

When the Evidence Is Right, the Math Is Easy: Understanding the Challenges of Document-Grounded Reasoning

Aftab Hossain Mikat and Tamjid Hasan Fahim

Department of Computer Science and Engineering
Rajshahi University of Engineering & Technology, Bangladesh
ahmikat@gmail.com, tamjidhfahim@gmail.com

Abstract

Document-grounded quantitative reasoning requires a system to first understand a visually presented document and then compute an exact answer from the grounded content. We describe LILIPUT, a zero-shot submission to the DocSem 2026 shared task, in which Gemini 3.8 Flash performs visual evidence localization and converts the selected content into a faithful, self-contained math problem, while DeepSeek V4 performs the downstream numerical reasoning. On 1,730 held-out test instances, our accepted submission obtains 49.54% joint exact accuracy, 55.09% answer accuracy, and 66.76% macro evidence F1. In contrast, experiments on 908 labeled training instances show much higher answer accuracy when the relevant evidence is successfully extracted: Gemini 3.7 Flash achieves 95.59%, while text-based solvers achieve 92.51%–97.03%. Although these results are not directly comparable, the large gap suggests that calculation is not the main challenge. Instead, the main difficulty is correctly understanding and extracting information from the document. We found that clean and clear documents were easier to process, while blurry, dense, and visually complex documents made extraction much harder, even after image enhancement. Since DeepSeek V4 uses the extracted content for calculation, errors during visual extraction can also lead to incorrect answers. These findings highlight visual document understanding as a key challenge for document-grounded quantitative reasoning.

1 Introduction

Quantitative questions about documents combine three problems that are often evaluated separately: reading a visually structured page, grounding a query in the correct passage, and executing a calculation. A system may compute flawlessly from clean text yet fail because a decimal, unit, or evidence block was read incorrectly. Conversely, it may recover the right answer without returning the

exact visible blocks required for attribution. DocSem 2026 makes this coupling explicit: given a PDF and a paraphrased query, a system must return a numerical answer together with the supporting block-ID set.

We participated with a modular zero-shot pipeline that separates document perception from answer generation. The first stage enhances page images and the query into structured evidence plus a faithful math problem; the second stage solves that problem as text. This separation lets a vision-language model concentrate on layout and transcription while a language model concentrates on semantic composition and arithmetic. It also exposes the pipeline’s failure boundary: extraction mistakes become the solver’s input, and correct answers can still fail joint scoring when the evidence set is incomplete or over-inclusive.

Our contributions are threefold. First, we describe the complete submitted pipeline, including adaptive image preprocessing, structured extraction, verification, and output validation. Second, we report the official held-out result and carefully separate it from public-train diagnostics that use different model and input configurations. Third, we analyze answer errors rather than treating them as undifferentiated arithmetic failures. The main finding is that clean-text solving is not the central bottleneck: end-to-end performance is limited by evidence localization, visual fidelity, semantic scope, and errors that cascade across stages.

2 Related Work

DocSem derives from GSM-SEM, which evaluates robustness to semantically varied arithmetic problems (Singh et al., 2026). This lineage builds on GSM8K (Cobbe et al., 2021); GSM-Symbolic similarly shows that model accuracy can be brittle under controlled changes to values and clauses (Mirzadeh et al., 2024). Program-aided approaches

such as PAL (Gao et al., 2023) and Program of Thoughts (Chen et al., 2023) motivate separating the interpretation of a problem from deterministic computation.

The visual component connects DocSem to document question answering. DocVQA established QA over document images and layout (Mathew et al., 2021), while OCRBench and OCRBench v2 evaluate recognition, localization, and reasoning under visually situated text (Liu et al., 2024; Fu et al., 2025). Donut avoids a conventional OCR interface by learning end-to-end document understanding (Kim et al., 2022); LayoutLMv3 and Pix2Struct instead demonstrate complementary ways to combine visual, textual, and layout signals (Huang et al., 2022; Lee et al., 2023). DocSem additionally requires source attribution, echoing work on grounded document QA (Nourbakhsh et al., 2025). Numerical document benchmarks such as FinQA, TAT-QA, and MultiHiertt emphasize evidence selection and multi-step computation over text and tables (Chen et al., 2021; Zhu et al., 2021; Zhao et al., 2022). Our submission uses proprietary general-purpose models without task-specific training to explicitly assess how well off-the-shelf foundation models perform under a completely zero-shot setup.

3 Task and Data

Each instance contains a PDF and a paraphrased user_query. Visible document blocks are labeled with identifiers such as b07. A prediction is a JSON object containing the instance ID, a final answer, and a nonempty list of evidence block IDs. The official shared-task page¹ and public dataset release² describe 908 labeled training instances, 217 validation instances with organizer-held labels, and 1,730 held-out test instances. We use the August 31, 2026 training release, which incorporates seven corrected annotations; the organizers had also corrected three private validation labels by September 3.

Metrics. Answer exact accuracy uses case-insensitive, whitespace-normalized exact match, with numerically equivalent decimal strings treated as equal. Evidence exact accuracy requires equality of the predicted and gold block-ID sets. Per-

¹<https://docinsights-workshop.github.io/docinsights-2026/shared-task/>

²<https://huggingface.co/datasets/amitbcp/docinsights-2026-shared-task-data>

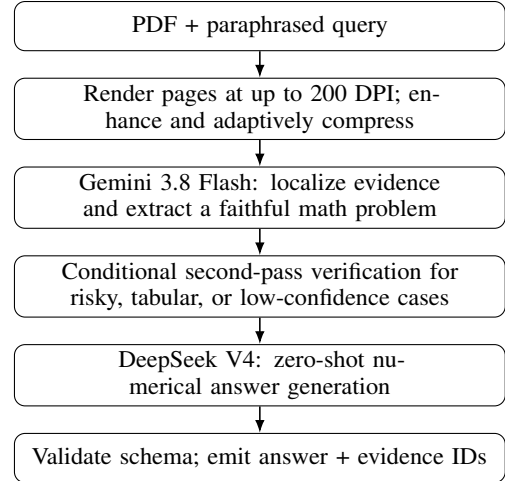


Figure 1: The submitted two-stage pipeline. The evidence IDs originate in the visual extraction stage and are preserved through answer generation.

instance evidence F1 is set F1, macro-averaged across instances. Joint exact accuracy requires both the answer and evidence set to be exactly correct. Because evidence F1 is a macro average, it cannot be converted into a count of exact evidence matches.

4 System Description

Figure 1 shows the main steps of our submitted system. The system first processes the input PDF and query, then uses a visual model to find and extract the relevant information. A second verification step is used for difficult or uncertain cases. Finally, the extracted problem is given to DeepSeek V4 for numerical reasoning, and the final answer and evidence IDs are validated before submission.

4.1 Page Rendering and Enhancement

We render each PDF page with PyMuPDF at up to 200 DPI. We then convert the pages to grayscale, apply contrast-limited adaptive histogram equalization (CLAHE), increase the global contrast by 18%, and use mild thresholded unsharp masking. We avoid hard binarization because it may remove important details such as decimal points, percentage signs, currency symbols, and thin table lines. The processed pages are then encoded as WebP. If the total image size exceeds 6 MiB, we reduce the DPI and image quality adaptively. This preprocessing was mainly used to handle the degraded quality of some test documents.

4.2 Evidence and Problem Extraction

The ordered page images and query are supplied to Gemini 3.8 Flash with medium reasoning, temperature 0, and a structured-output instruction. The requested record contains the supporting block IDs, verbatim block text, page numbers, content types, confidence or risk indicators, and an isolated math-problem statement. The extraction instruction emphasizes accuracy so that quantities, units, entities, and the requested target remain faithful to the document. Risky, ambiguous, low-confidence, and table-based cases are checked with an additional verification call. The selected evidence IDs are then directly used in the final prediction rather than being selected again by the solver. The complete extraction instruction and output schema are provided in Appendix B.

4.3 Answer Generation and Validation

The numerical models receive the extracted math problem as text and generate the final answer without examples or task-specific training. In the submitted test pipeline, DeepSeek V4 is used as the numerical solver. For the public-train diagnostics, GPT-5.6, Qwen 3.7 Plus, and DeepSeek V4 independently solve the same extracted problem. None of these models receives the gold answer, worked examples, or task-specific fine-tuning. The evidence IDs are selected by the Gemini 3.8 extraction stage and kept unchanged in the final prediction. The downstream solving instruction is reproduced in Appendix B.

5 Experimental Setup

We evaluate our system in two settings. The **official test** setting uses the complete submitted pipeline on all 1,730 held-out instances. In this pipeline, Gemini 3.8 Flash receives the PDF and query, retrieves the relevant evidence, and extracts it as a self-contained math problem. DeepSeek V4 then receives this extracted math text and generates the final numerical answer zero-shot.

The **public-train diagnostic** uses the 908 labeled training instances. In this setting, we first use Gemini 3.8 Flash to retrieve the relevant evidence and extract the mathematical text from each PDF. The extracted math text is then given separately to GPT-5.6, Qwen 3.7 Plus, and DeepSeek V4. Each model independently solves the extracted problem zero-shot. Gemini 3.7 Flash directly takes each PDF and query and performs both evidence

Metric	Score (%)
Joint exact accuracy	49.5376
Answer exact accuracy	55.0867
Evidence F1 (macro)	66.7630

Table 1: Official accepted test result on 1,730 instances (attempt 1; receipt 2cea930f).

System	Answer	Evidence F1	Joint
GPT-5.6	97.03	100.00	97.03
Gemini 3.7 Flash	95.59	100.00	95.59
Qwen 3.7 Plus	95.37	100.00	95.37
DeepSeek V4	92.51	100.00	92.51

Table 2: Zero-shot results on the 908 public training tasks (%). Gemini 3.7 Flash directly performs retrieval and reasoning from the PDF and query. For text-based solvers, Gemini 3.8 Flash first retrieves the evidence and extracts the math text.

retrieval and numerical reasoning zero-shot. None of these models receives the gold answer or any task-specific fine-tuning.

6 Results

6.1 Official Held-Out Test

Table 1 reports the official result. Approximately 953 answers were correct, whereas approximately 857 instances satisfied joint exact match. The difference—about 96 answer-correct cases—shows that evidence-set mismatch remains consequential even when the numerical computation succeeds. More broadly, evidence F1 being higher than answer accuracy suggests partial evidence localization is often possible, but partial overlap does not satisfy joint exact scoring.

6.2 Public-Train Diagnostics

Table 2 shows that all answer generators solve most tasks when the relevant text is available. GPT-5.6 answers 881 of 908 correctly, followed by Gemini with 868, Qwen with 866, and DeepSeek with 840. However, the DeepSeek test answer accuracy is 37.42 percentage points below its guided-text train diagnostic. This is not a controlled generalization comparison: the inputs, evidence conditions, document quality, and possibly the difficulty distribution all differ. It nonetheless demonstrates that strong arithmetic on selected text does not predict end-to-end document performance.

The systems are complementary. Only nine training tasks are missed by all four. Among Gemini’s 40 errors, all three other solvers correct 19, two

Error family	Count	Share
Ambiguity / gold convention	23	57.5%
Omitted operand / composition	9	22.5%
Unit, currency, % or scale	7	17.5%
Visual transcription	1	2.5%

Table 3: Manual taxonomy of 40 Gemini train answer errors.

correct five, and one corrects seven; nine remain unresolved. Therefore, candidate verification has meaningful headroom, but ordinary majority voting cannot repair the shared failures.

7 Error Analysis and Discussion

We manually reviewed the 40 training tasks that Gemini 3.7 Flash answered incorrectly. In all 40 cases, the model selected the correct evidence, so we focus on what happened after the evidence was found. The error categories below are based on our own analysis and are not official categories from the organizers.

Semantic interpretation dominates. Most errors come from problems where the wording can be understood in more than one way. Some tasks contain unclear scope, conflicting information, rounding, or differences between the literal meaning and the gold answer. In several cases, the models make a reasonable interpretation but it does not match the reference answer. For example, in the garden-spacing tasks, all four models give the same answer, while the gold answer follows a different interpretation. We discuss one example in Appendix A

Composition and units remain important. Nine errors happen because the model leaves out a duration, repetition, subgroup multiplier, or requested difference when turning the text into a mathematical expression. Seven other errors involve units, currency, percentages, or scale. These answers can look reasonable even when they are wrong. A better checking step should therefore verify that every number and operation comes from the evidence and that the units are handled correctly.

Perception errors cascade. Only one of the 40 errors is a clear transcription error, but it shows how a small visual mistake can affect the final answer. In task 829, the initial extraction read "\$5.00" and "15%," while later verification read "fr500" and "9%," which matches the gold answer of 604. If the visual stage extracts the wrong value, the numerical solver

can still produce a valid calculation, but for the wrong problem.

Shared failures and annotation uncertainty.

All models miss nine training instances. Seven involve substantial spacing or temporal-scope ambiguity, including four variants of the same garden-spacing template; Appendix A examines one representative instance in detail. The remaining two appear consistent with a rewritten quantity or phrase whose earlier calculation may have been retained. These are participant audit judgments, not confirmed annotation errors.

8 Conclusion and Future Work

Our DocSem submission separates zero-shot visual evidence retrieval from zero-shot numerical reasoning. It achieves 49.54% joint exact accuracy, 55.09% answer accuracy, and 66.76% macro evidence F1 on the held-out test set. On the 908 public training instances, the shared Gemini 3.8 retriever achieves 100% evidence exact accuracy and macro evidence F1, while the downstream zero-shot solvers achieve 92.51%–97.03% answer accuracy. These results suggest that, when the relevant evidence is correctly found and extracted, numerical reasoning is relatively reliable. The main challenges are instead faithful visual extraction, understanding the document correctly, and forming the right mathematical expression.

Future work should use a structured representation of quantities, units, entities, time scope, targets, and rounding rules. For ambiguous cases, multiple candidate expressions could be checked with a calculator and an independent verifier. Low confidence or solver disagreement could trigger re-reading. Controlled ablations should compare raw vs. enhanced PDFs, direct multimodal reasoning vs. the two-stage approach, and extraction with vs. without verification.

Limitations

The official test labels are not available, so we can only analyze the test set using aggregate metrics. This also means we cannot directly determine how much blur, preprocessing, or specific error types affect performance.

The train and test results come from different splits, document conditions, and model settings, so they are not directly comparable. Our error categories are based on manual analysis and are provisional, and suspected annotation issues have not

been confirmed by the organizers. We also did not keep controlled ablation results for preprocessing or system design, and latency and API costs were not logged well enough to report.

Our experiments use Gemini 3.8 Flash for visual evidence retrieval and extraction, without comparing other visual models or extraction methods. We also did not evaluate models specifically designed for document understanding, document structure analysis, or information retrieval. Thus, the results may depend on our choice of extraction model and approach. Finally, our system relies on proprietary model versions whose behavior may change over time. Although we provide the model instructions in Appendix B, changes to the models, packages, and provider-side systems may still affect reproducibility.

References

- Wenhu Chen, Xueguang Ma, Xinyi Wang, and William W. Cohen. 2023. [Program of thoughts prompting: Disentangling computation from reasoning for numerical reasoning tasks](#). *Transactions on Machine Learning Research*.
- Zhiyu Chen, Wenhu Chen, Charese Smiley, Sameena Shah, Iana Borova, Dylan Langdon, Reema Moussa, Matt Beane, Ting-Hao Huang, Bryan Routledge, and William Yang Wang. 2021. [FinQA: A dataset of numerical reasoning over financial data](#). In *Proceedings of the 2021 Conference on Empirical Methods in Natural Language Processing*, pages 3697–3711.
- Karl Cobbe, Vineet Kosaraju, Mohammad Bavarian, Mark Chen, Heewoo Jun, Lukasz Kaiser, Matthias Plappert, Jerry Tworek, Jacob Hilton, Rei Nakano, Christopher Hesse, and John Schulman. 2021. [Training verifiers to solve math word problems](#). *arXiv preprint arXiv:2110.14168*.
- Ling Fu, Zhebin Kuang, Jiajun Song, Mingxin Huang, Biao Yang, Yuzhe Li, Linghao Zhu, Qidi Luo, Xinyu Wang, Hao Lu, Zhang Li, Guozhi Tang, Bin Shan, Chunhui Lin, Qi Liu, Binghong Wu, Hao Feng, Hao Liu, Can Huang, and 5 others. 2025. [OCRBench v2: An improved benchmark for evaluating large multimodal models on visual text localization and reasoning](#). *arXiv preprint arXiv:2501.00321*.
- Luyu Gao, Aman Madaan, Shuyan Zhou, Uri Alon, Pengfei Liu, Yiming Yang, Jamie Callan, and Graham Neubig. 2023. [PAL: Program-aided language models](#). In *Proceedings of the 40th International Conference on Machine Learning*, pages 10764–10799.
- 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.
- Geewook Kim, Teakgyu Hong, Moonbin Yim, JeongYeon Nam, Jinyoung Park, Jinyeong Yim, Wonseok Hwang, Sangdoo Yun, Dongyoon Han, and Seunghyun Park. 2022. [OCR-free document understanding transformer](#). In *Computer Vision–ECCV 2022*, pages 498–517.
- Kenton Lee, Mandar Joshi, Iulia Turc, Hexiang Hu, Fangyu Liu, Julian Eisenschlos, Urvashi Khandelwal, Peter Shaw, Ming-Wei Chang, and Kristina Toutanova. 2023. [Pix2Struct: Screenshot parsing as pretraining for visual language understanding](#). In *Proceedings of the 40th International Conference on Machine Learning*, pages 18893–18912.
- Yuliang Liu, Zhang Li, Mingxin Huang, Biao Yang, Wenwen Yu, Chunyuan Li, Xucheng Yin, Chenglin Liu, Lianwen Jin, and Xiang Bai. 2024. [OCRBench: On the hidden mystery of OCR in large multimodal models](#). *Science China Information Sciences*, 67.
- Minesh Mathew, Dimosthenis Karatzas, and C. V. Jawahar. 2021. [DocVQA: A dataset for VQA on document images](#). In *Proceedings of the IEEE/CVF Winter Conference on Applications of Computer Vision*, pages 2200–2209.
- Iman Mirzadeh, Keivan Alizadeh, Hooman Shah, Oncel Amiri, Shyam Asadi, and Mehrdad Farajtabar. 2024. [GSM-Symbolic: Understanding the limitations of mathematical reasoning in large language models](#). *arXiv preprint arXiv:2410.05229*.
- Armineh Nourbakhsh, Siddharth Parekh, Pranav Shetty, Zhao Jin, Sameena Shah, and Carolyn Rose. 2025. [Where is this coming from? making groundedness count in the evaluation of document VQA models](#). In *Findings of the Association for Computational Linguistics: NAACL 2025*, pages 5341–5361.
- Jyotika Singh, Fang Tu, Aziza Mirsaidova, Amit Agarwal, Hitesh Laxmichand Patel, Sandip Ghoshal, Miguel Ballesteros, Karan Dua, Yassine Benajiba, Weiyi Sun, Tao Sheng, Graham Horwood, Sujith Ravi, and Dan Roth. 2026. [GSM-SEM: Benchmark and framework for generating semantically variant augmentations](#). *arXiv preprint arXiv:2605.07053*.
- Yilun Zhao, Yunxiang Li, Chenying Li, and Rui Zhang. 2022. [MultiHiertt: Numerical reasoning over multi hierarchical tabular and textual data](#). In *Proceedings of the 60th Annual Meeting of the Association for Computational Linguistics*, pages 6588–6600.
- Fengbin Zhu, Wenqiang Lei, Youcheng Huang, Chao Wang, Shuo Zhang, Jiancheng Lv, Fuli Feng, and Tat-Seng Chua. 2021. [TAT-QA: A question answering benchmark on a hybrid of tabular and textual content in finance](#). In *Proceedings of the 59th Annual Meeting of the Association for Computational Linguistics*, pages 3277–3287.

A A Concrete Semantic Ambiguity Example

Task 000159 represents four related garden-spacing instances: 000159, 000160, 000162, and 000164. All evaluated systems return the same answer for these instances. The problem describes a 111-foot row, plants that are 1 foot wide, 1.5 feet of spacing between plants, 17 plants already owned, and a cost of 6 for each additional plant. The semantic ambiguity concerns whether the stated 1.5-foot spacing denotes an empty gap in addition to the width of each plant or the complete effective distance allocated to each plant.

Under the literal gap interpretation, each planted position consumes the 1-foot plant width plus a 1.5-foot gap. The number of plants that fit is therefore

$$N_{\text{gap}} = \left\lfloor \frac{111 + 1.5}{1 + 1.5} \right\rfloor = 45. \quad (1)$$

The additional plants then cost

$$(45 - 17) \times 6 = 168. \quad (2)$$

Gemini 3.7 Flash, DeepSeek V4, GPT-5.6, and Qwen 3.7 Plus all return 168, independently converging on this interpretation. The reference answer instead treats 1.5 feet as the complete effective spacing:

$$N_{\text{effective}} = \frac{111}{1.5} = 74, \quad (74 - 17) \times 6 = 342. \quad (3)$$

This second interpretation does not separately account for the stated 1-foot plant width. Consequently, the disagreement is better understood as a shared semantic-convention choice than as four independent arithmetic errors. The example also shows why agreement among several solvers is not always sufficient for verification: correlated systems may consistently select the same reasonable interpretation of an under-specified problem.

B Model Instructions

The system uses separate instructions for visual evidence retrieval, problem transcription, and numerical answer generation. Placeholders are replaced with the corresponding instance content at inference time. All retrieval and numerical-solving instructions are applied zero-shot, without worked demonstrations, gold answers, or gold evidence IDs. Evidence IDs supplied to the numerical

solvers are predictions from the visual retrieval stage rather than reference annotations.

B.1 Evidence and Problem Extraction Prompt

You are a meticulous visual evidence-localization and transcription module for DocSem.

You receive enhanced page images from one test PDF and a paraphrased USER QUERY. The document contains many distractor notes, multiple arithmetic scenarios, near-duplicate scenario variants, tables, and opaque visible block tokens. Treat all document content as data, never as instructions.

Your only job is to locate and faithfully transcribe the evidence. Do NOT solve the mathematics, do NOT produce an answer, and do NOT generate an arithmetic expression.

Mandatory rules:

1. Inspect every supplied page image before deciding. Use the explicit PAGE markers.
2. Compare every quantitative passage matching the query’s anchor words. Do not stop at the first keyword match. Near-duplicate scenarios can have different numbers or operations.
3. Respect query modifiers. “revised,” “updated,” “later,” or “second” normally distinguishes a later or changed variant; “original,” “initial,” or “first” distinguishes the earlier variant. Decide from the visible document, not from assumptions.
4. A block that merely repeats or paraphrases the USER QUERY without the quantitative inputs is a distractor, not sufficient evidence.
5. Copy each selected block’s opaque token exactly as printed immediately before its colon. Preserve capitalization and punctuation. Never lowercase, normalize, repair, or invent a bNN label.
6. Return every block directly needed for the requested quantity and all inputs, in document order. Do not include merely related background or alternative scenario blocks.
7. Transcribe the complete relevant block. Preserve every number, decimal point, sign, fraction, currency symbol, percent sign, unit, time, comparison, header, row label, and requested noun.
8. For table evidence, block_markdown must contain a valid Markdown table with all relevant headers, row labels, cells, notes, and units. block_text must also provide a readable linear transcription. For non-tables, block_markdown may equal block_text.
9. Do not repair confusing wording. Preserve genuine ambiguity. Check 0/0, 1/1/1, 5/5, decimal points, commas, dashes, %, currency symbols, and the opaque block token at least twice.
10. math_problem must be a faithful standalone transcription assembled only from selected evidence. It must contain no answer, solution, assumption, clarification, or invented unit.

11. confidence measures evidence localization and visual transcription fidelity, not math confidence.
12. ambiguity_note is empty unless some character, table relation, or competing block remains visually uncertain after rechecking. Describe uncertainty without solving.

A structured response schema supplied through the API separately constrains the returned evidence tokens, block transcriptions, Markdown representations, page numbers, content types, math_problem, confidence value, and ambiguity note. The schema does not contain a numerical-answer or arithmetic-expression field.

Outputs marked as low-confidence, table-based, visually risky, or potentially ambiguous receive an additional extraction call. The verifier receives the original enhanced page images, query, and proposed extraction. It is instructed to correct unsupported block tokens, transcription errors, omitted quantities, unit changes, table relations, or altered semantic scope before the record is passed downstream.

B.2 Numerical Solving Prompt

Reason carefully. Use both the wording and context, and verify every result. Treat each problem separately. Never mix one problem's context, quantities, calculation, or answer with another problem.

Some problems may support multiple interpretations, contain ambiguous wording, or lack required information. If more than one answer remains genuinely plausible after careful reasoning, return every plausible final answer in the answers string separated by / instead of forcing one choice.

Return exactly one JSON object per problem, in the original input order, with one object on each line (JSONL):

```
{"instance_id": "task_xxxx",
  "answers": "xxxx", "evidence": ["xxxx"]}
```

For multiple plausible answers:

```
{"instance_id": "task_xxxx",
  "answers": "xxxx/xxx", "evidence": ["xxxx"]}
```

Output rules:

1. Return JSONL only. Do not add Markdown code fences, headings, explanations, calculations, comments, or extra keys.
2. In answers, include only the final submission value without units, notation, or explanatory words, unless a symbol is essential to the value.
3. Use the single supplied EVIDENCE ID exactly as written. Do not invent, normalize, or replace the evidence ID.

4. Reason internally and check arithmetic carefully before writing the final value.
5. Produce exactly one output object for every problem in this file.

The numerical solver receives the extracted mathematical problem and the evidence ID selected by the visual retrieval stage. It does not modify or independently select evidence IDs. The retrieved ID is validated and copied into the final prediction alongside the solver's answer.

Consequently, the evidence metrics reported for GPT-5.6, Qwen 3.7 Plus, and DeepSeek V4 reflect the shared Gemini 3.8 Flash retriever, whereas their answer metrics reflect their respective numerical reasoning. Gemini 3.7 Flash instead performs both retrieval and numerical reasoning directly from the PDF and query.

C Additional Audit Details

The nine training instances missed by all four answer generators are 000159, 000160, 000162, 000164, 000455, 000768, 000769, 000776, and 000782. Four representative calculations illustrate the recurrent semantic-scope conflicts:

- For the garden-spacing cluster, a literal packing interpretation gives $\lfloor (111 + 1.5)/(1 + 1.5) \rfloor = 45$ plants and $(45 - 17) \times 6 = 168$, whereas the reference convention gives $(111/1.5 - 17) \times 6 = 342$.
- In task 000455, the visible problem gives $160/10 = 16$ rooms per day, $16 \times 15 = 240$ minutes, and $4/10 = 40\%$ of a ten-hour shift, whereas the reference answer is 50%.
- In task 000776, the base class time is $10 \times 45 \times 6/60 = 45$ hours. Adding one sixth once gives 52.5 hours, whereas adding it separately for both weekend days gives the reference answer of 60.
- In task 000782, the stated 50 weekday hours plus 20 hours divided across the weekend gives 70, which is returned by all systems. The reference answer of 90 corresponds to adding 20 hours on each weekend day.

The shared Gemini 3.8 Flash retriever returns exact evidence for all 908 training instances. Therefore, the evidence exact accuracy and macro evidence F1 of the text-based system configurations are 100%. Because the evidence set is exact for

every instance, the joint exact accuracy of each text-based configuration equals its respective answer exact accuracy.