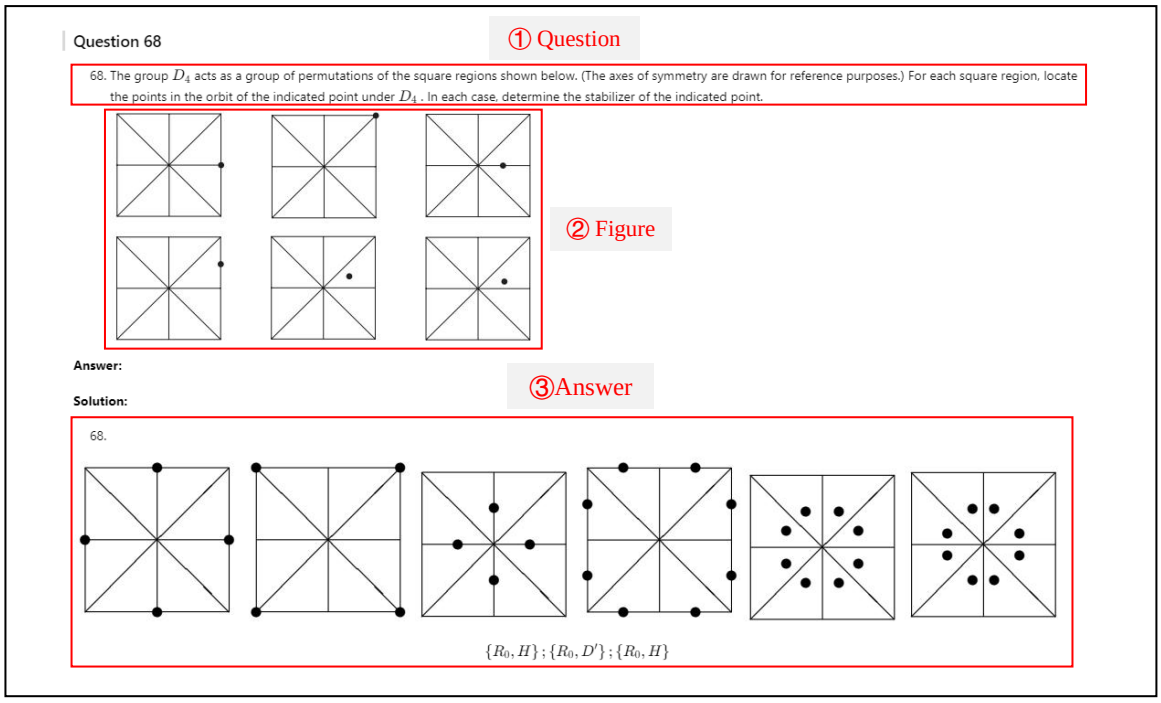


Page 168 of Contemporary Abstract Algebra (Joseph A. Gallian)

Page 169 of Contemporary Abstract Algebra (Joseph A. Gallian)

Page37 of Student Solutions Manual for Gallians Contemporary Abstract Algebra

(a) Original question, figure and answer arrangement in PDF.



(b) FlipVQA-Miner extraction and grouping results (Markdown rendered).

Figure 2: Demonstration of FlipVQA-Miner's ability to resolve long-distance associations. Although the question (①), figure (②), and answer (③) are separated across different pages and materials, the system successfully identifies, extracts, and groups them into a coherent VQA unit, with the final result rendered as Markdown.

and the LLM's semantic grouping is robust across long-distance and interleaved settings. Although minor OCR imperfections persist, their impact on the resulting QA/VQA pairs remains negligible for training purposes. Overall, these results validate that our pipeline is capable of transforming heterogeneous educational documents into high-quality, AI-ready supervision data at scale.

6 Future Work

While our current pipeline demonstrates strong performance in extracting high-quality QA and VQA pairs from structured educational documents, several research directions remain open. A natural next step is to explore how large collections of parsed educational materials can support broader evaluation of reasoning-oriented language models. In particular, automatically extracted questions from textbooks and examinations may enable the construction of curriculum-aligned benchmarks that capture fine-grained conceptual difficulty and domain progression. Another promising direction is to investigate how these authentic QA pairs can be leveraged to build large-scale training datasets for supervised fine-tuning and reinforcement learning, especially in domains where synthetic data remains unreliable. Finally, understanding how to model difficulty, quality, and diversity within extracted educational content may lead to more principled data curation strategies and contribute to the development of more robust and generalizable LLMs.

Specifically, we have a detailed plan of benchmark curation. Although the extracted VQA pairs faithfully capture the content of the original instructional materials, the raw data are not directly usable as an evaluation benchmark. Many items lack automatically verifiable answers (or intrinsically not verifiable), depend on missing contextual information, or are too trivial to serve as meaningful evaluation samples. We therefore design a multi-stage curation pipeline to ensure that the benchmark is clean, verifiable, self-contained, challenging, and diverse across different problem types and modalities. The curation process includes answer refinement, question-type selection, modality separation, and quality filtering, as detailed below.

Short Answers and Full Solutions The extracted answers exhibit two main forms: concise, directly verifiable short answers and lengthy full solutions. To promote automatic evaluation, we prompt LLMs to extract short, canonical answers from long-form solutions. However, for problems whose answers are inherently non-verifiable (e.g., proofs or open-ended explanations), we exclude them from the benchmark, since their correctness generally requires formal reasoning beyond current verification methods.

Question Type Classification and Selection We instruct LLMs to classify each question into one of several types: proof, explanation, fill-in-the-blank, calculation, multiple-choice, drawing, and others. To maintain automatic verifiability while preserving diversity, we currently retain fill-in-the-blank, calculation, and multiple-choice questions.

Modalities We further divide problem into two modality groups, text-only and text–image, to separately evaluate models with and without multimodal perception capabilities. This separation allows fair comparison between purely textual LLMs and multimodal models.

Problem Filtering To ensure benchmark quality, we apply several filtering criteria: (1) **Completeness and self-containment.** We instruct LLMs to remove problems that are incomplete or context-dependent (e.g., "According to Theorem X," "See previous result," or "Answered in text"), as such items cannot serve as standalone evaluation samples. (2) **Difficulty.** We evaluate several LLM on the problems, and remove questions that are either trivial or excessively difficult, maintaining a reasonable difficulty distribution for general-purpose evaluation.

References

Lichang Chen, Shiyang Li, Jun Yan, Hai Wang, Kalpa Gunaratna, Vikas Yadav, Zheng Tang, Vijay Srinivasan, Tianyi Zhou, Heng Huang, and 1 others. 2023. Alpagasus: Training a better alpaca with fewer data. *arXiv preprint arXiv:2307.08701*.

Gheorghe Comanici, Eric Bieber, Mike Schaekermann, Ice Pasupat, Noveen Sachdeva, Inderjit Dhillon, Marcel Blistein, Ori Ram, Dan Zhang, Evan Rosen, and 1 others. 2025. Gemini 2.5: Pushing the frontier with advanced reasoning, multimodality, long context, and next generation agentic capabilities. *arXiv preprint arXiv:2507.06261*.

Julian Gil-Gonzalez, David Cárdenas-Peña, Álvaro A Orozco, German Castellanos-Dominguez, and Andrés Marino Álvarez-Meza. 2025. Crowdattention: An attention based framework to classify crowdsourced data in medical scenarios. *Sensors*, 25(20):6435.

Daya Guo, Dejian Yang, Haowei Zhang, Junxiao Song, Ruoyu Zhang, Runxin Xu, Qihao Zhu, Shirong Ma, Peiyi Wang, Xiao Bi, and 1 others. 2025. Deepseek-r1: Incentivizing reasoning capability in llms via reinforcement learning. *arXiv preprint arXiv:2501.12948*.

Yupan Huang, Tengchao Lv, Lei Cui, Yutong Lu, and Furu Wei. 2022. Layoutlmv3: Pre-training for document ai with unified text and image masking. In *Proceedings of the 30th ACM international conference on multimedia*, pages 4083–4091.

Andreas Köpf, Yannic Kilcher, Dimitri Von Rütte, Sotiris Anagnostidis, Zhi Rui Tam, Keith Stevens, Abdullah Barhoum, Duc Nguyen, Oliver Stanley, Richárd Nagyfi, and 1 others. 2023. Openassistant conversations-democratizing large language model alignment. *Advances in neural information processing systems*, 36:47669–47681.

Kenton Lee, Mandar Joshi, Iulia Raluca Turc, Hexiang Hu, Fangyu Liu, Julian Martin Eisenschlos, Urvashi Khandelwal, Peter Shaw, Ming-Wei Chang, and Kristina Toutanova. 2023. Pix2struct: Screenshot parsing as pretraining for visual language understanding. In *International Conference on Machine Learning*, pages 18893–18912. PMLR.

Junbo Niu, Zheng Liu, Zhuangcheng Gu, Bin Wang, Linke Ouyang, Zhiyuan Zhao, Tao Chu, Tianyao He, Fan Wu, Qintong Zhang, and 1 others. 2025. Mineru2. 5: A decoupled vision-language model for efficient high-resolution document parsing. *arXiv preprint arXiv:2509.22186*.

Feargus Pendlebury, Fabio Pierazzi, Roberto Jordaney, Johannes Kinder, and Lorenzo Cavallaro. 2019. {TESSERACT}: Eliminating experimental bias in malware classification across space and time. In *28th USENIX security symposium (USENIX Security 19)*, pages 729–746.

Ilia Shumailov, Zakhar Shumaylov, Yiren Zhao, Yarin Gal, Nicolas Papernot, and Ross Anderson. 2023. The curse of recursion: Training on generated data makes models forget. *arXiv preprint arXiv:2305.17493*.

Aneeta Sylolypavan, Derek Sleeman, Honghan Wu, and Malcolm Sim. 2023. The impact of inconsistent human annotations on ai driven clinical decision making. *NPJ Digital Medicine*, 6(1):26.

Zhen Tan, Dawei Li, Song Wang, Alimohammad Beigi, Bohan Jiang, Amrita Bhattacharjee, Mansooreh Karami, Jundong Li, Lu Cheng, and Huan Liu. 2024. Large language models for data annotation and synthesis: A survey. *arXiv preprint arXiv:2402.13446*.

Bin Wang, Chao Xu, Xiaomeng Zhao, Linke Ouyang, Fan Wu, Zhiyuan Zhao, Rui Xu, Kaiwen Liu, Yuan Qu, Fukai Shang, and 1 others. 2024. Mineru: An open-source solution for precise document content extraction. *arXiv preprint arXiv:2409.18839*.

Yizhong Wang, Yeganeh Kordi, Swaroop Mishra, Alisa Liu, Noah A Smith, Daniel Khashabi, and Hannaneh Hajishirzi. 2023. Self-instruct: Aligning language models with self-generated instructions. In *Proceedings of the 61st annual meeting of the association for computational linguistics (volume 1: long papers)*, pages 13484–13508.

Zhiyuan Zhao, Hengrui Kang, Bin Wang, and Conghui He. 2024. Doclayout-yolo: Enhancing document layout analysis through diverse synthetic data and global-to-local adaptive perception. *arXiv preprint arXiv:2410.12628*.

Adam Zweiger, Jyothish Pari, Han Guo, Ekin Akyürek, Yoon Kim, and Pulkit Agrawal. 2025. Self-adapting language models. *arXiv preprint arXiv:2506.10943*.

A Case Studies of VQA Pair Extraction

To demonstrate the robustness of our pipeline in resolving diverse structural ambiguities commonly found in educational materials, we present a set of representative examples. Each example includes the original question, the original solution(s), and the corresponding extracted QA or VQA pair rendered in Markdown. These cases illustrate how the system reconstructs raw, unstructured content into a coherent, machine-readable format.

A.1 Interleaved QA with multiple solutions case

This case features an interleaved question (Figure 3) accompanied by two distinct solution paths. Despite the intertwined layout, our pipeline faithfully preserves both solutions and correctly associates them with the question. The resulting structured output is shown in Figure 4.

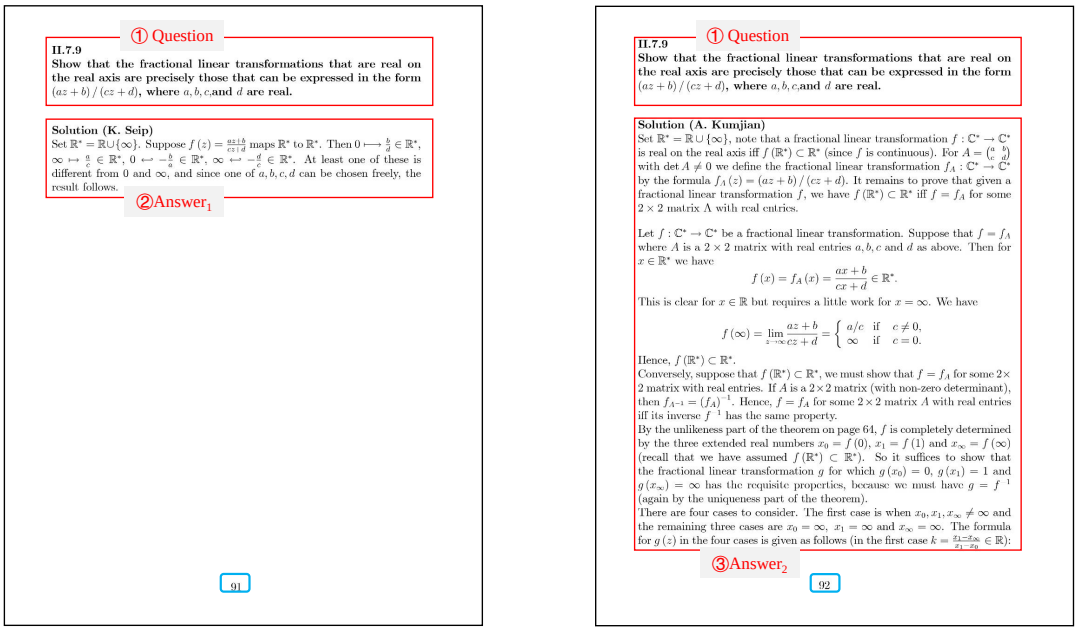


Figure 3: Original question with two interleaved solutions.

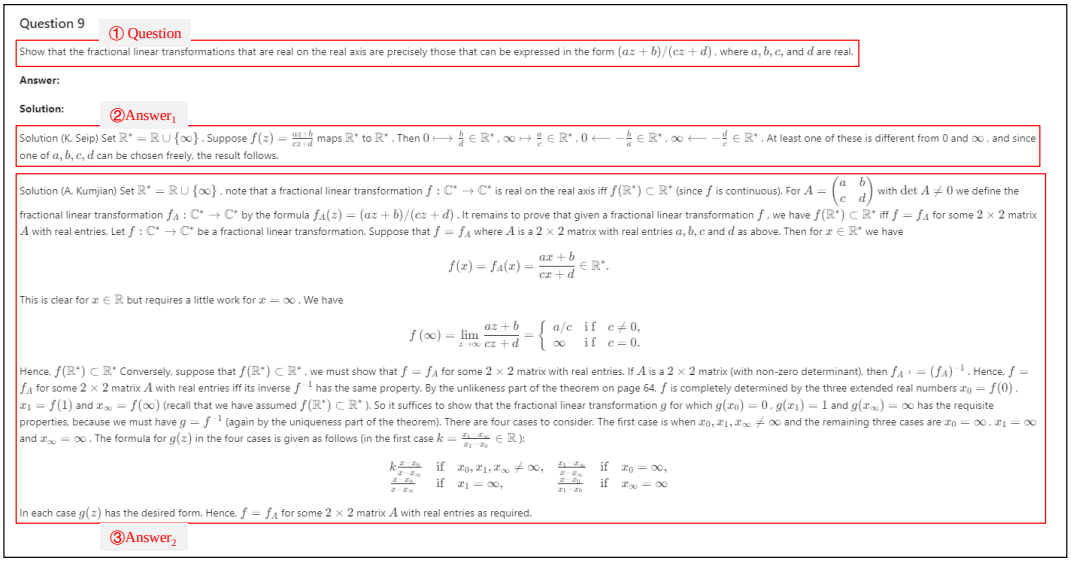


Figure 4: Extracted and reconstructed QA pair rendered in Markdown.

A.2 Multi-column, cross-page Chinese exercise book case

This example showcases a particularly challenging scenario drawn from a Chinese exercise book. As shown in Figure 5, the question appears within a dense multi-column layout interspersed with unrelated textual elements. Complicating matters further, the corresponding answer (Figure 6) is located in a separate PDF file altogether. Despite these structural and formatting hurdles, our pipeline accurately identifies, extracts, and associates the VQA pair. The final Markdown-rendered output (Figure 7) highlights the system's resilience to multi-column layouts, cross-page and cross-document separation, heterogeneous formatting conventions, and multilingual content.

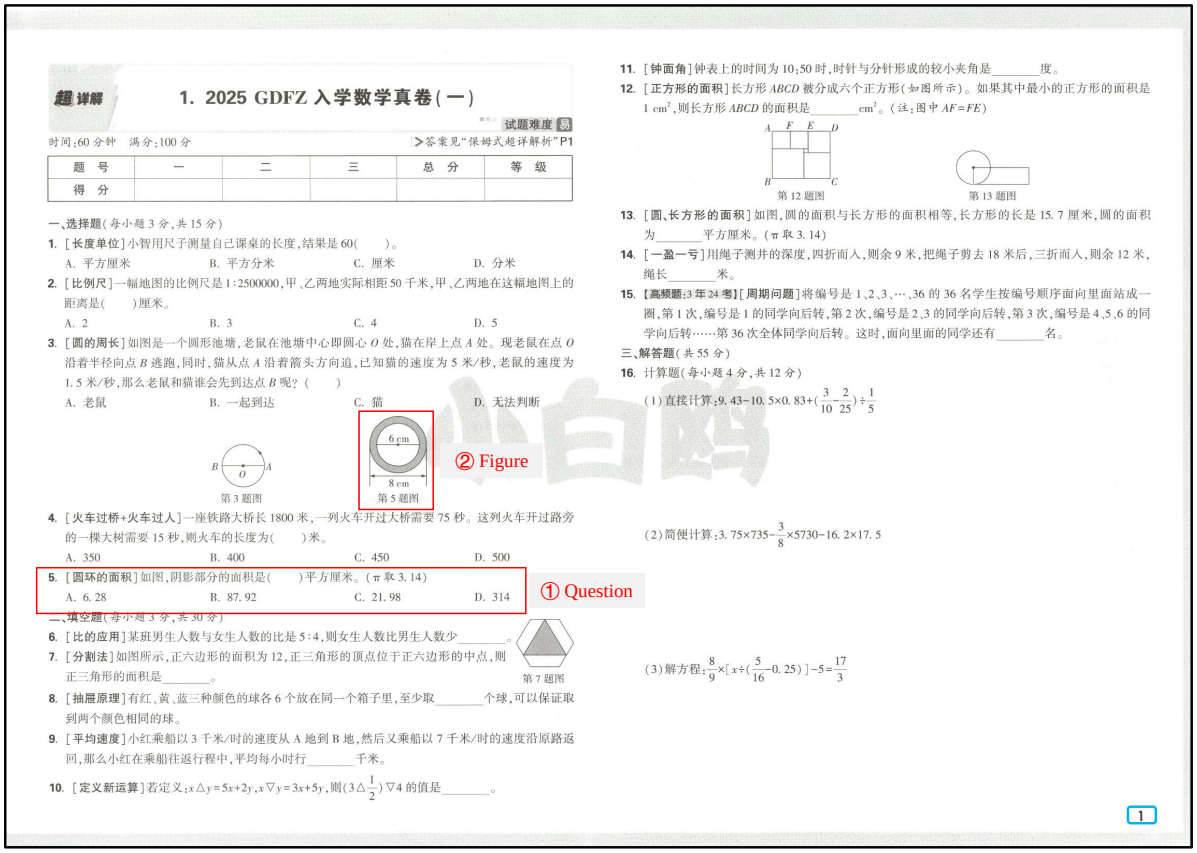


Figure 5: Original question within a multi-column Chinese exercise book layout.

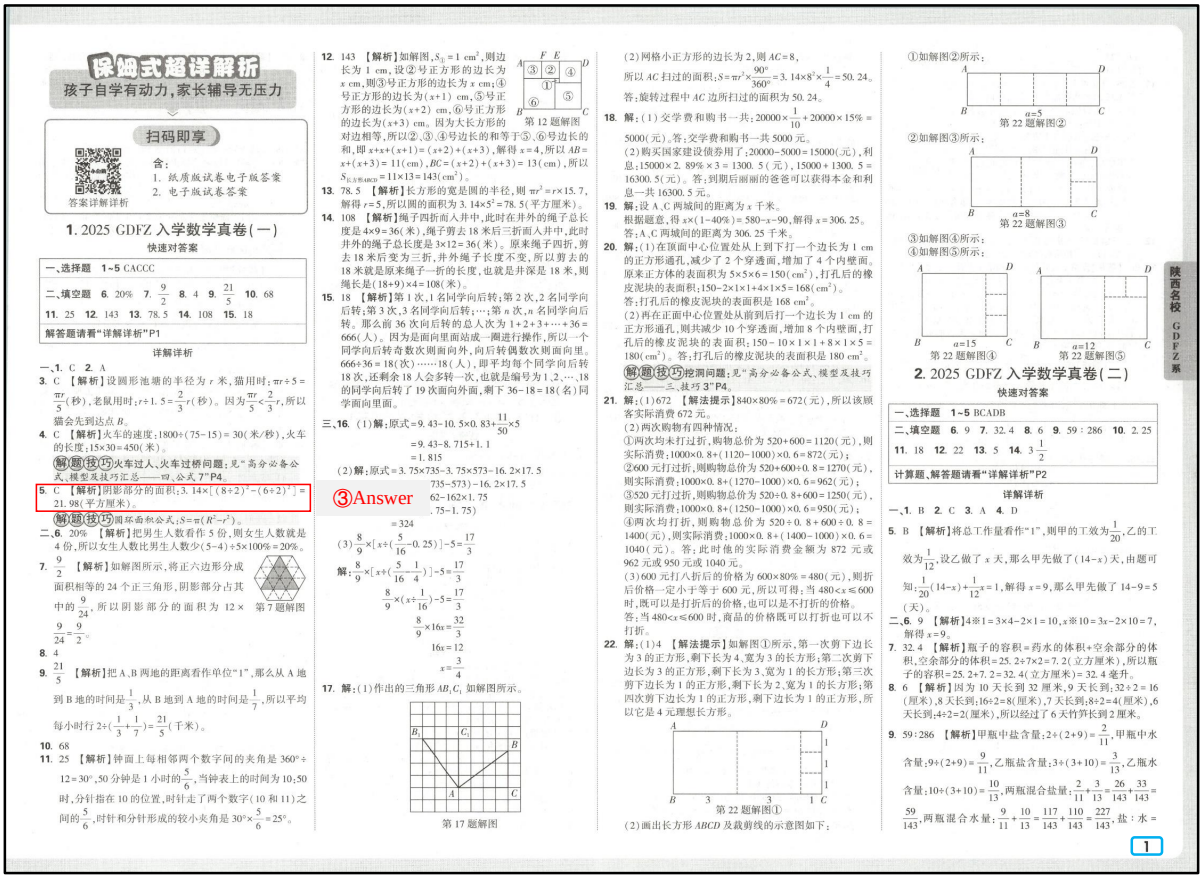


Figure 6: Original answer located in a separate multi-column document.

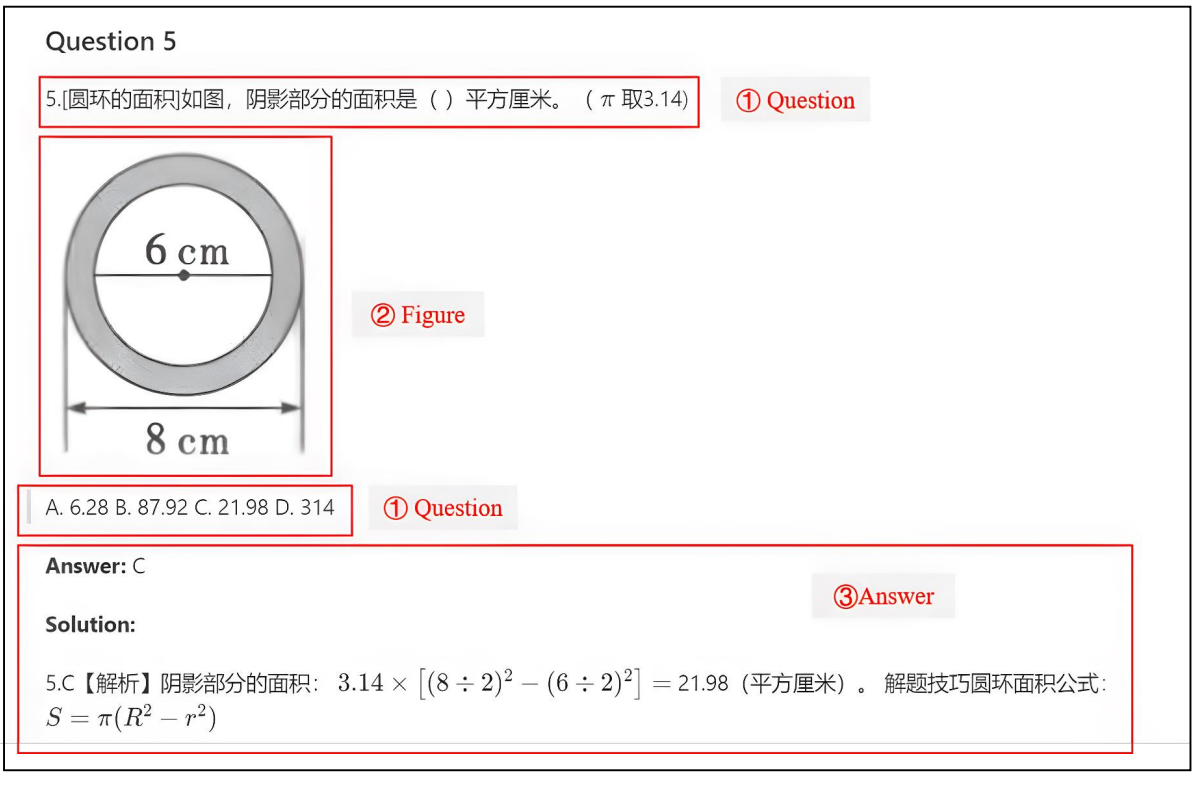


Figure 7: Extracted and reconstructed VQA pair rendered in Markdown.

B Prompt of VQA Pair Extraction

The prompt we used for VQA pair extraction (Section 3.2) is provided below:

VQA Pair Extraction

```
You are an expert in {subject}. You are given a json file. Your task is to segment the content, insert images tags, and extract
 labels:
1. Every json item has an "id" field. Your main task is to output this field.
2. You need to segment the content into multiple `<qa_pair>`...`</qa_pair>` blocks, each containing a question and its
corresponding answer with solution.
3. If the problem or answer is not complete, omit them.
4. You need to put the images id into proper positions. You could look at the caption or context to decide where to put the
image tags.
5. You will also need to extract the chapter title and each problem's label/number from the text.
6. You only need to output "id" field for chapters, questions and solutions. DO NOT OUTPUT ORIGINAL TEXT. Use ',' to separate
different ids.
7. However, use original labels/numbers for labels, and use original numbers for answers. DO NOT output "id" field for labels
and answers. You will need to extract them from the text.

Strict extraction rules:
** About questions and answers/solutions **
- Preserve each problem's original label/number, such as "Example 3", "Exercise 1", "11". Do not include the period after the
number. Use Arabic numerals only. For example, if the label is "IV", convert it to "4".
- If the full label is "III.16", keep only "16". If the full label is "5.4", keep only "4".
- If there are multiple sub-questions (such as "(1)", "(a)") under one main question, always put them together in the same `<
qa_pair>`...`</qa_pair>` block.
- If a question and its answer/solution are contiguous, wrap them together as a single `<qa_pair>`...`</qa_pair>` block, e.g.:
  `<qa_pair><label>Example 1</label><question>...</question><answer>...</answer><solution>...</solution></qa_pair>`
- If only questions or only answers with solutions appear, wrap each question or answer with solution in a `<qa_pair>`...`</
qa_pair>` block with the missing part left empty. For example, if only questions appear:
  `<qa_pair><label>Example 1</label><question>...</question><answer></answer><solution></solution></qa_pair>`
- If multiple questions and solutions appear, wrap each question/solution pair in its own `<qa_pair>`...`</qa_pair>` block.
- If you do not see the full solution, only extract the short answer and leave the solution empty. YOU MUST KEEP QUESTIONS WITH
 ONLY SHORT ANSWERS !!!
** About chapter/section titles **
- Always enclose qa pairs in a `<chapter>`...`</chapter>` block, where <title>MAIN_TITLE</title> is the chapter title or
section title.
- Normally, chapter/section titles appear before the questions/answers in an independent json item.
- There could be multiple `<chapter>`...`</chapter>` blocks if multiple chapters/sections exist.
- **Any titles followed by a question/answer whose label/number is not 1, or with a score, should NOT be extracted.**
- Do not use nested titles.
- Leave the title blank if there is no chapter title.
** About figures/diagrams **
- Whenever the question or answer/solution refers to a figure or diagram, record its "id" in question/answer/solution just like
 other text content.
- You MUST include all images referenced in the question/answer/solution.


If no qualifying content is found, output:
<empty></empty>

Output format (all tags run together, no extra whitespace or newlines except between entries):
<chapter><title>MAIN_TITLE_ID</title>
<qa_pair><label>...</label><question>QUESTION_IDS</question>
<answer>ANSWER(EXTRACTED FROM SOLUTION)</answer><solution>SOLUTION_IDS</solution></qa_pair>
<qa_pair><label>...</label><question>QUESTION_IDS</question>
<answer>ANSWER(EXTRACTED FROM SOLUTION)</answer><solution></solution></qa_pair>
</chapter>
<chapter><title>MAIN_TITLE</title>
<qa_pair><label>...</label><question>QUESTION_IDS</question>
<answer>ANSWER(EXTRACTED FROM SOLUTION)</answer><solution>SOLUTION_IDS</solution></qa_pair>
</chapter>


Example:
<chapter><title>1</title>
<qa_pair><label>Example 1</label><question>2,3</question>
<answer>4/5</answer><solution>5,6,7</solution></qa_pair>
<qa_pair><label>Example 2</label><question>8,9,10</question>
<answer>3.14</answer><solution></solution></qa_pair>
</chapter>
<chapter><title>12</title>
<qa_pair><label>Example 1</label><question>13,14</question>
<answer>2^6</answer><solution>16</solution></qa_pair>
</chapter>

Please now process the provided json and output your result.
```