

Analysis of Reasoning in Visual Mathematical Problems

Adria Aguilar Mínguez, Jorge Aranda Honrubia, Santiago Font Piña, Pablo Gandía Míñana, Pablo Gil Martínez, Carlota Gui Escamilla

1 Introduction

While multimodal language models (MLLMs) have present an exceptional performance in text-based reasoning, visual mathematical problems remain challenging, especially when the solution depends upon an accurate interpretation of images. The main limitation is the documented disconnection between visual perception and symbolic reasoning. Current models often function as sophisticated text transcribers, successfully recognizing statements within images but overlooking critical spatial data or geometric relationships (see Figure 12 in Annex A.2, where examples of both types of problems are shown to better understand the situation: cases where the image is not needed and cases where visual information is essential).

The limitations of these models are examined in the reference paper *Evaluating Visual Mathematics in Multimodal LLMs: A Multilingual Benchmark Based on the Kangaroo Tests*[1]. This study evaluates visual language models on Kangaroo mathematics problems (multiple-choice contests for secondary-school students covering areas such as logic, geometry, algebra, and patterns). The benchmark concludes that performance drops significantly on mathematically simple problems when the solution requires a visual understanding of the accompanying figure, such as interpreting perspective, proportions, or spatial distributions.

As an additional point of comparison, a similar behaviour was observed in *LLM performance on mathematical reasoning in Catalan language*[2]. This study has a close relationship to the main reference paper.

Benchmark contamination must be considered, but it is not used here as the main explanation because the reference paper reports similar behaviour on recent unseen problems, so our analysis focuses on the visual/non-visual gap, option-level instability, and reasoning coherence.

Model	Parameters	Company	CoT (Type)	Reference Results (%)
Pixtral-12B	12B	Mistral AI	Weak emergent	24.5 / 27.8
Qwen-VL 2.5 7B	7B	Alibaba	Implicit multimodal	32.5 / 47.3
LLaMA 3.2 Vision 11B	11B	Meta AI	Prompt-based	14.0 / 15.8
LLaMA 3.2 Vision 90B	90B	Meta AI	Prompt-based*	15.0 / 16.5
Pixtral-Large	48B (est.)	Mistral AI	Weak emergent	29.2 / 39.3
GPT-4o	Unknown	OpenAI	Implicit (latent)	40.2 / 65.3
Gemini 2.0 Flash	Unknown	Google DeepMind	Enhanced implicit	45.4 / 75.9
Gemini 2.0 Flash Lite	Unknown	Google DeepMind	Enhanced implicit	39.4 / 66.5
Qwen-VL 2.5 72B	72B	Alibaba	Advanced implicit	43.5 / 70.6

Table 1: Benchmark results reported in the reference paper.

Note. *Reference Results (%)* column reproduces the results reported in the reference paper: the left value corresponds to purely visual problems, and the right value to problems that do not require visual reasoning. The *CoT (Type)* column indicates the Chain of Thought reasoning mechanism used by each model (see Annex A.3 for a detailed breakdown). "*Gemini 2.0 Flash*", which achieves the best performance, features enhanced implicit CoT, while prompt-based models like "*Llama 3.2*" require explicit prompting to reason. *Scale-enhanced.

The Kangaroo benchmark introduced an evaluation methodology primarily based on accuracy using simple prompting techniques. However, it did not deeply analyze the underlying reasoning traces or apply advanced inference techniques. Consequently, our project aims to extend this research in three directions:

- **Reasoning and Error Analysis:** We shift the focus from simple metrics to identifying recurring error patterns through a strategic subsampling of critical problems.
- **Modular Pipeline Optimization:** We explore and implement advanced pipelines aimed at improving visual reasoning and outperforming the reference benchmark.
- **Educational Application:** We develop a prototype tool which uses these optimized pipelines to provide students with transparent, step-by-step guidance for solving visual math problems.

2 Data Description

The data preparation process was based on the *official repository*[3] associated reference study. The relevant information was not provided as a single analysis-ready file, but rather distributed across several files and formats. Consequently, the first step was to construct a unified working dataset that could support benchmark replication, descriptive analysis, and subsequent inspection of model reasoning.

Original dataset structure. The *base dataset*[4] contained, for each problem, the statement, the multiple-choice answer options, the correct answer, and a link to the segmented image of the problem. This segmented image corresponds to the input presented to the model: it may include the statement, the figure when present, and the answer options. This structure is central to the analysis because it makes it possible to distinguish between problems that can be solved from textual information alone and problems for which correct interpretation of the image is necessary.

In addition to the base file, the repository included separate files containing the final answers and the reasoning traces produced by each evaluated model. These sources were merged by the problem index to produce a single working table. The resulting dataset aligns these variables at the row level, as detailed in Table 2.

Information block	Variables included
Problem content	Problem index, statement, multiple-choice answer options, correct (ground-truth) answer, topic, language, year, and school level.
Visual component	Visual dependence indicator (text-only vs. image-necessary), problem figure, and link to the segmented image.
Model outputs	Final answers and outputs produced by each evaluated model.
Reasoning traces	Explanations and reasoning traces generated by each model to justify its answer.

Table 2: Information integrated into the working dataset.

Experiment replication. After constructing the working dataset, the principal experiments of the reference study were replicated in order to verify whether the observed behaviour was consistent with the results reported by the authors. The numerical values are not identical, but the qualitative pattern is preserved: all evaluated models exhibit substantially lower performance when the problem requires visual interpretation.

Model	Overall	With img.	Without img.	No Answer
Gemini 2.0 Flash	60.10	40.98	74.50	13.30
Qwen-VL 72B	58.40	41.44	71.17	0.05
GPT-4o	52.45	35.86	64.94	6.30
Gemini 2.0 Flash Lite	51.80	35.74	63.89	14.65
Qwen-VL 7B	40.60	30.27	48.38	1.00
Pixtral Large	35.15	27.24	41.10	14.95
Pixtral-12B	25.45	22.58	27.61	10.85
Llama 3.2 90B Vision Instruct	16.15	15.02	17.00	21.80
Llama 3.2 11B Vision Instruct	10.65	8.27	12.45	34.65

Table 3: Global metrics of the replicated experiment on the working dataset.

Cleaning and exploratory analysis. Once the information has been unified and the experiment replicated, the exploratory analysis was used to extract insights from the data and assess whether the visual-performance gap reflected a genuine multimodal limitation or merely a consequence of the dataset composition. The process included consistency checks, normalization of labels such as difficulty levels across languages, and the study of distributions by topic, language, school level, and visual-information requirement.

The resulting working data set contains 2,000 questions, of which 863 require the image and 1,137 do not. It also covers four languages: Catalan, Spanish, French, and English. This composition is sufficiently broad to compare visual and non-visual cases, but the aggregate counts alone do not explain why models fail so sharply on the visual subset.

The first relevant check was whether image-dependent problems were simply longer or assigned to more

advanced school levels. The data do not support that interpretation. The median statement length is almost identical in both groups: 37 words for visual questions and 35 words for non-visual questions. Moreover, as shown in Figure 1, the lowest school levels are precisely those with a higher concentration of image-dependent problems. The degradation observed in model performance is therefore not adequately explained by longer statements or more advanced problems. It is better understood as a failure to use visual information reliably. This point is central to the project: problems that are elementary for a human student may become difficult for a model when the solution depends on shapes, spatial positions, patterns, or relations between visual elements.

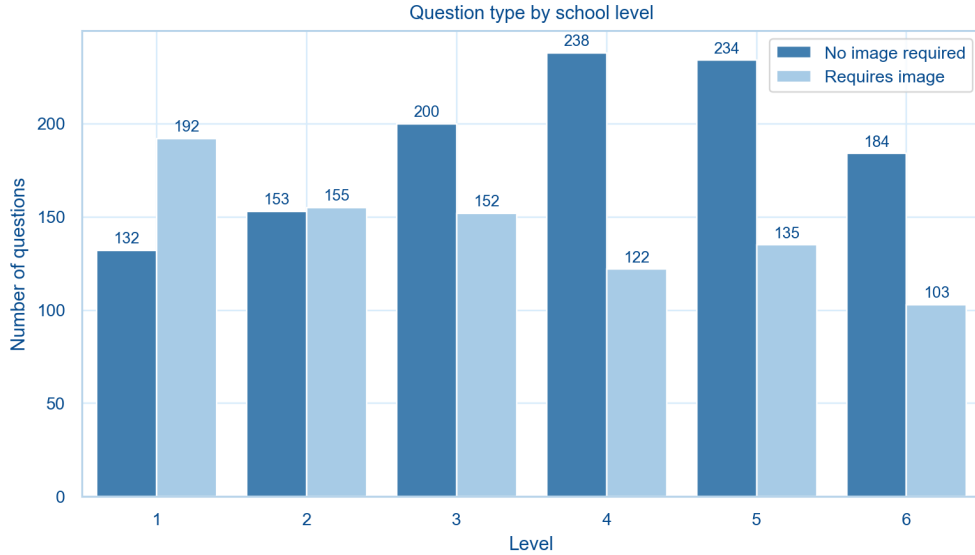


Figure 1: Questions with and without visual dependence by school level.

This interpretation becomes stronger when the visual subset is examined by topic. The lowest mean performance appears in Logic and Visual Reasoning and in Patterns and Sequences, as shown in Figure 2. These categories often require identifying transformations, spatial relations, repeated structures, or subtle differences between answer options. The errors are therefore not exclusively arithmetic or symbolic; they frequently originate before the mathematical reasoning stage, at the point where the visual structure of the problem has to be understood.

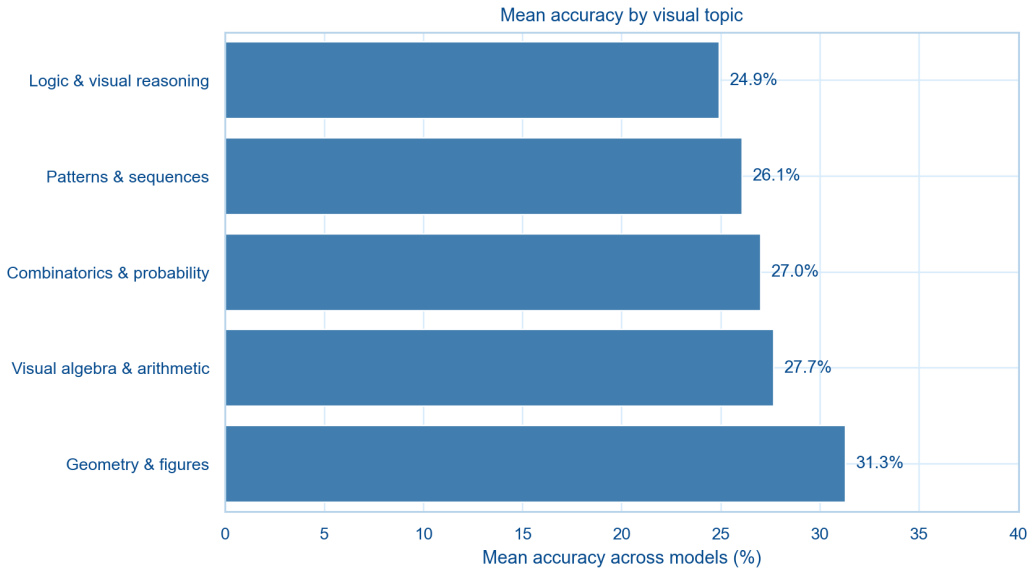


Figure 2: Mean difficulty by visual topic.

The exploratory analysis was then extended from answers to reasoning traces. This additional layer is important because a model can produce a long and fluent explanation even when its visual interpretation is

wrong. All nine reasoning columns in the *dataset*[4] contain non-empty traces for the 2,000 questions in the working set. Table 4 shows that verbosity varies substantially across models, but this variation does not follow the same ordering as benchmark accuracy. "*Pixtral-12B*" and "**Qwen-VL 7B**" produce the longest explanations on average, whereas stronger models such as "**Qwen-VL 72B**" and "**GPT-4o**" remain in a more moderate range.

Table 4: Length of model reasoning traces in the working dataset.

Model	Mean words	Median words	Std. dev.
Pixtral-12B	204.5	167.0	540.6
Qwen-VL 7B	203.7	169.0	262.6
Gemini 2.0 Flash	200.2	133.0	183.8
Llama 3.2 90B Vision Instruct	198.0	172.0	234.2
Pixtral Large	195.9	190.0	84.3
GPT-4o	172.7	166.5	66.3
Gemini 2.0 Flash Lite	172.3	120.5	187.0
Qwen-VL 72B	167.2	153.0	84.2
Llama 3.2 11B Vision Instruct	165.8	153.0	104.2

This indicates that longer explanations are not, by themselves, evidence of better reasoning. The standard deviation also shows that some models, especially "*Pixtral-12B*", produce highly uneven reasoning traces, combining ordinary-length explanations with a small number of very long outputs. After normalizing frequent terms across languages, the most recurrent words are centred on generic mathematical and geometric content, such as "*number*", "*square*", "*triangle*", "*total*", "*area*", "*sum*", "*angle*", and "*side*". The relevant question is therefore not whether the models can produce step-by-step reasoning, but whether that reasoning is grounded in the visual evidence required by the problem.

Overall, the data preparation process provides a rigorous basis for the remainder of the project. It reconstructs the benchmark in a form suitable for analysis, verifies that the replicated experiment preserves the central pattern of the reference study, and shows that the main limitation to be addressed is not general mathematical difficulty, but, as stated before, the interaction between visual perception, mathematical reasoning, and the quality of the explanations generated by the models.

3 Methodology

Development of the Study For the development of this study, a working methodology appropriate to the context of the problem has been defined. Since this is a scientific research project, as discussed in Unit 2, the approach is structured around the scientific method: formulating a hypothesis, testing it through controlled experiments, analysing the results, and iterating to refine conclusions.

In this case, the initial observation was the behavior of the models when solving image-based mathematical problems. The two initial broad questions, why this phenomenon occurs and whether the results can be improved, were refined into the three operational research questions below. The proposed hypothesis is that the models fail on image-based tasks because they struggle with specific subtasks rather than with all tasks in general, and that if these problematic tasks can be identified, the models' performance could be improved through an appropriate pipeline.

The final analysis is guided by three operational research questions:

- **RQ1.** Which visual and reasoning failures explain the visual-performance gap?
- **RQ2.** Can a segmentation-based, option-level pipeline improve over a simple multimodal prompt?
- **RQ3.** To what extent can an inspectable MVP provide useful educational feedback and research insight despite imperfect model performance?

These questions guide the structure of the document: RQ1 is addressed in the Reasoning Analysis, RQ2 in the visual solving pipeline, and RQ3 in the App and Future and Legacy sections.

Workflow As discussed in Unit 1 of the seminars, an agile methodology (Scrum) was adopted to manage the project and ensure continuous progress within the established deadlines.

The work was organized into two parallel workstreams designed to converge in the final pipeline. The first workstream was the Reasoning Analysis, a qualitative study focused on identifying and cataloging recurring error patterns. The second workstream focused on technical optimization, testing advanced inference techniques such as “Axiom Prompting” and “Chain of Thought” to determine whether these weaknesses could be reduced through more sophisticated processing.

Subsampling Since carrying out these experiments on the entire dataset would have required excessive computational time we decided to do a subsampling of the dataset using the most representative samples, specifically the image-based questions that most models failed to solve, since these are the reasoning patterns of greatest interest for this study. Some random samples were also added to make the subset more representative and realistic. This subsampling was then used to conduct iterative experiments, as discussed in Unit 4, in order to identify the best-performing pipeline. The same concept was applied to the Reasoning Analysis, where seventy samples were manually evaluated. This process will be described in greater detail in the corresponding Reasoning Analysis section.

Tools To ensure that the experiments were reproducible while also managing costs, a local model was used instead of commercial APIs, as these were too expensive. “Ollama” and LangChain were employed to run models such as “Qwen-VL 7B”, “Llama 3.2 visual 11b” and “Qwen-3 4B”. The computational cost was handled by the UPV GPU Cluster, which significantly accelerated the inference time and provided greater control over the models themselves. The main tools used were Python for inference and the application backend, FastAPI for local endpoints, and Florence combined with OpenCV for PaddleOCR and image processing.

Use of AI tools Nano Banana 2 generated explanatory images, such as Figure 10, and Codex supported the app’s frontend development. All generated content and outcomes were supervised to ensure there were no significant hallucinations. Furthermore, specialized research skills (Codex) designed by our team were applied throughout the process to personalize performance and ensure the integrity of the results.

4 Reasoning Analysis

As shown in previous sections, language models obtain considerably lower results when solving mathematical problems requires interpreting visual information. Although current models show great performance in textual tasks, the results evidence significant limitations in multimodal contexts.

For this reason, this section will conduct a detailed analysis of the process followed by the models when working with mathematical images. The main objective is to identify at which stages errors appear and which types of tasks present the greatest difficulty. In this way, it will be studied whether the limitations stem from the visual interpretation of the image, the subsequent mathematical reasoning, or a combination of both processes. Furthermore, this analysis will allow for the detection of failure patterns and a better understanding of the real capabilities of these systems in problems that require integrating visual and textual information.

Understanding how these models operate is especially relevant due to the scarce explainability that characterizes many current artificial intelligence systems, frequently considered “black boxes.” Analyzing their behavior allows for increasing model transparency, identifying their limitations, and developing more reliable and secure systems. This issue is especially important in fields such as education, research, or the automation of complex tasks, where knowing the origin of errors is fundamental. Likewise, this type of study relates to the objectives of a more responsible artificial intelligence aligned with United Nations initiatives, especially Sustainable Development Goal 4, focused on quality and accessible education through the appropriate use of new technologies.

Even though, in the reference paper this phenomenon is studied across a broad set of models, this work will only perform a detailed analysis of two of them, while still taking the rest into account. The selected models were “Qwen-VL 2.5 7B,” due to its good results and accessibility, and “Gemini 2.0 Flash Lite,” as it was the best-performing model and the one that possibly came closest to more complex systems such as “GPT-5.5” or “Gemini 3.1 PRO.” Other accessible models, such as “LLaMA 3.2 Vision 11B,” were also evaluated, but their results ultimately led to discarding this option.

Before performing the manual qualitative analysis of the responses generated by the models, an unsupervised analysis approach based on clustering techniques was initially proposed. The objective was to automatically identify possible resolution patterns, error types, or strategies used by the models when solving visual mathematical

problems.

To do this, textual descriptions of the problems were generated using "*Qwen-VL 2.5 7B*," and subsequently, semantic embeddings of the responses produced by the models were obtained. Different clustering methods were applied to these embeddings to detect similar behaviors between responses.

However, the results showed that the generated clusters were organized mainly according to the theme of the exercises, for example, geometry, probability, or logic, instead of reflecting real resolution patterns or error types. This made it difficult to extract relevant conclusions about the internal behavior of the models, as the grouping remained dominated by the semantic content of the problems and not by the way the models developed their inferences.

Due to these limitations, it was finally decided to perform a manual qualitative analysis. To this end, a significant sample of mathematical problems was studied in which the majority of the models failed, specifically those in which more than 7 of the 9 models committed errors.

The purpose of this analysis was to identify patterns in the visual characteristics required by each problem, for example, reasoning about proportions, overlapping figures, or counting tasks. Additionally, it also sought to detect patterns in the errors present in the responses generated by the models, such as calculation failures, errors in figure detection, or incorrect spatial interpretations.

The study of visual characteristics aimed to understand which capabilities the models failed in most frequently. On the other hand, the error analysis allowed for both studying the limitations of the models and subsequently developing the functionality of a judge, which will be described later.

To find these patterns, 60 mathematical problems were manually analyzed along with the responses generated by "*Qwen-VL 2.5 7B*" and "*Gemini 2.0 Flash Lite*." Because the selected problems corresponded to those with the highest error rates, the responses produced by both systems were mostly incorrect. During the analysis, each problem was assigned a description of the information that had to be extracted from the image and two summaries corresponding to the errors present in the responses of both models.

Once the 60 problems were analyzed, categories were defined to classify both the required visual capabilities and the errors present in the responses. Initially, 20 labels were proposed for each group through an iterative manual review process. Subsequently, several labels were simplified and grouped due to their semantic overlap, finally reducing the set to 10 categories for each type. The categories of visual capabilities can be seen in the Annex B.1.1 and the categories of reasoning errors can be seen in the Annex B.1.2.

Once the labels were defined, inference was performed using few-shot prompting with "*Qwen-VL 2.5 7B*" on the entire database containing visual information. From this inference, it was possible to extract several relevant conclusions. The dataset and the specific prompts can be seen in the GitHub repository. A similar process can be observed in the paper *Large Language Models as Annotators: Enhancing Generalization of NLP Models at Minimal Cost*[16]. This paper supports the use of LLMs to scale annotation, while our results also show why this strategy must be checked for judge bias and over-diagnosis.

In the Figure 3, it can be observed that both Qwen and Gemini present significant differences in their performance depending on the capabilities required by each problem. Although both models obtain good results in classical geometry categories, such as calculating angles, lengths, and areas, their performance decreases considerably in spatial reasoning tasks, especially in rotations, folding, and grid reasoning. This indicates that the models are more effective at applying explicit mathematical formulas and rules than at mentally manipulating objects in space or understanding complex three-dimensional relationships.

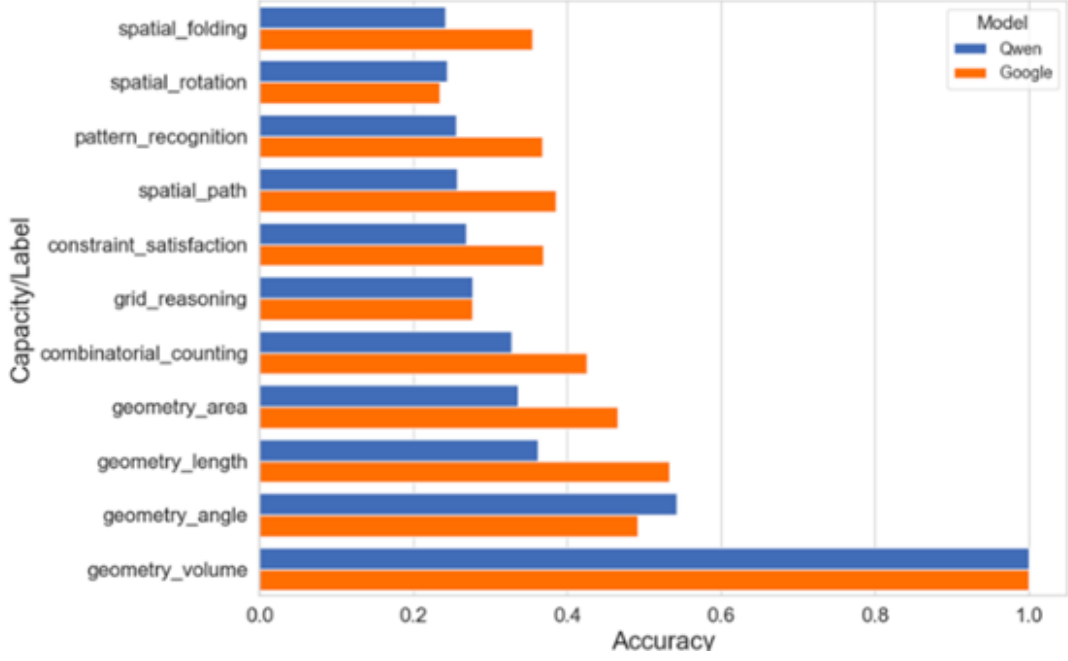


Figure 3: Visual Capability

In both models, the *"geometry_volume"* category reaches 100% accuracy; however, this result is not representative because it only contains one sample. More relevant are the low success rates observed in spatial capabilities. *"Qwen"* presents especially poor results in *"spatial_folding," "spatial_rotation,"* and *"pattern_recognition,"* while Gemini also shows significant weaknesses in *"spatial_rotation"* and *"grid_reasoning."* These limitations suggest that current architectures still lack deep visual and spatial understanding.

The comparison between models evidences that Gemini outperforms *"Qwen"* in most categories, especially in geometry and combinatorial counting. However, *"Qwen"* obtains better results in *"geometry_angle,"* and both models show very similar performance in complex spatial tasks, pointing to a common limitation in this type of reasoning.

Altogether, the results suggest that although current multimodal models have achieved notable performance in structured mathematical tasks, they still present significant limitations in problems that require complex spatial manipulation or deep visual understanding. In order to see more about the description analysis watch Annex B.3.

On the other hand, the study of errors present in the responses was divided into two main parts. In the first, taking advantage of the fact that both the inference and the detection of errors in the responses of *"Qwen"* and *"Gemini"* had been performed using *"Qwen"*, the efficacy of using a model as a judge to determine whether or not an error existed in the generated response was analyzed. In the second part, those cases that the judge correctly classified as reasoning errors were studied to identify what those failures consisted of.

As can be observed in the Figure 4 and the Figure 5, the analysis of *"Qwen-VL 2.5 7B"* as an external judge reveals that the model is effective at detecting logical failures but is excessively inflexible. When evaluating *"Gemini"*'s responses, it managed to identify the majority of errors, although it generated a high rate of false positives by rejecting correct but unconventional resolution processes. This demonstrates that the model has a strong tendency to over-diagnose errors when analyzing inferences generated by other systems.

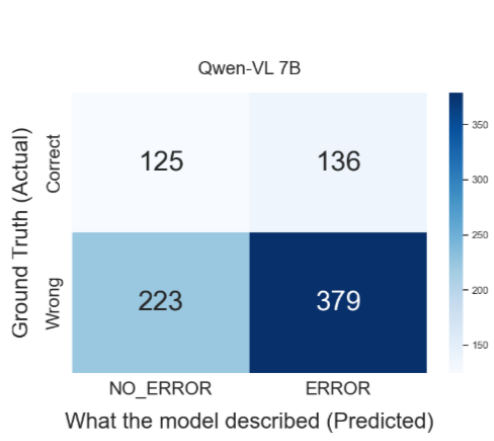


Figure 4: No Error Analysis: Qwen-VL 7B

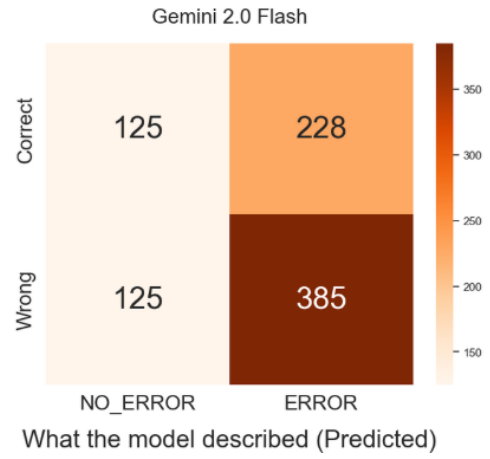


Figure 5: No Error Analysis: Gemini 2.0 Flash

On the contrary, when "*Qwen*" acted as a judge of its own responses, its accuracy decreased drastically due to an evident self-validation bias. The model overlooked its own errors almost three times more than when evaluating "*Gemini*"'s responses, which evidences that artificial intelligences are not reliable when analyzing their own performance. Therefore, it is concluded that in any automated system, it is essential to use an external and independent model to audit the generated results.

These results suggest that the choice of the judge model depends heavily on the application context. Using the model itself as an evaluator can cause numerous errors to go unnoticed, while employing an external model increases the probability of incorrectly classifying valid but unusual responses as erroneous. If more information about the error reasoning analysis see Annex B.4.

The analysis of the Figure 6 and the Figure 7 reveals that the main obstacle for both models is the inability to count with precision, an error that reaches its critical point in pattern recognition tasks, with 97 failures in "*Gemini*" and 124 in "*Qwen*". Although the models manage to identify complex logical rules, they fail in the quantification of elements, especially in constraint satisfaction tasks, where both exceed 89 errors. This evidences the existence of a perceptual "bottleneck": the models understand the general logic of the problem but fail to correctly measure the visual components necessary to solve it.

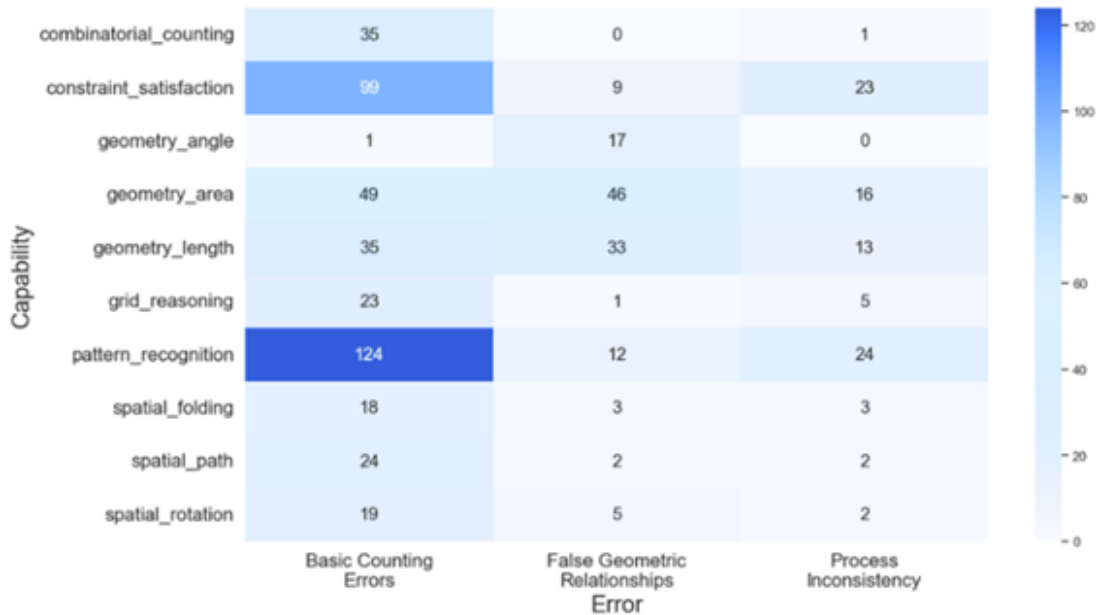


Figure 6: Visual Capability Vs Reasoning Error: Qwen-VL 7B

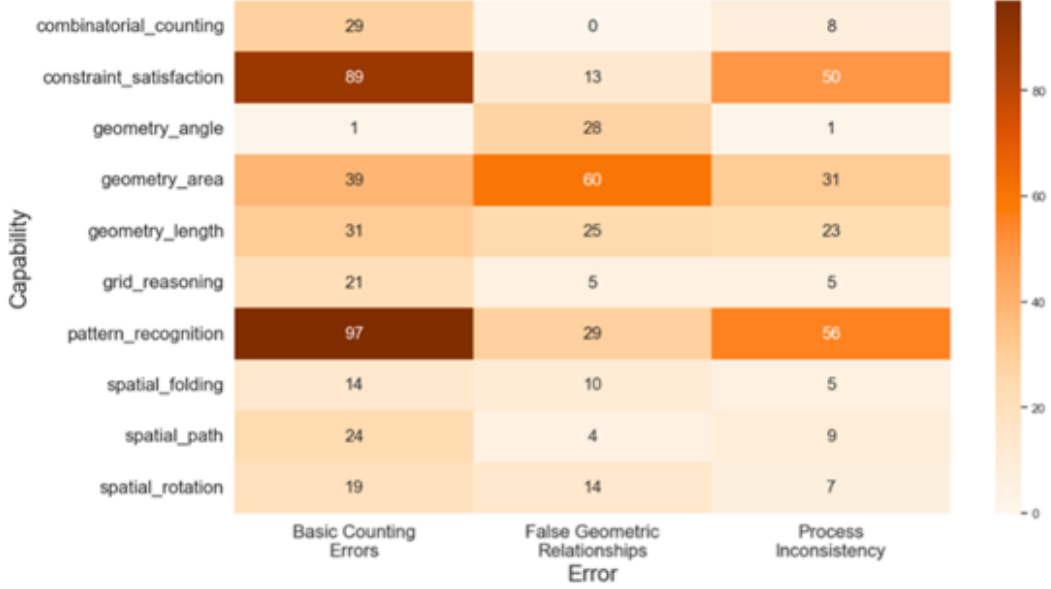


Figure 7: Visual Capability Vs Reasoning Error: Gemini 2.0 Flash

In the area of geometry and spatial reasoning, the models tend to invent non-existent relationships, an error that stands out especially in the calculation of areas, with 60 failures in Gemini and 46 in *"Qwen"*. While *"Gemini"* shows a tendency to overcomplicate solutions, *"Qwen"* maintains a high rate of counting errors even in basic geometry problems. Additionally, spatial folding and rotation tasks trigger specific failures confirming that three-dimensional reasoning constitutes a distinct challenge, processed with much less fluidity than two-dimensional logic by both systems.

Finally, a loss of coherence is also observed in high-complexity situations, evidenced in the *"process inconsistency"* category, whose errors increase considerably in pattern and constraint satisfaction tasks, reaching 56 and 50 errors in Gemini, respectively. Under multiple simultaneous rules and constraints, the models lose the thread of their own inferential process or generate internal contradictions, especially when the cognitive load increases. Consequently, the deterioration in performance seems to be due more to perceptual limitations and saturation derived from managing too much information simultaneously than to the theoretical difficulty of the mathematical problems.

In conclusion, the study evidences that the limitations of multimodal models in mathematics do not lie primarily in mathematical theory, but in a perceptual and spatial "bottleneck." While systems such as *"Gemini 2.0 Flash Lite"* and *"Qwen-VL 2.5 7B"* show proficiency in structured geometry tasks and the application of formulas, they systematically fail in the mental manipulation of objects, especially in rotations, folding, and visual counting tasks. These results suggest that current architectures, although logically advanced, still lack deep three-dimensional understanding and tend to generate inconsistencies in their inferential processes when the visual load and the complexity of constraints saturate their processing capacity.

Likewise, the analysis of explainability reveals that the use of artificial intelligence models as judges of their own responses is unfeasible due to a marked self-validation bias, through which systems overlook their own errors with a significantly higher frequency than when evaluating external models. Conversely, the use of external judges, although more effective for detecting logical failures, tends toward greater inflexibility and over-diagnosing false positives. Therefore, it is concluded that, to guarantee reliability and transparency in critical areas such as education, it is essential to implement audits through independent external models, recognizing that basic visual perception remains the weakest link in the resolution of complex visual mathematical problems.

5 Improvement of the visual solving pipeline

As observed in the reference experiment, the original methodology mainly consisted of sending a simple prompt together with a segmented screenshot containing only the target question. Starting from that baseline, we hypothesized that applying a more structured multimodal pipeline, instead of relying solely on a generic

image+prompt approach, could improve the benchmarks or, at least, provide useful insights to better characterize the behavior and explainability of current vision-language models when solving visual reasoning problems.

Previous replication and diagnosis Before designing the pipeline, the base experiment [5] was replicated with "*Qwen2.5-VL 7B*", using the same type of input and a prompt equivalent to the benchmark one. The replication was run on a sample close to 900 problems. The reference paper reported 40.52% accuracy on the full dataset and 40.77% on the comparable subset. Our local replication obtained 35.33%. The number was slightly lower, but the qualitative behavior was similar. The model solved some cases, although it failed repeatedly when the answer depended on comparing shapes, interpreting positions or imagining visual transformations.

Also, when reviewing the answers, an important point appeared. Although the aggregate results were similar to those in the paper, many specific answers did not match. The model could get right or wrong different cases even with an equivalent methodology. This suggested that part of the answers could have a random or unstable component. In several problems it seemed that the model chose an option and then generated a justification that fitted it, rather than reasoning consistently from the image.

Experiment	Model	Accuracy
Reference paper, full dataset	Qwen2.5-VL 7B	40.52%
Reference paper, comparable subset	Qwen2.5-VL 7B	40.77%
Local replication	Qwen2.5-VL 7B	35.33%

Table 5: Replication of the base experiment with Qwen2.5-VL 7B.

"*Llama 3.2 Vision 11B*" was also tested on a small sample. It was not evaluated on the full dataset because of time cost, but to observe the type of output it generated. The conclusion did not change. The models could transcribe the statement and describe the image, but they failed when the solution required using visual relations as a central part of the reasoning.

As a first attempt at improvement, a reasoning pipeline [6] was evaluated on 15 problems that "*Qwen2.5-VL 7B*" had failed. The idea was to split the solution into four phases. First, *visual grounding* was applied, where the model was asked to list the important visual elements without solving the question yet. Then *axiom prompting* [7] was applied, understood as a phase to extract useful rules, formulas or mathematical relations before trying to solve. The third phase used *chain of thought* [8], which asks the model to reason step by step and make intermediate steps explicit before giving the final answer (these reasoning papers are used as methodological inspiration, but our results show that making reasoning explicit does not guarantee correctness if the visual evidence is not integrated reliably). Finally, *self-refinement* was added, where the model reviewed its own solution and tried to correct inconsistencies. Finally, *self-refinement* was added, where the model reviewed its own solution and tried to correct inconsistencies.

This first pipeline worked as a pilot test. In that small sample, accuracy went from 13.30% with the simple prompt to 26.70% with the pipeline. Even so, there were only 15 problems, so it could not be presented as a representative improvement for the dataset. The most useful part was the diagnosis. The model worked well as OCR and as a descriptor. It was able to read the statement, recognize options and describe figures in a fairly convincing way. The problem appeared afterwards, when it had to integrate that visual information into the mathematical reasoning. Many times the description was correct, but it was not really used to decide. The image remained as an explanation added after the fact and not as the basis of the answer.

The manual review also showed a repeated pattern. The model chose an option and then built an explanation compatible with it, but it rarely justified why the other options were false. In visual multiple-choice problems, this lack of rejection is critical. If the system cannot reject the other options, the final explanation can look reasonable even if the decision is basically unstable.

Design of the final pipeline From that diagnosis, a more demanding pipeline was designed, implemented in the code folder of the repository [9]. "*Qwen2.5-VL 7B*" was kept as the main visual model because it was the local model with the best performance. The central idea was to reduce the noise of the full capture. In an unprocessed image, text, options, figures, separators and incorrect alternatives appear together. This makes it easier for the model to rely on the textual part and not analyze the visual information with enough precision.

The new pipeline aims to isolate the relevant figures and force the model to evaluate each option separately. The process starts by sending the full image to "*Qwen2.5-VL 7B*". The model classifies the problem in two

dimensions.

- **A1/A2.** The statement does not need an image or does need an image.
- **B1/B2.** The options are not images or the options are images.

Also, in this phase the text of the statement and the options is extracted when it is available. A smaller visual model was tested for this task, but "*Qwen2.5-VL 3B*" failed even on simple classifications, so "*Qwen2.5-VL 7B*" was kept.

If the problem is fully textual, case A1/B1, it is sent directly to "*Qwen3 4B*", used as a textual reasoner because it is faster and does not require extra visual processing. If there is any figure in the statement or the options are images, the segmentation phase is activated. First, PaddleOCR [10] is used, an OCR tool designed to detect and recognize text in images. In our case it is not used only to read the text, but also to obtain the boxes where the statement appears and where the options begin. These coordinates make it possible to apply spatial rules. For example, the statement is usually in the upper part of the capture and the options appear below. With this information, the system can discard noise when it detects statement text in lower areas or option text in upper areas. In the same way, the figure of option A should be close to the text block of option A.

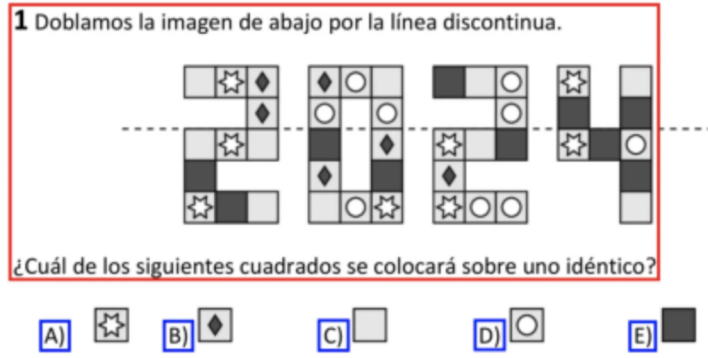


Figure 8: Example of text and box detection with PaddleOCR.

With these coordinates, Florence-2 [11] is used, a vision-language model from Microsoft that can perform detection, localization and visual understanding tasks from text instructions. In this part, Florence-2 base fine-tuned is used to locate the figures and return their coordinates. The goal is to obtain isolated images of the statement and of each visual option. Then, "*Qwen2.5-VL 7B*" validates the crops. If it detects that a region is badly segmented, that part is replaced by the full image of the problem to avoid losing necessary information.

Before fixing this configuration, different prompts and parameters for PaddleOCR and Florence-2 were tested on small sets, until finding the approach that segmented better without introducing too much noise.

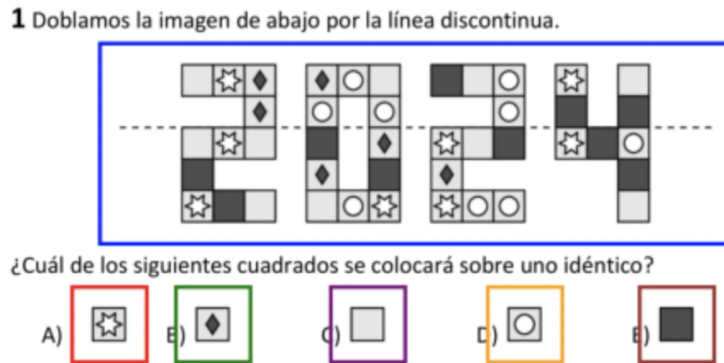


Figure 9: Example of visual localization and cropping with Florence-2 base fine-tuned.

Once the images have been isolated, the full problem goes through "*Qwen2.5-VL 7B*" again, but a final answer is still not requested. In this phase, Chain of Thought is used to force more explicit reasoning. The idea

is not for the model to give an option as soon as possible, but to write a solution plan. It must indicate which rules, comparisons or checks should be applied to decide whether each option is correct or false. This tries to reduce the behavior observed in the manual tests, where the model chose first and reasoned afterwards to make the explanation fit.

Then the system evaluates the options one by one. For each option, the statement, the plan and the corresponding option are sent. If the statement includes an image, its isolated crop is also sent. If the option is an image, the segmented crop is used. If the option is textual, its text is used. This separation is important because in a single-answer problem the model should not limit itself to defending one answer, but should accept the correct one and reject the other options.

Once the individual evaluations are available, they are sent together to "*Qwen3 4B*". This model acts as the final selector and compares the justifications before choosing the answer. Finally, a judge model receives the full image and the proposed answer to try to verify it. This design aims for two improvements. The first is a more ordered solution process. The second is a less noisy visual input. Its cost is latency, since the pipeline takes around one minute per problem, compared with the 6–8 seconds of "*Qwen2.5-VL 7B*" with a simple prompt.

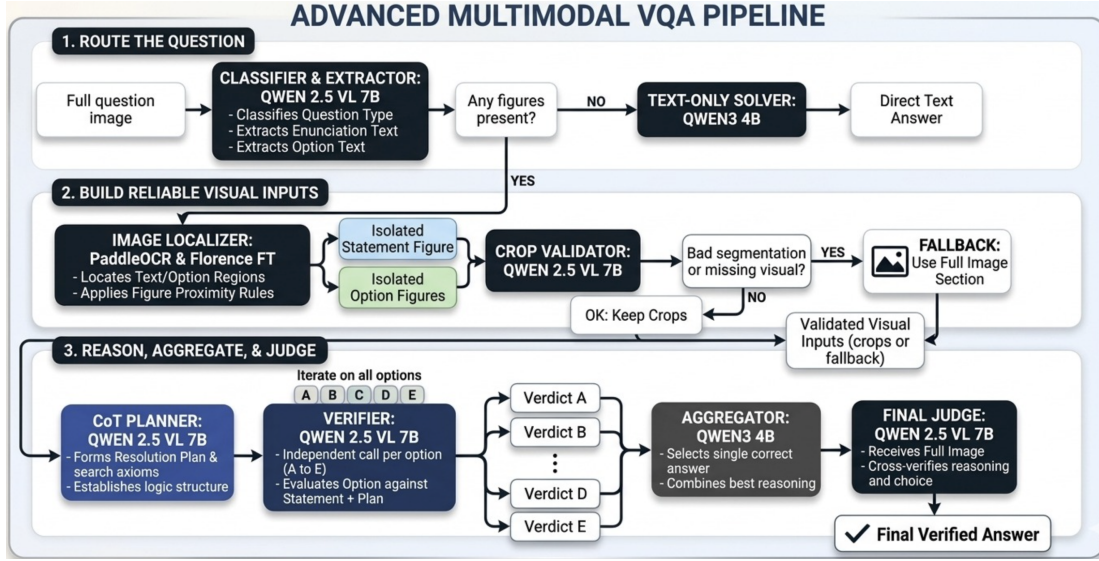


Figure 10: General view of the implemented final pipeline. Pipeline overview used in the technical integration.

Results [12] During inference, the 863 problems that needed an image were used. For each problem, the justification for each option, the answer before the judge, the final answer and the judge output were stored.

The final pipeline obtained 279 correct answers out of 863 problems, that is, 32.33% accuracy. "*Qwen2.5-VL 7B*" obtained 261 correct answers, 30.24%. The improvement is 2.09 percentage points. Therefore, the pipeline outperforms the local base model, but the improvement is limited. Around two out of three problems still fail. Also, as shown in the option-level analysis, the system rejects all alternatives in 320 cases, which is incoherent with a single-answer test and strongly suggests that visual reasoning is still not well calibrated.

Model / system	Correct answers	Accuracy
Qwen2.5-VL 72B	357/863	41.37%
Gemini 2.0 Flash	353/863	40.90%
Gemini 2.0 Flash Lite	309/863	35.81%
GPT-4o	309/863	35.81%
Final pipeline	279/863	32.33%
Pre-judge pipeline	274/863	31.75%
Qwen2.5-VL 7B	261/863	30.24%

Table 6: Main results on the visual subset. Null answers or *No Answer* are counted as failures.

The result shows an improvement over "*Qwen2.5-VL 7B*", but the pipeline remains far from the strongest models. There is not a large enough jump to state that the system solves visual reasoning robustly. The

improvement is better understood as a specific gain from structuring inference than as a general solution to the problem of visual interpretation.

The judge model had a small effect. It corrected 40 errors, but damaged 35 correct answers. The net balance was 5 cases, equivalent to 0.58% of the dataset. Also, when the previous answer was correct, the judge changed it to an incorrect one in 12.77% of the cases. This indicates that the judge does not work as a reliable verifier. It brings some corrections, but it also introduces errors.

Component	Cases	Interpretation
Incorrect $\rightarrow$ correct	40	Corrections by the judge
Correct $\rightarrow$ incorrect	35	Damage introduced by the judge
Net balance	+5	+0.58 percentage points

Table 7: Net effect of the visual judge on the final answer.

Internal coherence of the options The option-level evaluation is the most important part of the analysis. The pipeline does not only produce a final answer, it also marks each option as accepted or rejected. In a well calibrated system, the correct option should appear as accepted and the other options as rejected. The results show that this happens only a few times. Precisely by forcing the model to reason more explicitly, the problem and the nature of the weak visual reasoning of these models become clearer. Many times it is not answering because it has understood the image, but because it selects an option and then tries to build an explanation that fits the statement.

Internal metric	Value
Correct option among accepted options	41.46%
Average accepted options per problem	1.73
False acceptance rate over the other options	32.77%
Internal contradiction	81.69%
Exact set match	6.03%
All options rejected	320 cases
Correct option accepted, but final answer incorrect	130 cases

Table 8: Coherence metrics in the option-by-option evaluation.

These data show that the pipeline contains some signal, but uses it badly. In many problems it manages to include the correct option among the plausible ones. That 41.46% coverage indicates that the model sometimes detects useful information. The problem is that at the same time it accepts too many incorrect options, with a false acceptance rate of 32.77%, and produces internal contradictions in 81.69% of the cases. There are also 320 cases where the model rejects all options, which is especially contradictory in a single-answer test. Also, there are 130 problems where the correct option was accepted, but the final answer was incorrect. This means that even when the intermediate reasoning contains the good answer, the final selector does not always know how to use it.

The *exact set match* is especially clear. Only in 6.03% of the problems does the system accept exactly one option and that option is the correct one. If the pipeline were reasoning visually in a solid way, this value should be much higher. Instead, what appears is a mix of useful clues, accepted incorrect options and final decisions with little coherence. Many explanations seem made to satisfy the form of the question. The model selects an option and justifies it so that it fits what the statement asks for, even if that is not really the reason why the option would be correct.

The distribution by number of accepted options confirms this interpretation. When the system accepts two or three options, accuracy increases and there seems to be useful signal. In those cases the correct option appears more often inside the candidate set. But when it accepts four or five options, coverage stops being informative. Accepting many options increases the probability of including the correct one, but it does not prove visual understanding.

Accepted options	n	Final accuracy	Correct coverage
0	294	18.37%	0.00%
1	150	36.00%	34.67%
2	123	42.28%	58.54%
3	107	49.53%	69.16%
4	88	38.64%	84.09%
5	75	41.33%	100.00%

Table 9: Accuracy and coverage according to the number of options accepted by the pipeline.

The paired comparison with "*Qwen2.5-VL 7B*" makes it easier to see the real size of the improvement. Both systems get 144 cases right. Only "*Qwen2.5-VL 7B*" gets 117 right. Only the pipeline gets 135 right. Neither gets 467 right. The pipeline gains 18 net cases, but the most relevant point is that both fail on more than half of the subset. The structure corrects some failures of the base model, but it does not change the general difficulty of the problem.

The difference between textual and visual options is even more diagnostic. On textual options, the pipeline reaches 33.37% accuracy, but on visual options it falls to 14.58%. This result is critical because the pipeline was designed precisely to improve the treatment of figures. Segmentation and option-by-option evaluation are not enough to solve complex visual comparisons. The model can describe the figures, but it does not select the correct one in a stable way.

Comparison	n	Accuracy
Textual options	815	33.37%
Visual options	48	14.58%

Table 10: Pipeline performance according to the type of options.

By topic, level and language, the same pattern remains, as shown in more detail in Appendix C. The correct answer sometimes appears among the plausible options, but the final decision is still weak. In Logic and Visual Reasoning, for example, the correct answer appears among the accepted options in 54.92% of the cases, but final accuracy is only 30.40%. This separates two different abilities. One is finding candidates. The other is solving the problem.

There is also an answer bias. Option A is correct in 104 problems, but the pipeline predicts it 216 times. Option D is correct in 228 problems, but it is only predicted 157 times. This mismatch indicates poor calibration of the final selector and cannot be explained only by the visual content.

Final reading of the pipeline Overall, the pipeline slightly improves the "*Qwen2.5-VL 7B*" benchmark, but it does not show robust visual reasoning. The improvement is explained by a more controlled structure and by the separate evaluation of options, which makes it possible to recover some cases. However, the system still shows internal contradictions, excessive acceptance of incorrect options, weakness of the judge and low performance when the options are visual.

The main reading is that decomposing the reasoning and generating logs for each step makes it easier to see what is happening. By forcing the model to use isolated images, justify each option and leave intermediate traces, it becomes quite clear that it is not reasoning visually in a stable way. These models are effective at reading text and describing images, but they have difficulties using the image as the basis for an abstract decision. In problems that require comparing figures, imagining transformations or rejecting visual alternatives, the model tends to generate a plausible justification rather than a consistent visual verification.

For this reason, with the models tested, it seems very difficult to substantially improve the benchmarks obtained with simple prompt techniques. The limit is not only in the prompt, but in the nature of the model. Its behavior fits better with that of an advanced OCR or a visual descriptor than with that of a reasoner able to integrate the image into a mathematical decision. For a pipeline of this type to be defensible as a visual reasoning system, increasing accuracy by a few points is not enough. It must reduce contradictions, reject incorrect options better and keep coherence between accepted options and the final answer.

6 App

Kangaroo-style mathematical problems are valuable precisely because many of them cannot be solved by applying a standard classroom procedure directly. Even when the required mathematics is elementary, the solution often depends on an unconventional observation, a visual comparison, a spatial transformation, or a “think out of the box” strategy. This makes them useful for developing imagination, flexible reasoning, and transferable academic skills. However, because this type of reasoning is not usually taught explicitly in schools, students have few reliable sources from which to learn how to approach these problems step by step.

The results of this project show that current multimodal models are not yet a fully reliable source for this task. Even with a structured pipeline, the system does not solve visual Kangaroo problems in a robust way: it can misread figures, accept incorrect options, or produce explanations that are visually unsupported. For this reason, the app is not presented as a tool that improves the resolution of the problems by itself, nor as an autonomous solver. The improvement attempt belongs to the pipeline described in the previous section. The role of the app is different: it makes that pipeline usable in an educational setting, exposes its intermediate outputs, and allows students and teachers to inspect the reasoning instead of only receiving a final answer.

In practical terms, the app integrates the complete pipeline into a web interface, whose implementation is available in the **App** folder of the repository [13]. The repository also includes the necessary instructions to set up and run the app on a local computer, making the prototype accessible beyond the deployed version and allowing other users to inspect, adapt, or extend the code. The user uploads or introduces a Kangaroo problem image, and the system first applies the same classification and segmentation logic described before. If the problem requires visual information, the app separates the statement, the relevant figure, and the answer options so that the model can analyze cleaner inputs. The pipeline then generates a reasoning plan, evaluates each option individually, and produces an explicit justification for accepting or rejecting every alternative. Finally, a selector combines these option-level evaluations into a proposed final answer, and the interface presents the final option together with the reasoning trace and the individual justifications.

Figure 11 shows this interaction at interface level. The outputs are shown in Appendix D: Figure 15 presents the final answer and the option-by-option justification, and Figure 16 shows the central reasoning steps. The chat view for later clarification questions is shown in Figure 17. The prototype is available when connected to the UPV network at <http://158.42.94.34:8001>.

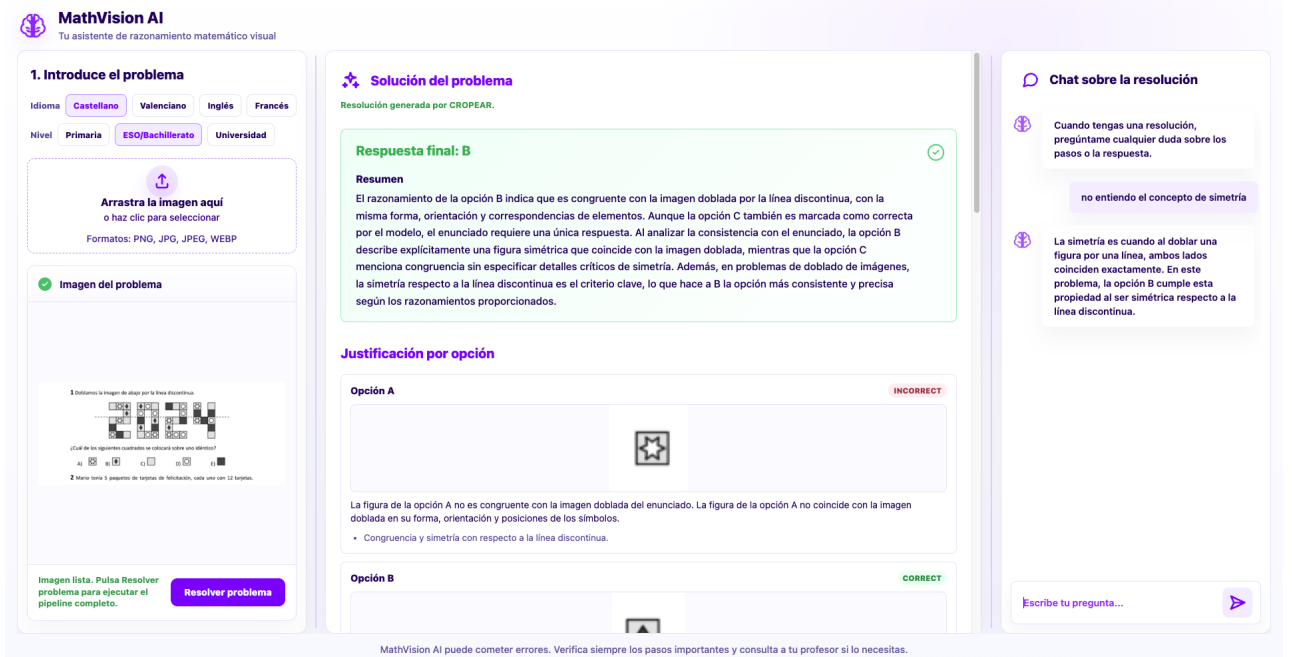


Figure 11: Interface mock-up of the educational app.

The educational value of the app is therefore not based on assuming that the model is always correct. It is based on making the answer auditable. By separating the final option, the justification for each alternative, and the reasoning trace, the interface helps students and teachers identify whether a failure comes from a missed

visual cue, an invalid mathematical assumption, or a weak comparison between possible answers. The model output becomes material for practice and discussion, rather than a result that has to be accepted blindly.

The app should therefore be understood as a minimum viable prototype derived from the research carried out in this project, rather than as a finished educational product. Its purpose is to materialize the main findings of the study in an interactive form by bringing together the pipeline, the error analysis, and the explainability approach developed during the project. In this way, the prototype makes it possible to explore how these research results could be transferred to an educational context while keeping clear that the system remains experimental.

This makes the app useful for individual study, classroom discussion, exam preparation, and formative assessment. Students can use it to receive structured feedback while working through non-routine visual problems; teachers can use the generated explanations as examples to critique or improve in class; and researchers can inspect model behaviour beyond aggregate accuracy. In this sense, the project does not only document the weakness of multimodal models in visual mathematics, but also proposes a controlled way to use them despite that limitation.

The app also defines a realistic path for social impact. It supports SDG 4, Quality Education, by making explanation-based mathematical practice more accessible and interactive; SDG 10, Reduced Inequalities, because a local or low-cost version can support students without access to private tutoring; and SDG 9, Industry, Innovation and Infrastructure, because it converts a benchmark workflow into an operational prototype. The main limitation remains that the model can still produce incorrect answers or visually unsupported explanations. For that reason, the app should not be used for high-stakes evaluation or unsupervised grading. Its role is formative: it supports practice, verification, and discussion.

7 Future and Legacy

The project leaves behind a reusable foundation for future research and development in visual mathematics: a curated dataset, an error analysis, an inspectable reasoning pipeline, and an educational interface. From this foundation, the most relevant next steps are improving reliability through executable reasoning and through a memory of verified solved problems.

One promising direction is the use of *Program-Aided Language Models* (PAL), where the model translates part of the reasoning into executable Python code and delegates the final computation to an interpreter [14]. This line of work is already being tested on problems that the previous pipeline failed to solve, and in some of these cases the program-aided formulation recovers the correct answer by making the numerical or symbolic operations explicit. Its main limitation is evaluation: the reasoning is no longer only a natural-language explanation, but also Python code, so validating the result requires checking whether the generated program faithfully formalizes the mathematical and visual assumptions of the problem.

The second direction consists of a retrieval-augmented memory containing correctly solved examples, following the RAG paradigm that combines text generation with an external dense memory accessed during inference [15]. In this project, the vector database would store only successful solutions as triples composed of the problem statement, the reasoning trace, and the final answer. The first entries would come from questions correctly solved by Gemini, because its reasoning traces are robust enough to be reused as solved examples, the same memory can then be expanded with Qwen solutions whenever Qwen also answers correctly. When a new problem appears, the system retrieves semantically similar solved cases and uses them as verified examples to guide the reasoning, while keeping retrieval restricted to validated answers to reduce false analogies.

Next Steps Our project’s journey began as an experiment, it has grown into a meaningful commitment to educational innovation. The most valuable lesson we learned was how to transform raw data and research findings into a tangible tool, a process that resulted in our app. This transition taught us that technology is most powerful when it serves a clear social purpose. By aligning our work with Quality Education (SDG 4) and Reduced Inequalities (SDG 10), we aim to support the academic community and the “Conselleria d’Educació” in making math learning more accessible for everyone.

We also plan to extend the benchmark and submit it to the NeurIPS 2027 workshop on foundation models, reasoning, or benchmark evaluation [17], while releasing the dataset, error taxonomy, and evaluation pipeline to ensure reproducibility and facilitate model comparison.

We have hosted all detailed documentation, our code, and the data gathered on GitHub, leaving the doors open for anyone to be able to build upon this project. In particular, the README of the repository **MathVisionProy3** includes a tutorial explaining how to download the project, configure the required environment, and run the app locally. This ensures that our work remains useful and replicable. Moreover, we are committed to contributing to the scientific community by sharing our methodology and findings through formal channels. We want to do an arXiv publication to ensure our research is accessible to other researchers, while simultaneously preparing a more extensive submission for the journal "Computer Vision and Image Understanding." By presenting our work to the Conselleria d'Educació, we aim to turn our academic results into a real-world proposal for schools. This ensures that our project serves as both a practical resource and a theoretical reference for the future of educational technology.

8 Conclusions

Although it was initially hypothesized that structured multimodal pipelines and explicit option level justifications could outperform simple prompting methods, empirical results demonstrate that by increasing the complexity, it is revealed the fundamental lack of internal coherence, rather than a leap in reasoning. Decomposing the model's decision making process highlights an unstable behavior where justifications are retroactively constructed to satisfy the prompt rather than being derived from visual evidence. With an internal contradiction rate of 81.69%, it becomes evident that these models operate as sophisticated "pleasers" that guess an option (that could be random) and then elaborate a plausible narrative to fit the question without being genuine certainty about the given answer.

The analysis of specific failure modes confirms that current multimodal architectures encounter a persistent perceptual bottleneck, excelling as advanced OCR systems but failing as spatial reasoners. Systemic errors in basic counting, spatial folding, and rotation tasks persist regardless of inference complexity, indicating that models lack the intrinsic spatial intuition required for three-dimensional mental manipulation. These findings suggest that the limitation is structural, models effectively describe visual inputs but struggle to integrate that information into a consistent mathematical framework. Future advancements should therefore move beyond purely image description and focus on developing stronger numerical grounding, spatial consistency, and executable reasoning to bridge the gap between visual perception and abstract logic.

9 References

References

- [1] Igualde Sáez, A., Rhomrasi, L., Ahsini, Y., Vinuesa, R., Hoyas, S., García Sabater, J. P., Fullana i Alfonso, M. J., and Conejero, J. A. (2025). *Evaluating Visual Mathematics in Multimodal LLMs: A Multilingual Benchmark Based on the Kangaroo Tests*. arXiv:2506.07418. <https://arxiv.org/abs/2506.07418>
- [2] Rhomrasi et al. (2025). LLM performance on mathematical reasoning in Catalan language. <https://www.sciencedirect.com/science/article/pii/S2590123025004475>
- [3] Rhomrasi et al. (2025). Evaluation of LLM performance on mathematical reasoning (Kangaroo tests). GitHub. <https://github.com/mc-flai/kangur-ai-evaluation>
- [4] *Own project data*. Cleaned and unified the main Kangaroo repository data. <https://github.com/gandpablo/MathVisionProy3/blob/main/DATA/DatasetFull.csv>
- [5] *Own project data*. Replicated experiment with Qwen2.5-VL 7B. <https://github.com/gandpablo/MathVisionProy3/blob/main/DATA/ReplicadoQwen2-5VL7b.csv>
- [6] *Own project code*. First test pipeline. <https://github.com/gandpablo/MathVisionProy3/tree/main/Pipeline/first>
- [7] Xu, J., Fei, H., Pan, L., Liu, Q., Lee, M.-L., and Hsu, W. (2024). *Faithful Logical Reasoning via Symbolic Chain-of-Thought*. In *Proceedings of the 62nd Annual Meeting of the Association for Computational Linguistics*, pages 13326–13365. <https://aclanthology.org/2024.acl-long.720/>
- [8] Wei et al. Chain-of-Thought Prompting Elicits Reasoning in Large Language Models. <https://arxiv.org/abs/2201.11903>
- [9] *Own project code*. Final pipeline code. <https://github.com/gandpablo/MathVisionProy3/tree/main/Pipeline/second/code>
- [10] PaddleOCR. <https://github.com/PaddlePaddle/PaddleOCR>
- [11] Florence-2, Microsoft vision-language model. <https://arxiv.org/abs/2311.06242>
- [12] *Own project data*. Dataset answered by the final pipeline. <https://github.com/gandpablo/MathVisionProy3/blob/main/DATA/DatasetRespondidoPipeline.csv>
- [13] *Own project code*. Educational application prototype. <https://github.com/gandpablo/MathVisionProy3/tree/main/App>
- [14] Luyu Gao et al. PAL: Program-aided Language Models. ICML 2023. <https://arxiv.org/abs/2211.10435>
- [15] Patrick Lewis et al. Retrieval-Augmented Generation for Knowledge-Intensive NLP Tasks. NeurIPS 2020. <https://arxiv.org/abs/2005.11401>
- [16] Bansal, P., and Sharma, A. (2023). *Large Language Models as Annotators: Enhancing Generalization of NLP Models at Minimal Cost*. arXiv:2306.15766. <https://arxiv.org/abs/2306.15766>
- [17] NeurIPS. Call for Workshops. <https://neurips.cc/Conferences/2025/CallForWorkshops>

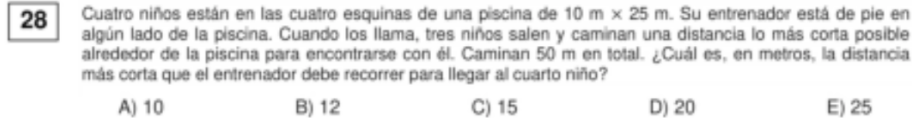
A Annexes

A.1 Referenced Repository Structure

Repository part	Path or link
Reference paper repository	mc-flai/kangur-ai-evaluation
Full working dataset	DATA/DatasetFull.csv ^{own repo}
Exploratory analysis notebook	General code ^{own repo}
Qwen replication data	DATA/ReplicadoQwen2-5VL7b.csv ^{own repo}
First pipeline	Pipeline/first ^{own repo}
Final pipeline code	Pipeline/second/code ^{own repo}
Pipeline prompts	Pipeline/second/prompts ^{own repo}
Pipeline output dataset	DATA/DatasetRespondidoPipeline.csv ^{own repo}
Local application	App ^{own repo}

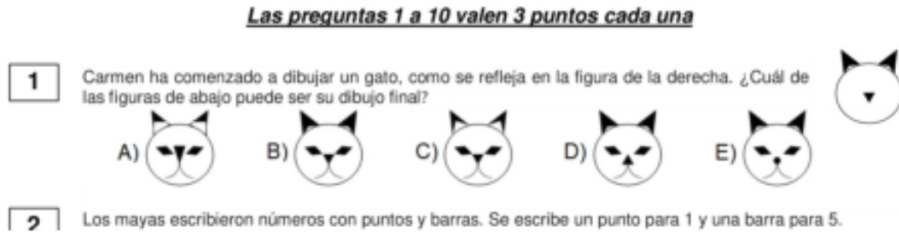
Table 11: Repository structure referenced throughout the report.

A.2 Examples of Problems With and Without Visual Dependence



Problem without visual dependence

All the information required by the model is explicitly provided in the statement. Therefore, the problem can be solved without needing to understand the image. In this case, most of the stronger models evaluated in the experiments were able to solve it correctly.



Problem with visual dependence

In this example, solving the problem correctly requires understanding the visual characteristics of the image shown in the statement and comparing them with the possible answer options. Although the problem is very simple and belongs to the most basic level of the test, it requires visual acuity and accurate shape recognition across all the provided images.

As a result, all the evaluated models give an incorrect answer. Some models either do not provide a clear answer or produce long, unfocused reasoning based on mathematical arguments that are unrelated to the actual solution, which simply consists of identifying visual features. The closest model is Gemini 2.0 Flash: it describes the image in the statement reasonably well, but still fails to select the correct option, even though its reasoning is comparatively coherent.

Figure 12: Representative examples of problems with and without visual dependence.

Note. Top: an example that can be solved mainly from the text. Bottom: an example whose solution does require interpreting the figure.

A.3 Chain of Thought (CoT) Analysis in Multimodal Models

The following table summarizes the key characteristics of Chain of Thought reasoning across the evaluated multimodal models. The main distinction among models is not the presence of CoT, but its depth, control, and consistency.

Model	CoT Type	Native	Depth	Control	Reliability	Planning
GPT-4o	Implicit (latent)	Yes	Low	Low	Medium	No
Gemini 2.0 Flash	Enhanced implicit	Yes	Medium	Low	Med-high	Limited
Gemini 2.0 Flash Lite	Enhanced implicit	Yes	Med-low	Low	Medium	Limited
Llama 3.2 11B	Prompt-based	No	Low	High	Low	No
Llama 3.2 90B	Prompt-based*	No	Medium	High	Med-low	Limited
Pixtral Large	Weak emergent	Partial	Medium	Low	Medium	Limited
Pixtral-12B	Weak emergent	Partial	Low	Low	Low-med	No
Qwen-VL 72B	Advanced implicit	Yes	High	Low	High	Limited
Qwen-VL 7B	Implicit multimodal	Yes	Medium	Low	Med-high	Limited
Qwen2.5VL (3B)	No CoT	N/A	N/A	N/A	High	No
Qwen3 (4B)	Implicit + induced	Yes	Medium	High	Medium	Limited

Table 12: Comparison of Chain of Thought characteristics across multimodal models.

Note. *Scale-enhanced. The key difference is not the presence of CoT, but its depth, control, and consistency.

B Reasoning Annex

B.1 Reasoning context

We will study the limitation of the reasoning of the models by analysing the performance of Gemini 2.0 Flash and Qwen-VL 7B on the mathematical Kangaroo challenges, especially on those questions where an image is necessary to solve the problem.

To establish the baseline for this study, we did a manual analysis of 60 representative problems. We dissected the underlying reasoning patterns and the specific linguistic and visual characteristics of these questions, identifying the different cognitive demands needed to solve the problem. This step is crucial in understanding the specific mental leaps required to arrive at a correct solution, as these leaps often prove to be stumbling blocks for AI architectures.

Building upon this groundwork, we performed an inference process across the entire dataset. To do this, we used Qwen-VL 7B to categorise the problems. We designed the following labels:

B.1.1 Problem Description

1. **geometry_area**: Problems focused on calculating, comparing, or transforming geometric areas. Includes shaded regions, equivalent areas, and the subdivision of surfaces.
2. **geometry_angle**: Problems where the primary objective is to calculate or compare angles. Includes angular relationships in polygons, circles, and triangles.
3. **geometry_length**: Problems related to distances, perimeters, sides, or geometric proportions. Includes unknown lengths and metric relationships.
4. **grid_reasoning**: Problems based on grids, tables, or cells. The discrete spatial arrangement is a key element.
5. **spatial_rotation**: Problems requiring the mental rotation of objects or pieces. Includes tiling, interlocking figures, and the rotation of patterns.
6. **spatial_folding**: Problems that require imagining how a 2D figure folds or transforms into 3D. Includes cubes, dice, and net layouts.
7. **spatial_path**: Problems related to routes, paths, or connectivity. Includes valid trajectories and movements across graphs or grids.
8. **pattern_recognition**: Problems where a visual or numerical rule or pattern must be identified. Includes

sequences and repetitive regularities.

9. **constraint_satisfaction**: Problems where several constraints must be met simultaneously. Includes sudokus, logical restrictions, and multiple conditions.
10. **combinatorial_counting**: Problems involving the counting of valid configurations or possibilities. Includes combinations, arrangements, and case-by-case counting.

B.1.2 Reasoning Errors

1. **Spatial Folding Deficit**: The inability to visualize 3D objects from 2D nets (such as folding layouts into cubes, dice, or prisms).
2. **Rule Hallucination**: Inventing non-existent constraints (e.g., prohibiting numbers that are actually permitted) or ignoring provided rules.
3. **Visual Processing Failure (OCR)**: Confusion regarding numbers, colors, or the incorrect localization of elements (such as the position of a question mark) within an image.
4. **False Geometric Relationships**: Assuming non-existent equalities (e.g., claiming a side has the same length as a radius without evidence).
5. **Basic Counting Errors**: Incorrectly counting line segments, the number of grid cells, arrows, or repeated elements.
6. **Unnecessary Overcomplication**: Performing complex calculations (such as trigonometry or using the value of π) in problems that only require simple visual logic.
7. **Poor Figure Decomposition**: Inability to break down shaded or complex areas into simple geometric shapes.
8. **Positional and Hierarchical Confusion**: Failure to understand the circularity of an object or the order of overlapping layers (determining which element is on top of another).
9. **Coincidental Correctness**: Providing the correct answer through a completely fabricated or flawed reasoning process.
10. **Process Inconsistency**: Changing values or premises midway through the exercise, losing the thread of one’s own logic.
11. **No_error**: Indicates that no reasoning error was detected.

Then, we created two prompts: the first prompt was used to categorise the descriptions of the images, and the second one was used to categorise the reasoning problems. It is important to note that this was a multilabel problem. These prompts were then translated into Spanish, French, and Catalan.

The specific prompts can be seen in the GitHub repository.

Ultimately, by analysing the categorization of the failure modes of these models in the context of the Kangaroo problems, we provide insights into the current state of multi-modal reasoning.

B.2 General Analysis

In this section, we present a general performance analysis of the evaluated models, focusing on their overall success rates and their ability to handle different types of mathematical challenges. We will observe the contrast between the performances of Gemini 2.0 Flash and Qwen-VL 7B.

B.2.1 Accuracy of the Models

Firstly, we turn our attention to the overall effectiveness of the models, as illustrated in Figure 1 (Appendix). We can see that Gemini 2.0 Flash demonstrates a clear superiority in overall performance. While Qwen-VL 7B achieves an accuracy of approximately 0.3, Gemini 2.0 Flash reaches a higher accuracy of roughly 0.4.

Despite this lead, the results indicate that both models struggle significantly with the complexity of the mathematical problems, as neither manages to surpass the 50% accuracy mark globally. This highlights substantial room for improvement in how multi-modal models synthesize visual information with mathematical logic.

B.2.2 Accuracy per Topic

Secondly, we analyse the accuracy of the evaluated models across the specific mathematical topics identified in the study. There are five core areas: *Combinatoria y Probabilidad*, *Geometría y Figuras*, *Lógica y Razonamiento Visual*, *Patrones y Secuencias*, and *Álgebra y Aritmética Visual*. The performance of the models can be observed in Figure 2 (Appendix).

When breaking down the performance by category, as seen previously, Gemini 2.0 Flash consistently outperforms Qwen-VL 7B across all specific fields. The models show their highest proficiency in the categories of *Geometría y Figuras* and *Álgebra y Aritmética Visual*, where Gemini approaches an accuracy of 0.45.

The most challenging area for both models is *Lógica y Razonamiento Visual*, which exhibits the lowest accuracy scores. This suggests that while the models are relatively better at identifying geometric shapes or performing visual arithmetic, they face severe difficulties in problems that require abstract logical deductions derived from an image.

The initial data analysis establishes Gemini 2.0 Flash as the benchmark model for this study due to its higher reliability across all tested categories. However, the low accuracy scores in areas such as *Lógica y Razonamiento Visual* suggest that the primary difficulty for these LLMs lies in the high cognitive processing required to solve abstract puzzles. This general overview provides the necessary context to move forward with a deeper investigation into the specific reasoning failures that contribute to these performance gaps.

B.3 Description Analysis

Once an initial general analysis has been performed to determine the basic characteristics of the data, it is time to analyze the added variables (*description*, *reasoning_error*). The first variable to be analyzed is *description*. This variable indicates not the subject matter of the problem, but rather the knowledge or actions required to solve it. This allows us to understand which types of reasoning processes or cognitive skills are most challenging for language models.

B.3.1 Description Distribution

The first aspect of interest is the distribution of this variable, with the aim of determining the frequency with which the different labels appear in the dataset.

In Figure 3 (Appendix), the frequency of these categories can be observed. The categories are sorted in descending order, allowing the identification of the most predominant ones. The first category, clearly standing out from the rest, is *pattern_recognition*, with a frequency close to 375 cases. It is followed by *constraint_satisfaction* and *geometry_area*, with approximate values of 250 and 200 respectively, suggesting that the dataset is heavily skewed toward these fundamental concepts.

As we move toward the right side of the graph, a gradual decrease in the frequency of the labels can be observed. Topics such as *spatial_rotation* and *geometry_length* show a moderate presence, while more specific categories such as *spatial_folding*, *spatial_path*, *combinatorial_counting*, *geometry_angle*, and *grid_reasoning* present a much lower and more uniform distribution, all below 100 cases.

At the far end, the category *geometry_volume* appears with only one case. This is an important observation, since such a low representation means that the category will not be considered statistically significant, even though it is included in the subsequent analysis.

B.3.2 Description Distribution per Topic

Next, the distribution of descriptions by subject matter is analyzed in order to visualize which capabilities are most closely associated with each thematic area.

As can be observed in the heatmap shown in Figure 4 (Appendix), the capabilities are distributed in diverse ways. The first aspect that immediately stands out is that the topic *Geometry and Shapes* is the densest category, especially linked to the labels *geometry_area* (with the maximum value of 132) and *pattern_recognition* (89). This suggests that the dataset is heavily oriented toward visual surface problems and shape recognition.

On the other hand, an uneven distribution in the use of descriptive labels can be observed. While *pattern_recognition* and *constraint_satisfaction* are transversal and appear frequently across almost all topics,

labels such as *geometry_volume* are practically absent from all categories due to their extremely low frequency.

This imbalance indicates that the dataset focuses predominantly on logical reasoning, visual patterns, and area-related problems, while concepts involving volume, trajectories, or advanced grid reasoning are comparatively underrepresented. In summary, the graph reveals a predominant emphasis on visual reasoning and two-dimensional geometry, leaving more specific concepts related to three-dimensional geometry or grid analysis in the background.

B.3.3 Co-occurrence Analysis

Once the distribution of this specific problem classification has been established, it becomes interesting to determine the frequency with which its categories overlap. To illustrate this phenomenon, a co-occurrence matrix was created, showing the percentage of overlap between the different categories.

As shown in Figure 5 (Appendix), several highly specific relationships emerge. For example, there is a total relationship (100%) between *geometry_volume* and both *geometry_angle* and *geometry_length*. However, as previously mentioned, *geometry_volume* only appears once, meaning that these percentages are not statistically meaningful.

A more representative and significant relationship can be observed between *pattern_recognition* and *constraint_satisfaction*, with a co-occurrence of 52.3%. This suggests that pattern recognition tasks are frequently accompanied by logical constraints.

Other relevant relationships, although slightly lower, also exceed 20%. First, the relationship between *pattern_recognition* and *spatial_path* reaches 34.3%, indicating that many problems involving paths or trajectories also require the identification of structured visual patterns.

Similarly, the relationship between *spatial_folding* and *spatial_rotation* reaches 30.4%, showing that tasks involving the mental folding of figures are intrinsically linked to the ability to mentally rotate objects in space.

Furthermore, the relationship between *pattern_recognition* and *spatial_rotation* stands at 29.6%, indicating that the analysis of repetitive shapes or sequences frequently requires mental orientation changes. Additionally, *grid_reasoning* and *constraint_satisfaction* show a co-occurrence of 25.5%, meaning that puzzles based on grids often rely on fulfilling logical rules or restrictions.

Finally, the relationship between *spatial_path* and *combinatorial_counting* reaches 20%, suggesting that some trajectory problems require calculating or counting the number of valid possible paths.

The co-occurrence analysis reveals that the dataset is not composed of isolated categories, but rather interconnected cognitive skills. The strong relationship between pattern recognition and constraint satisfaction (52.3%) positions this pairing as the central core of the evaluated reasoning tasks. Likewise, a solid block of spatial reasoning emerges, where rotation, folding, and trajectories frequently overlap, demonstrating that solving complex geometric problems requires the simultaneous integration of multiple cognitive processes.

B.3.4 Capabilities Accuracy

Once the co-occurrence patterns of these categories have been studied, it is necessary to examine the accuracy achieved by both models within each capability. This analysis allows the identification of which cognitive categories generate the greatest difficulties for the models.

Gemini Figure 6 (Appendix) presents the accuracy associated with each of the previously described categories for the Gemini model, allowing several important conclusions to be extracted.

First, a category with 100% accuracy stands out: *geometry_volume*. However, since it only contains a single sample and that sample was answered correctly, this value is not representative of the model’s actual performance.

More representative is the model’s poor performance in the area of spatial reasoning. The categories *spatial_rotation* (23.5%) and *grid_reasoning* (27.7%) exhibit the lowest success rates despite containing a significant number of samples. This suggests limitations in the model’s ability to mentally manipulate objects in space or solve coordinate- and grid-based problems, both of which require a deeper understanding of three-dimensionality and relative position.

Conversely, the model demonstrates significantly higher competence in categories related to classical geometry and measurement calculations. Labels such as *geometry_length* (53.3%), *geometry_angle* (49.2%), and *geometry_area* (46.6%) show substantially higher precision.

This pattern indicates that the system performs more effectively in problems that can be solved through deterministic formulas or explicit geometric rules, rather than tasks depending on pure spatial intuition or abstract pattern manipulation.

In summary, the results reveal a substantial gap between the model’s ability to execute logical-mathematical operations and its aptitude for abstract spatial reasoning. While the system shows relative proficiency in analytical geometry tasks, its performance drops drastically when faced with problems requiring mental manipulation of objects or the interpretation of complex visual structures. This disparity suggests that, although the model has learned the fundamentals of geometric calculation, it still lacks true spatial intuition.

Qwen Figure 7 (Appendix) shows the accuracy associated with each evaluated category for the Qwen model, allowing several conclusions to be drawn.

Again, the category *geometry_volume* reaches 100% accuracy. However, since this category contains only a single sample, this result is not statistically representative.

What is truly representative is the model’s weak performance in spatial and logical reasoning. The categories *spatial_folding* (24.2%) and *spatial_rotation* (24.3%) record the lowest success rates, closely followed by *pattern_recognition* (25.6%) and *spatial_path* (25.7%).

This suggests structural limitations in the model’s ability to mentally manipulate objects or solve problems involving trajectories and rotations, skills that require a deep understanding of relative spatial positioning.

Conversely, the model demonstrates superior competence in categories related to technical geometry. Labels such as *geometry_angle* (54.2%), *geometry_length* (36.2%), and *geometry_area* (33.7%) show considerably higher precision compared to the spatial reasoning block.

This indicates that the system performs more efficiently when solving problems through explicit formulas or regulated geometric procedures than when relying on abstract spatial intuition.

In summary, the results reveal a clear gap between the model’s ability to execute regulated geometric operations and its aptitude for abstract spatial processing. While the system demonstrates relative proficiency in analytical geometry tasks such as angles and lengths, its performance drops sharply in categories such as *grid_reasoning* (27.7%) and *constraint_satisfaction* (26.8%).

This disparity suggests that, although the model handles the fundamentals of calculation reasonably well, it still lacks genuine visual intuition, limiting its effectiveness in environments requiring complex spatial reasoning and three-dimensional understanding.

B.3.5 Models Comparison

Figure 8 (Appendix) presents a comparative analysis between Gemini and Qwen, evaluating their accuracy across different logical, spatial, and geometric reasoning capabilities.

It can be observed that Gemini generally outperforms Qwen across almost all categories, with only a few exceptions. The categories *spatial_rotation* and *grid_reasoning* exhibit nearly identical precision values in both models, suggesting that both architectures share similar limitations or have reached a common performance ceiling in tasks involving orientation and grid structuring.

On the other hand, Qwen slightly surpasses Gemini in *geometry_angle*, indicating that the Qwen model may possess stronger training or specific optimization for angular calculations and geometric relationships.

Overall, the graph reveals a clear competitive advantage for Gemini in most evaluated capabilities. Its dominance is particularly notable in complex geometric reasoning tasks, including area and length calculations, as well as combinatorial counting.

However, Gemini does not hold an absolute advantage. Qwen demonstrates superior efficacy in angle interpretation, while both models exhibit similarly weak performance in spatial orientation tasks.

Following the analysis of the descriptions, several important conclusions can be drawn. A fundamental finding is the strong interconnection between reasoning capabilities identified through the co-occurrence analysis.

The close relationship between pattern recognition and constraint satisfaction (52.3%) demonstrates that these skills do not operate independently but instead form the central core of the evaluated reasoning tasks.

Likewise, a solid spatial reasoning block emerges where rotation, folding, and trajectories overlap significantly, confirming that solving complex geometric problems requires the simultaneous integration of multiple cognitive processes.

Regarding model performance, both Gemini and Qwen exhibit a substantial gap between regulated logical operations and genuine spatial intuition. Both models achieve their best results in classical geometry tasks involving angles and lengths, where explicit formulas and deterministic rules facilitate problem solving.

In contrast, performance declines drastically in spatial rotation and grid reasoning tasks, suggesting that current large language model architectures still lack a deep understanding of three-dimensionality and mental object manipulation.

Finally, the comparison between the models positions Gemini as the stronger overall system in most categories, particularly in area and length calculations. Nevertheless, it does not maintain a universal advantage, as Qwen performs better in angle interpretation and both models share similarly low performance ceilings in spatial orientation tasks.

In conclusion, this study highlights that, although artificial intelligence has made considerable progress in logical-mathematical processing, genuine visual and spatial intuition remains one of the most critical unresolved challenges in the development of multimodal reasoning systems.

B.4 Reasoning Error Analysis

Following the analysis of the visual descriptions, we now focus on the analysis of reasoning failures. While the previous section examined how models perceive the input, this section investigates the internal logic applied by the LLMs to reach a solution. By systematically categorizing the specific errors made during the inference process, we aim to uncover recurring patterns when the LLMs are faced with mathematical logic requiring visual interpretation.

To achieve a deeper understanding of these failures, we utilize the specialized error labels defined during the manual review of the initial 60 problems. By mapping these errors across the entire dataset, we can determine whether certain models are predisposed to specific types of reasoning failures.

By identifying these patterns, we can understand whether the failures originate from a lack of mathematical knowledge or from a more fundamental inability to maintain logical coherence, such as in cases of *Coincidental Correctness* or *Positional and Hierarchical Confusion*.

B.4.1 Reasoning Error Distribution

To better understand the cognitive barriers encountered by the models, we perform an extensive diagnostic analysis of their failures. Before directly comparing both systems, we analyze the reasoning error distribution for each model individually. This approach allows us to construct a “failure profile” for each architecture, identifying whether their mistakes are systemic or dependent on specific mathematical challenges.

Gemini Firstly, we analyze Gemini’s reasoning error distribution. Figure 9 (Appendix) ranks the frequency of each previously defined reasoning error category.

At first glance, the distribution is heavily dominated by the *NO_ERROR* category, which reaches a frequency of approximately 250 instances. This label indicates that the model was unable to identify any error in the reasoning process. However, this category introduces a considerable amount of noise, since some incorrect solutions may have been misclassified as error-free.

Beyond this dominant category, several logical failure types emerge prominently, including *Basic Counting Errors*, *False Geometric Relationships*, and *Spatial Rotation Errors*. This distribution suggests that while the model exhibits systemic weaknesses, these weaknesses are diverse and task-specific rather than concentrated in a single reasoning domain.

To conduct a more meaningful diagnostic of the model’s weaknesses, it is necessary to isolate the true error cases. This is achieved by filtering out the *NO_ERROR* instances, allowing the relative impact of each error

category to become visible. Consequently, we can better identify which cognitive domains are most challenging for Gemini 2.0 Flash.

To evaluate the impact of the *NO_ERROR* category, a confusion matrix was generated (Figure 10, Appendix). In this matrix:

- True Negatives correspond to cases where Gemini answered correctly and the judge identified the reasoning as correct.
- True Positives correspond to cases where Gemini answered incorrectly and the judge successfully detected the error.

The error detection system used in this process was Qwen-VL 7B, which analyzed the reasoning chains and assigned qualitative labels whenever errors were identified. This setup also allowed us to evaluate Qwen’s capabilities as an external judge model.

The results reveal that Qwen-VL 7B behaves as a highly conservative and critical evaluator when judging Gemini’s outputs. The confusion matrix shows 385 True Positives, indicating that the judge successfully detected most of Gemini’s reasoning failures. This suggests that Qwen is effective at recognizing logical inconsistencies in external reasoning chains.

However, this effectiveness is contrasted by 30 False Negatives, where Gemini produced an incorrect answer that Qwen failed to identify as erroneous.

Another significant issue is the high number of False Positives, totaling 120 cases. In these situations, Gemini reached the correct solution, yet Qwen incorrectly flagged the reasoning as flawed. This suggests that the judge may lack the flexibility to recognize unconventional but valid reasoning paths.

When compared to the 29 cases where both models agreed on a correct and error-free solution, it becomes evident that the judge is considerably more prone to over-diagnosing errors than validating successful reasoning.

Following the confusion matrix analysis and the initial label distribution, the dataset was further cleaned. Specifically, all instances where the models answered correctly or where the judge failed to provide a descriptive error label were removed.

The resulting dataset therefore represents only confirmed failures where:

1. The ground truth answer was incorrect.
2. The judge successfully assigned a qualitative reasoning error label.

This refinement shifts the analysis from a quantitative perspective of “how often” the models fail to a qualitative exploration of “why” they fail.

The filtered bar chart shown in Figure 11 (Appendix) reveals the true distribution of Gemini’s reasoning failures once the dominant *NO_ERROR* category has been removed.

The most significant finding is the prevalence of *Basic Counting Errors*, which emerge as the most frequent failure mode. This indicates that despite the model’s advanced reasoning capabilities, fundamental arithmetic and enumeration remain its greatest cognitive blind spots in visually grounded tasks.

Following counting errors, the dataset reveals a high frequency of *False Geometric Relationships* and *Spatial Rotation Errors*, suggesting systemic difficulties in handling visual-spatial reasoning.

These results indicate that the model’s reasoning weaknesses are not random, but rather concentrated in tasks requiring precise numerical grounding and spatial consistency.

By focusing exclusively on confirmed failures successfully identified by the judge, this refined distribution provides a reliable map of Gemini’s functional limitations. The analysis strongly suggests that improvements in basic numerical grounding and spatial reasoning would likely produce the most substantial gains in overall reliability.

Qwen We now analyze the reasoning error distribution of Qwen-VL 7B. Figure 12 (Appendix) presents the frequency of the different reasoning error labels, following the same methodology applied to Gemini.

As with Gemini, the initial bar chart is dominated by the *NO_ERROR* category. However, in Qwen’s case,

this dominance is even more pronounced, with nearly 350 instances labeled as error-free.

This massive “correctness spike” suggests that when Qwen acts as its own judge, it exhibits a strong tendency to validate its own reasoning, even when the final answer may be incorrect. This behavior is further analyzed through the confusion matrix presented in Figure 13 (Appendix).

Unlike the previous experiment, Qwen now evaluates its own reasoning chains. The confusion matrix reveals a substantial drop in diagnostic reliability compared to when Qwen judged Gemini.

Most notably, Qwen produces approximately 90 False Negatives, meaning that the model frequently failed to solve the problem correctly while simultaneously classifying its own reasoning as flawless.

When judging Gemini, Qwen only missed approximately 30 incorrect solutions. Therefore, the self-evaluation setting produces nearly three times more undetected failures.

This strongly suggests that Qwen performs significantly better as an external evaluator than as a self-judge. One possible explanation is that Gemini produces more transparent and interpretable reasoning chains, making its logical inconsistencies easier to detect.

These findings suggest that future AI-judge pipelines would likely benefit from using external models as evaluators whenever technically feasible.

Following the same methodology used for Gemini, the Qwen dataset was also cleaned by removing all *NO_ERROR* cases and unlabeled failures.

The resulting refined visualization, shown in Figure 14 (Appendix), reveals that *Basic Counting Errors* remain the dominant failure mode, closely followed by *False Geometric Relationships*.

This pattern indicates that Qwen’s primary weaknesses, much like Gemini’s, are rooted in fundamental numerical grounding and spatial interpretation.

A particularly interesting observation is the strong presence of *Spatial Folding Deficit* and *Process Inconsistency* as secondary but still significant error categories.

This suggests that while the model often understands the general context of the problem, it struggles during the final execution stages where precise spatial relationships or multi-step logical consistency are required.

The importance of this cleaned distribution lies in demonstrating that the model’s failures are not random but instead follow a predictable structural pattern. Consequently, future training strategies should prioritize datasets involving counting, spatial reasoning, and geometric consistency.

B.4.2 Comparison Between Models

The next step is to perform a comparative analysis between both architectures in order to identify systemic patterns of failure shared across multimodal reasoning systems.

By placing the reasoning error distributions side by side, we can determine whether certain logical failures are model-specific or whether they represent broader limitations of current AI reasoning architectures.

Figure 15 (Appendix) reveals a striking alignment between both models, particularly in *Basic Counting Errors*. In both systems, numerical inaccuracy emerges as the dominant failure mode, suggesting that accurately quantifying visual objects remains a fundamental weakness of current vision-language models. Although both models struggle in similar domains, Gemini exhibits slightly higher frequencies in some reasoning categories, while Qwen’s failures appear more concentrated in geometric misinterpretations.

This indicates that although both systems fail for similar reasons, the underlying logical structure of their reasoning differs slightly. Additionally, the prevalence of *False Geometric Relationships* across both models confirms that spatial reasoning represents the second major source of failure.

The fact that two independently developed architectures share nearly identical high-frequency error types strongly suggests that these are not isolated implementation issues but structural limitations of current multimodal reasoning systems.

These findings imply that any future “AI Judge” or supervisory architecture must be specifically reinforced in counting and geometry-related tasks, as these domains remain universally difficult even for advanced models such as Gemini and Qwen.

B.4.3 Reasoning Error Overlap

To further investigate the similarities between both systems, we compare the overlap between all reasoning error labels.

This analysis allows us to examine how each model perceives and reproduces similar logical failures, providing insights into their shared biases and cognitive limitations. As shown in Figure 16 (Appendix), one of the most significant findings is the strong symmetry observed in *Spatial Folding Deficit*, where both models exhibit a 66.7% overlap.

This strongly suggests that 3D visualization remains a fundamental challenge for the current generation of multimodal models, regardless of their underlying architecture. Additionally, when Gemini exhibits *Basic Counting Errors*, there is a 50.5% probability that Qwen commits the same failure. This suggests that both models rely on similar internal heuristics when interpreting geometric relationships and visual quantities.

An especially interesting pattern emerges when Gemini exhibits *Rule Hallucination* or *Process Inconsistency*. In these cases, Qwen frequently responds with *Basic Counting Errors* (100% and 46.4% overlap, respectively).

This suggests a hierarchy of cognitive failure:

- Gemini tends to lose logical coherence at a higher abstract reasoning level.
- Qwen collapses at a more perceptual level related to quantifying image elements.

In other words, Gemini tends to over-rationalize confusion by inventing rules, whereas Qwen fails more directly at the level of visual grounding.

The data also reveals a critical shared blind spot involving *Visual Processing (OCR)*. There exists a 100% overlap between Gemini’s OCR failures and Qwen’s *Positional and Hierarchical Confusion*, implying that when text or labels are misread, both models lose the ability to correctly interpret the spatial organization of the image.

Overall, these overlaps demonstrate that the models share common structural limitations despite differences in reasoning style and abstraction depth.

B.4.4 Gemini Reasoning Errors by Topic

The next analysis cross-references the mathematical problem categories with the reasoning failure labels. This allows us to determine whether certain mathematical topics systematically trigger specific types of logical collapse.

The data is presented as a frequency heatmap, where color intensity represents the total number of errors detected during evaluation.

Gemini The most striking observation in Figure 17 (Appendix) is that *Geometría y Figuras* constitutes the primary source of failure for Gemini 2.0 Flash.

This topic produces the highest number of errors across multiple categories, particularly:

- *False Geometric Relationships*: 76 instances.
- *Basic Counting Errors*: 75 instances.

This suggests that when confronted with complex geometric shapes, the model struggles both with correctly quantifying visual elements and with identifying accurate mathematical relationships between them.

Furthermore, *Basic Counting Errors* appear consistently across every analyzed topic, including:

- *Patrones y Secuencias*: 28 instances.
- *Lógica y Razonamiento Visual*: 31 instances.
- *Combinatoria y Probabilidad*: 25 instances.

This demonstrates that even when the model successfully captures high-level concepts, its final reasoning frequently collapses due to failures in basic visual perception or enumeration.

Another recurring theme is *Process Inconsistency*, especially prevalent in:

- *Geometría y Figuras*: 51 instances.
- *Álgebra y Aritmética Visual*: 25 instances.

This indicates that the model often begins with valid premises but gradually loses the logical coherence of its own reasoning chain.

Finally, *Unnecessary Overcomplication* appears notably in:

- *Geometría y Figuras*: 20 instances.
- *Álgebra y Aritmética Visual*: 10 instances.

This suggests that Gemini frequently attempts to apply advanced mathematical methods to problems that require only simple visual observation or elementary arithmetic.

Qwen We now analyze the relationship between mathematical topics and reasoning failures for Qwen.

The most important observation from Figure 18 (Appendix) is the overwhelming dominance of *Basic Counting Errors* across all thematic categories.

This error type peaks at:

- 83 occurrences in *Geometría y Figuras*.
- 61 occurrences in *Álgebra y Aritmética Visual*.

Unlike Gemini, which exhibited a more diverse distribution of reasoning failures, Qwen’s primary bottleneck appears foundational.

This strongly suggests that regardless of mathematical complexity, Qwen’s reasoning collapses primarily due to failures in perceiving and quantifying visual elements.

Similarly to Gemini, Qwen also struggles significantly with *False Geometric Relationships*, especially in geometry-based problems.

Interestingly, Qwen exhibits considerably lower frequencies of:

- *Unnecessary Overcomplication*.
- *Rule Hallucination*.

This suggests that Qwen adopts a more conservative reasoning style, failing earlier at the perceptual level rather than inventing complex but incorrect logical explanations.

B.4.5 Reasoning vs Visual Capability

Finally, we analyze the relationship between reasoning failures and visual capability categories across both models simultaneously.

The heatmap presented in Figure 19 (Appendix) reveals that certain visual capabilities systematically trigger distinct reasoning collapses.

The strongest concentration of errors appears in:

- *Pattern Recognition*.
- *Constraint Satisfaction*.

These categories show the highest frequencies of *Basic Counting Errors*, with 97 and 89 instances respectively.

This suggests that even when the models correctly identify patterns or logical rules, they fail during execution because they cannot accurately quantify the visual elements required to apply those rules.

Additionally, *Constraint Satisfaction* exhibits a strong concentration of *Process Inconsistency* (50 instances), indicating that increasing logical complexity causes the models to lose track of their own reasoning chains.

Geometry Area problems frequently produce:

- *False Geometric Relationships*: 60 instances.
- *Unnecessary Overcomplication*: 15 instances.

This suggests that when solving area problems, Gemini often invents unnecessary geometric equalities or resorts to advanced mathematical methods instead of using simple visual decomposition.

In contrast, *Geometry Angle* problems are much more specialized, with failures concentrated almost entirely in *False Geometric Relationships*.

Tasks involving *Spatial Folding* show a clear relationship with *Spatial Folding Deficit*. Although the raw number of errors is smaller, it represents a large proportion of the total attempts in that category.

This confirms that 3D mental manipulation tasks differ fundamentally from standard 2D reasoning problems.

On Qwen’s side, Figure 20 (Appendix) reveals that *Basic Counting Errors* dominate almost all capability categories, especially:

- *pattern_recognition*: 124 instances.
- *constraint_satisfaction*: 99 instances.

Additionally, geometric tasks frequently trigger *False Geometric Relationships*, reinforcing the conclusion that Qwen struggles to anchor its reasoning to the physical structure of images. Overall, these results demonstrate that although both models can partially understand high-level logic, their primary bottleneck remains grounded in visual perception and spatial consistency.

B.5 Conclusion

This study reveals that multimodal reasoning is a deeply interconnected process where the central challenge lies in combining pattern recognition with spatial intuition. Although both models demonstrate competence in classical geometry tasks involving formulas and deterministic calculations, they encounter a clear performance ceiling in problems requiring three-dimensional reasoning or mental object manipulation.

This suggests that current multimodal architectures possess mathematical knowledge but still lack genuine spatial intuition. The analysis of internal reasoning further demonstrates that these failures are systemic rather than random. The dominance of *Basic Counting Errors* and *False Geometric Relationships* indicates that the primary bottleneck is not advanced mathematical reasoning itself, but rather the inability to accurately perceive and quantify visual elements while maintaining logical consistency with the image structure.

While Gemini tends to over-rationalize confusion through *Rule Hallucination*, Qwen more commonly collapses at a perceptual grounding level. Nevertheless, both systems share the same structural blind spots. Ultimately, the study concludes that future progress in multimodal AI will require shifting the focus from increasingly complex reasoning architectures toward stronger numerical grounding, spatial consistency, and visual perception capabilities.

B.6 Reasoning Annex Figures

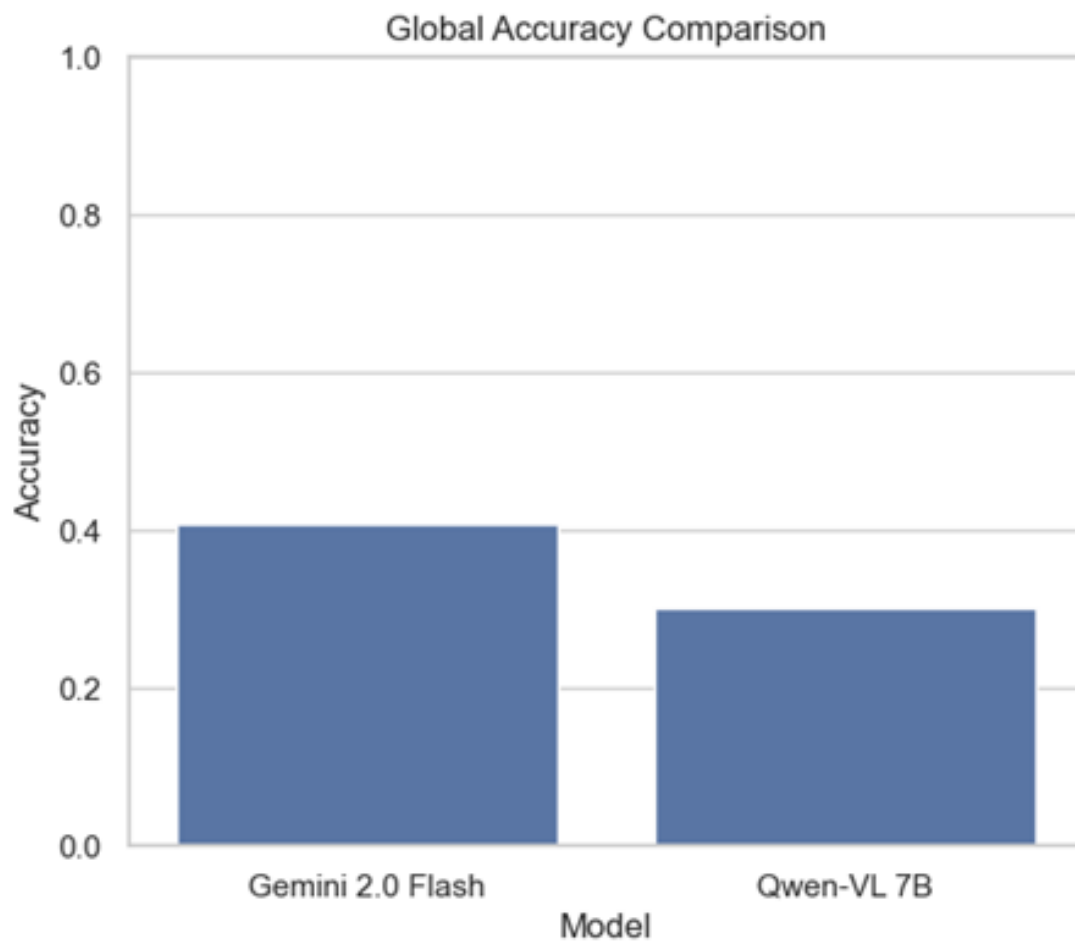


Figure 1 (Appendix).

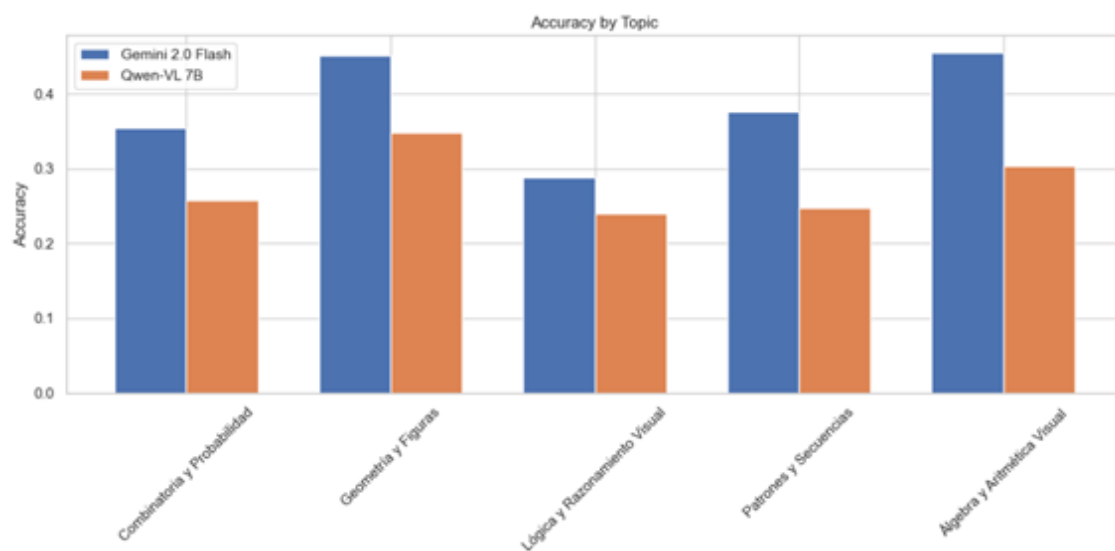


Figure 2 (Appendix).

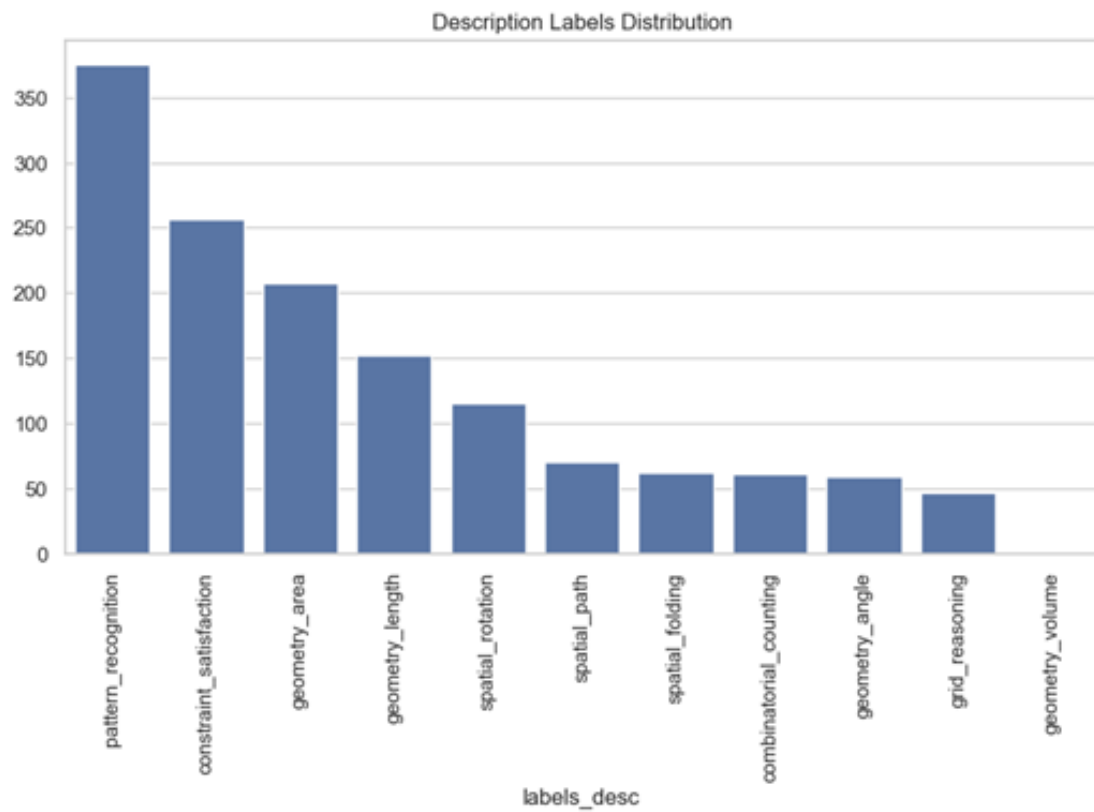


Figure 3 (Appendix).

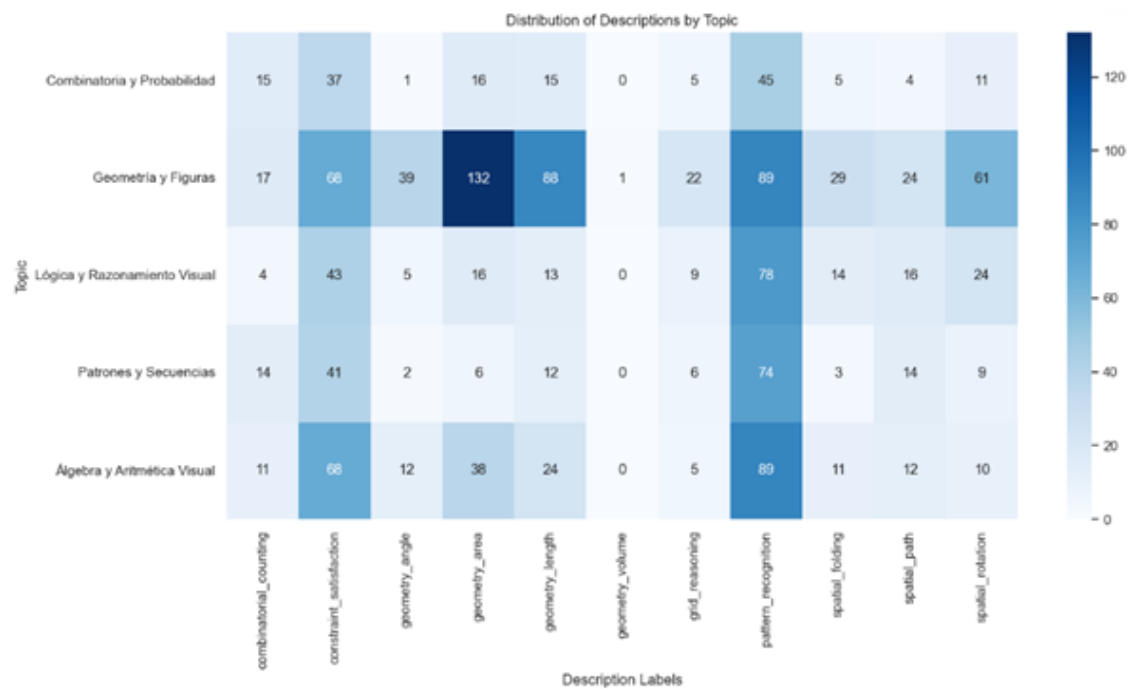


Figure 4 (Appendix).

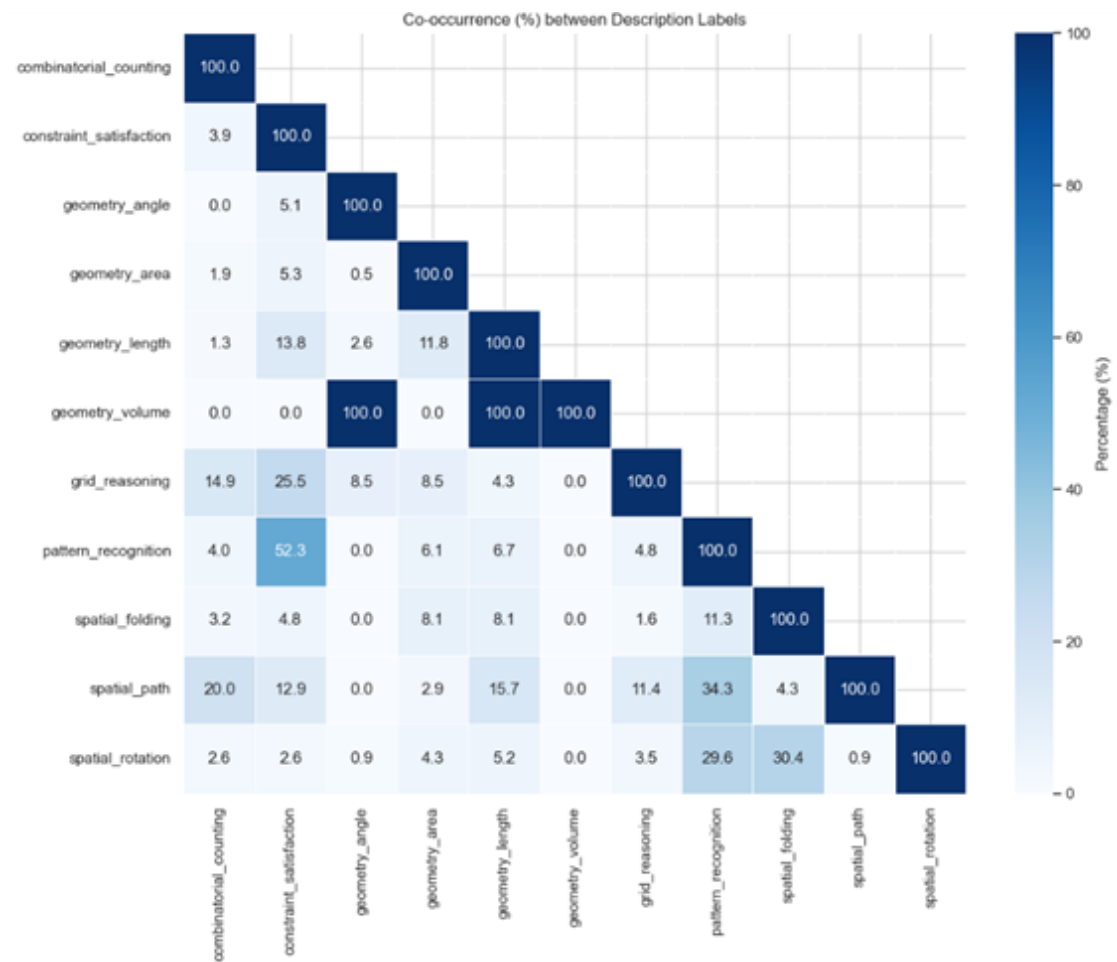


Figure 5 (Appendix).

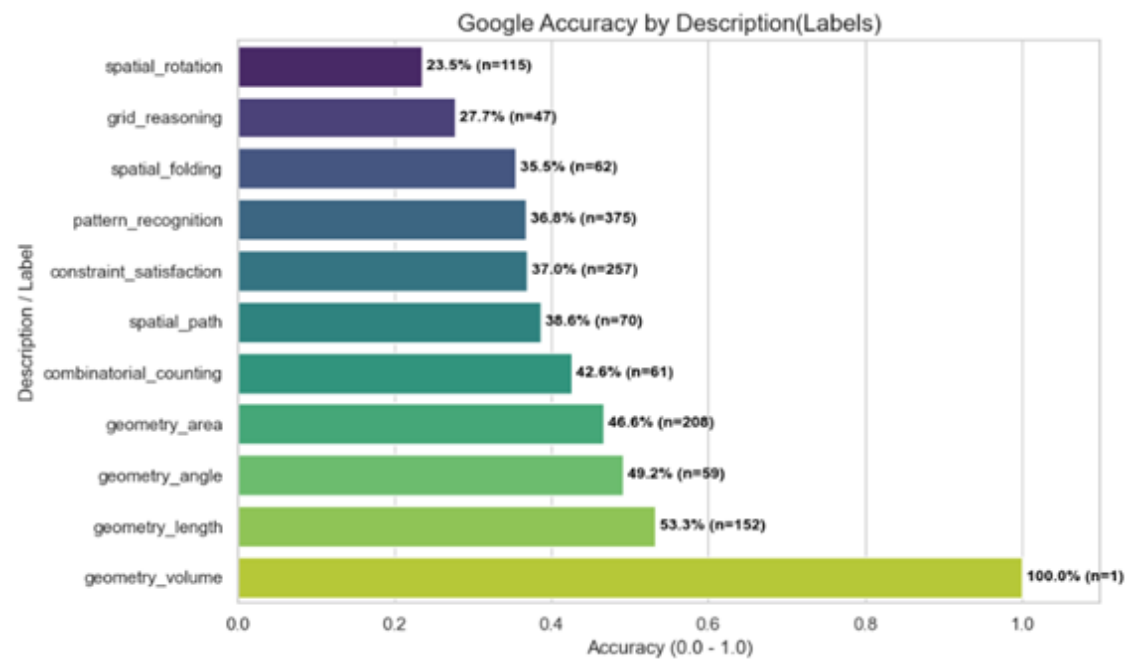


Figure 6 (Appendix).

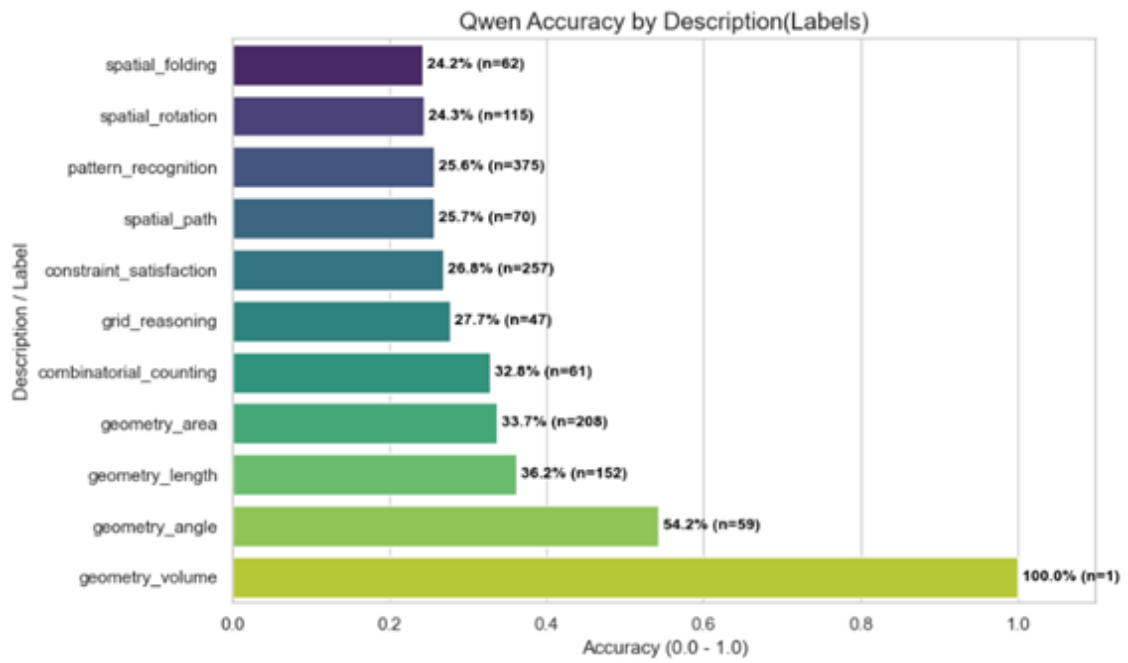


Figure 7 (Appendix).

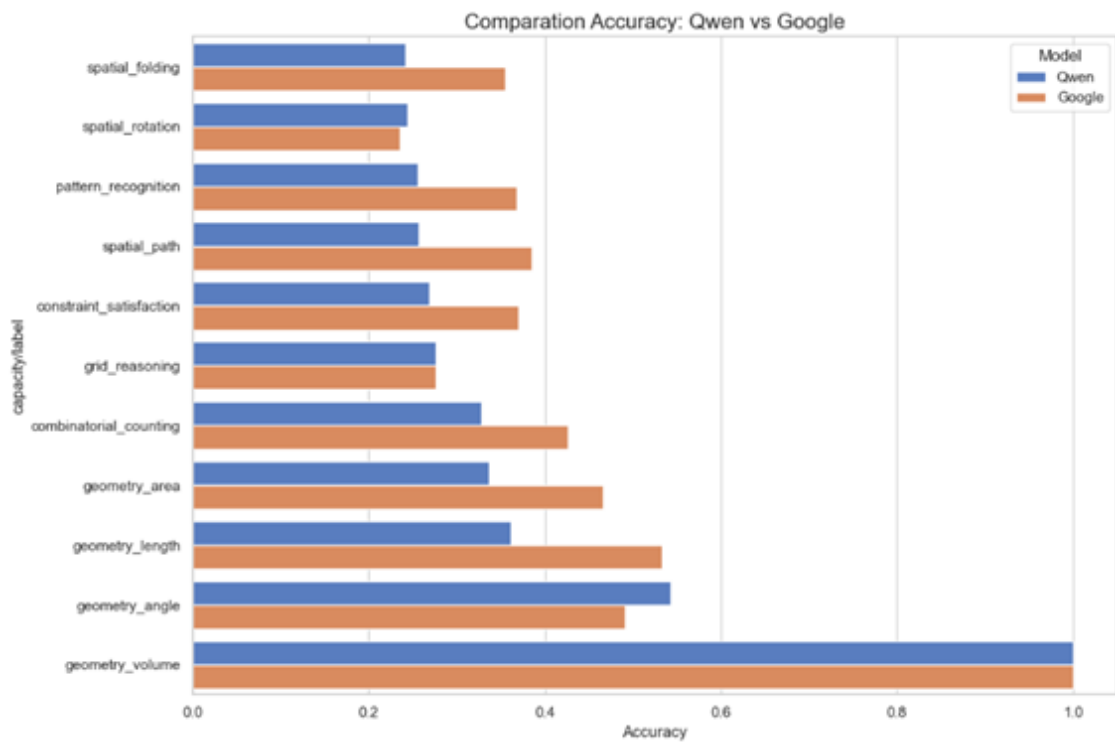


Figure 8 (Appendix).

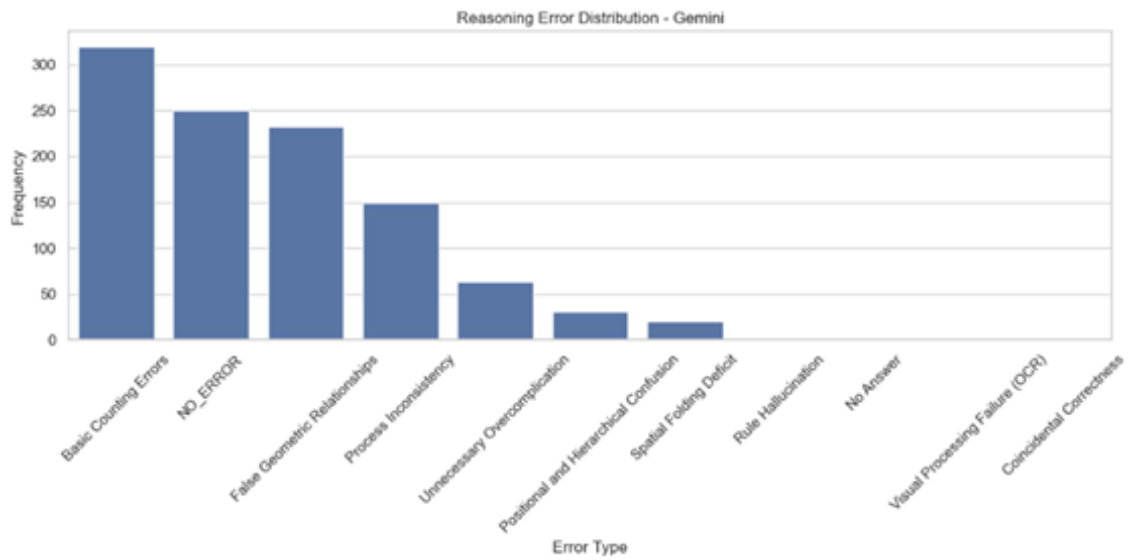


Figure 9 (Appendix).

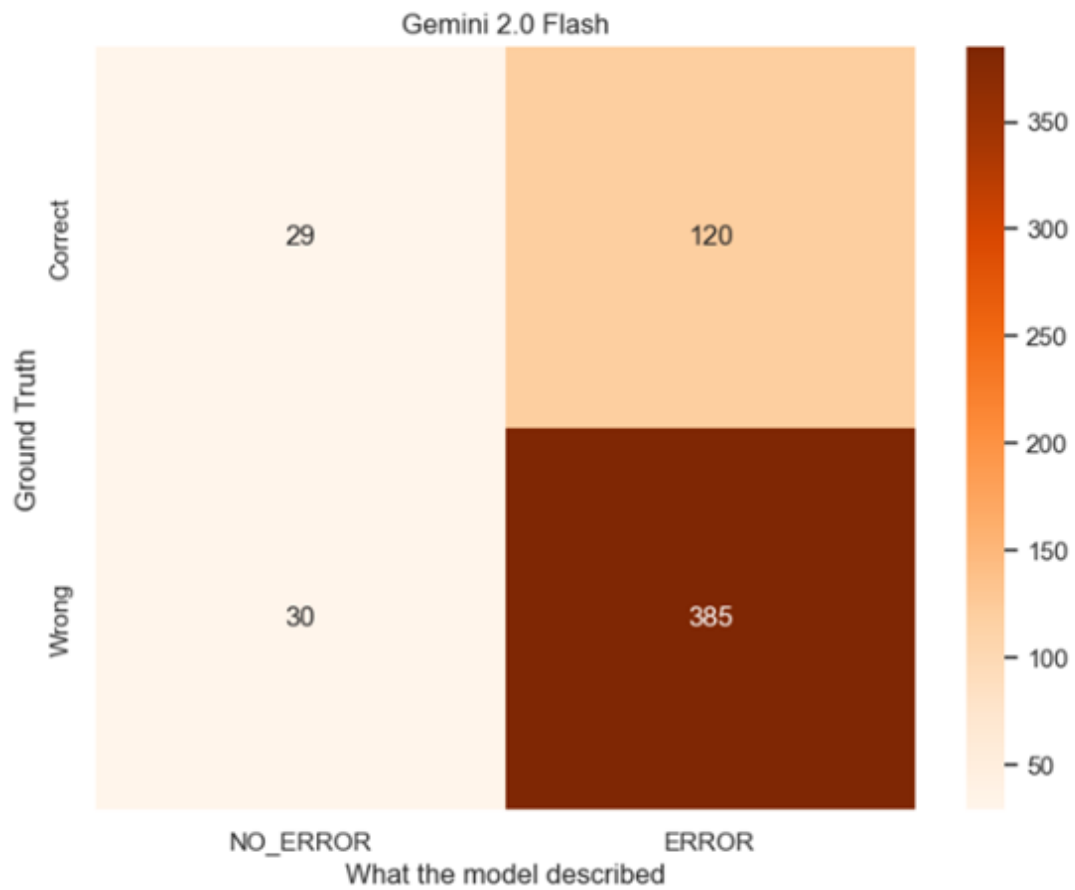


Figure 10 (Appendix).

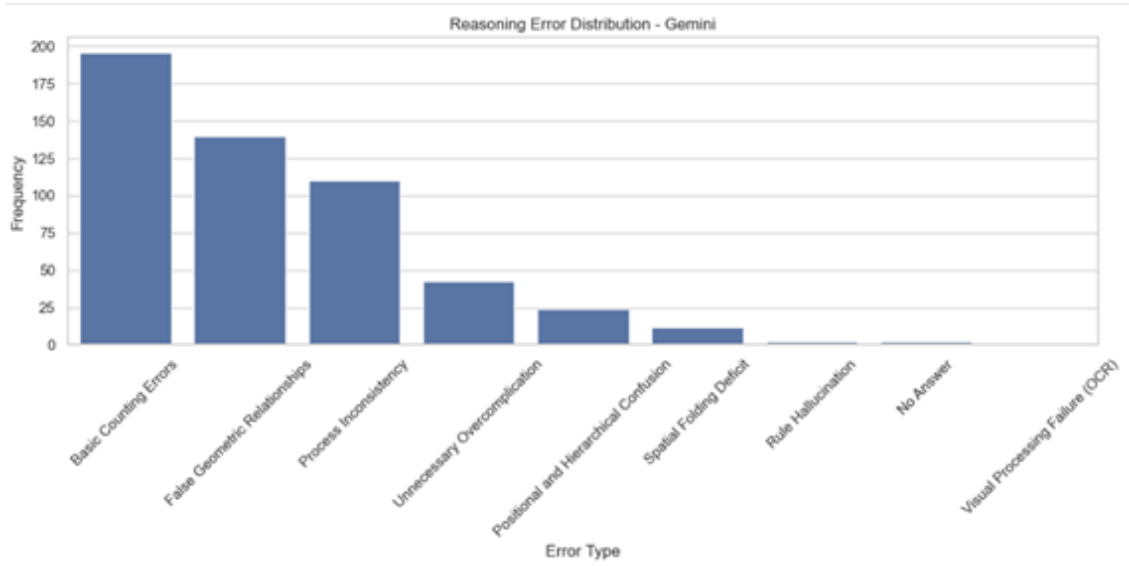


Figure 11 (Appendix).

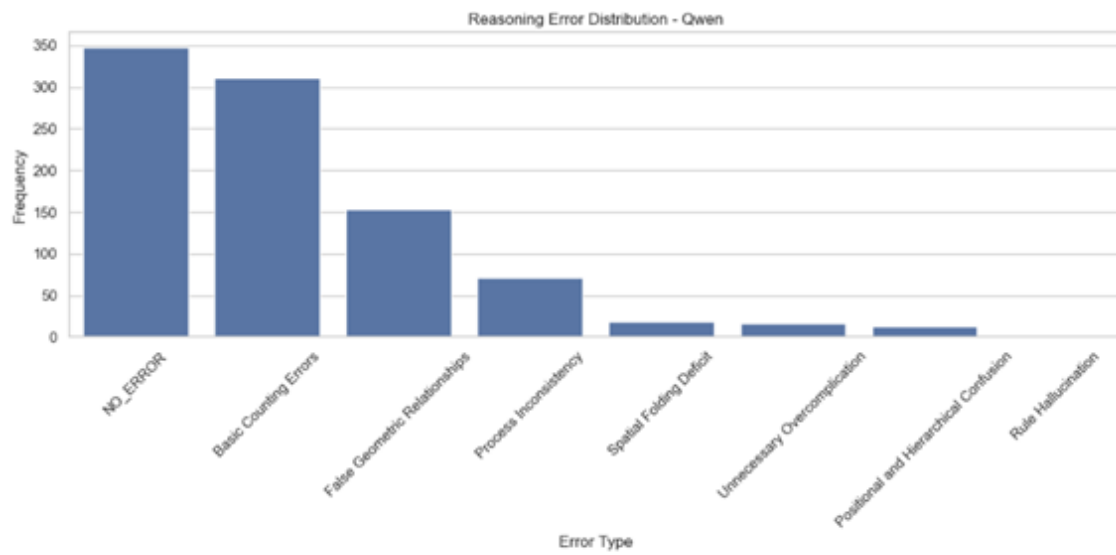


Figure 12 (Appendix).

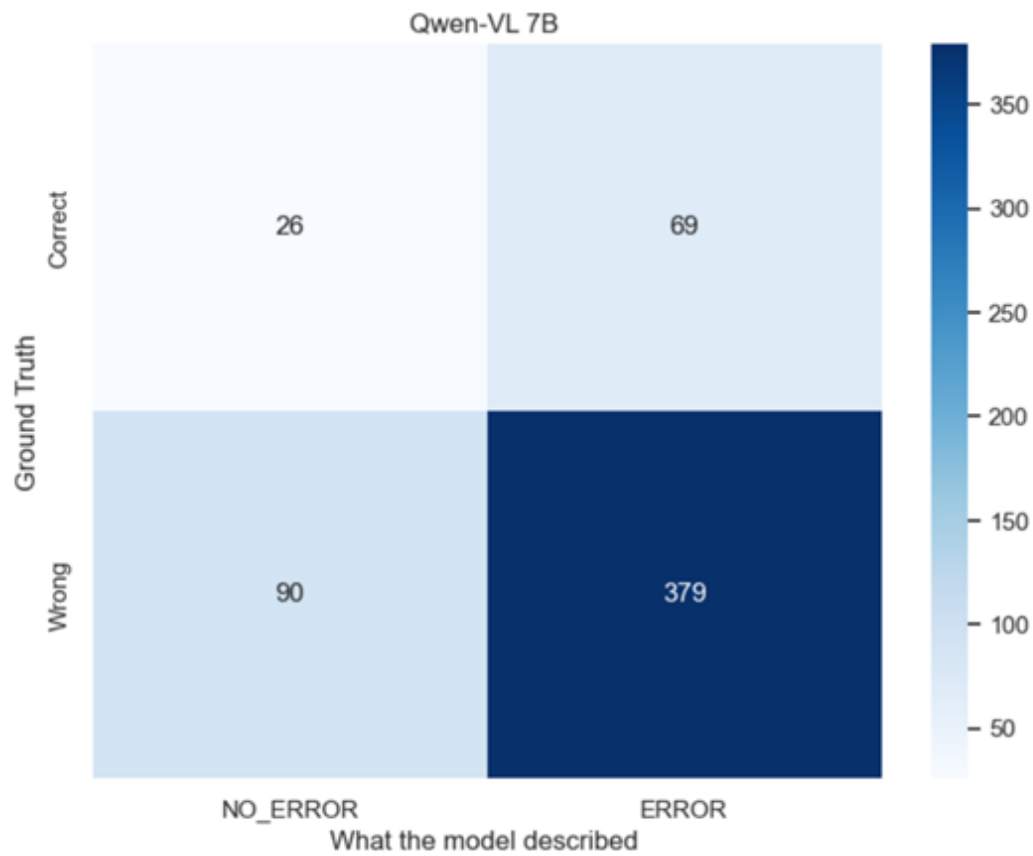


Figure 13 (Appendix).

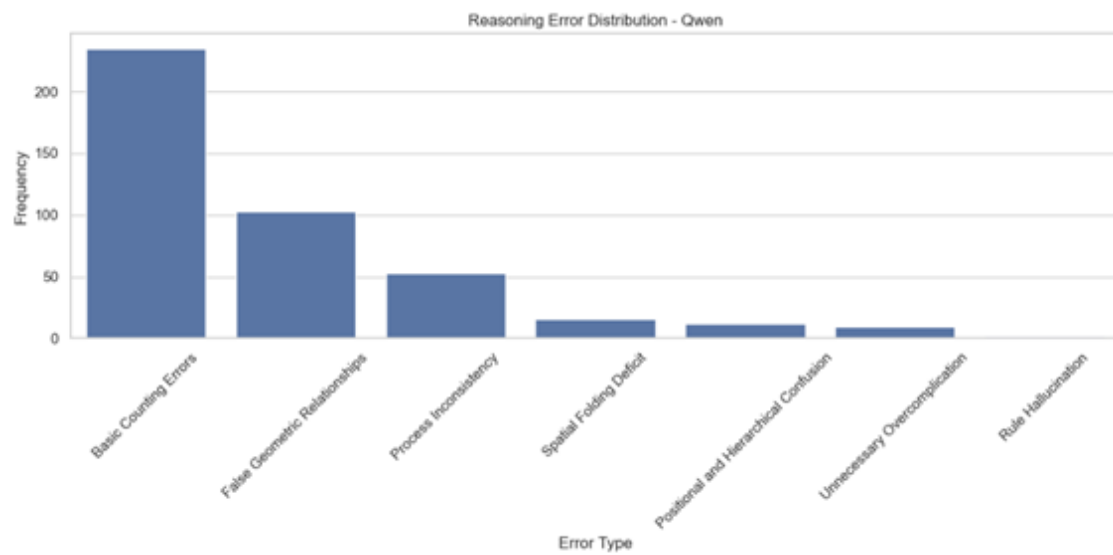


Figure 14 (Appendix).

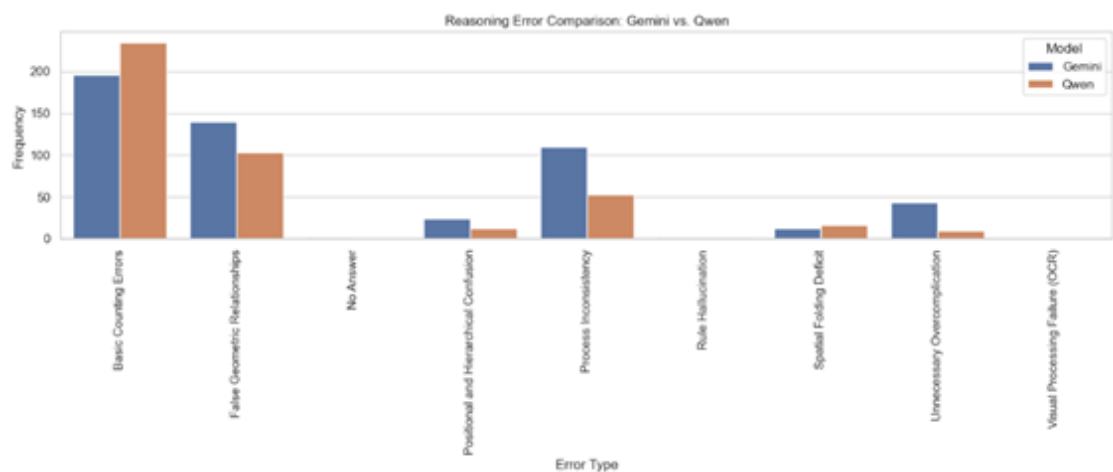


Figure 15 (Appendix).

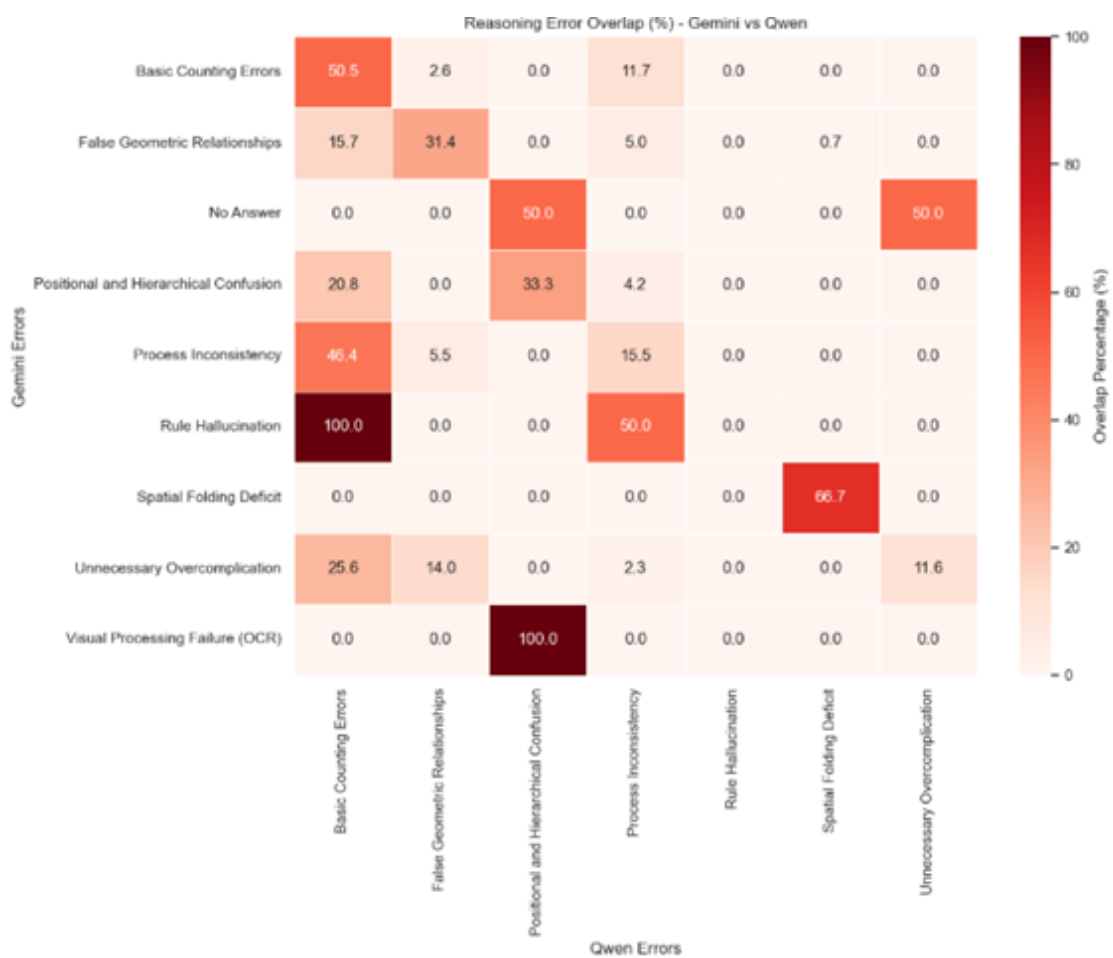


Figure 16 (Appendix).

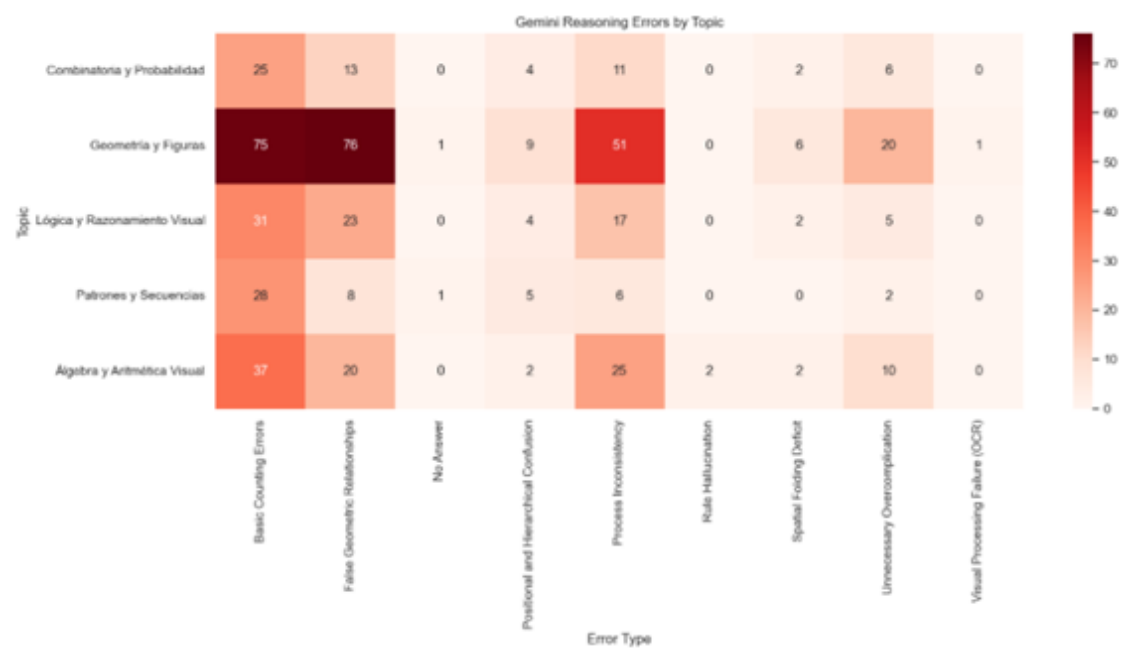


Figure 17 (Appendix).



Figure 18 (Appendix).

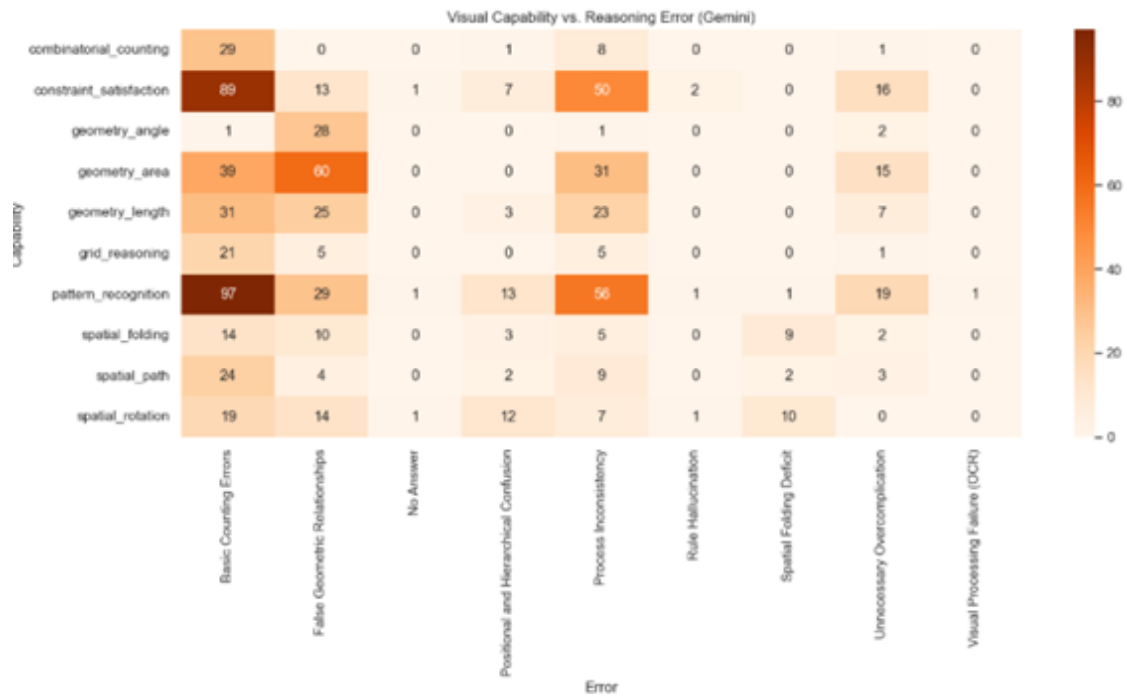


Figure 19 (Appendix).

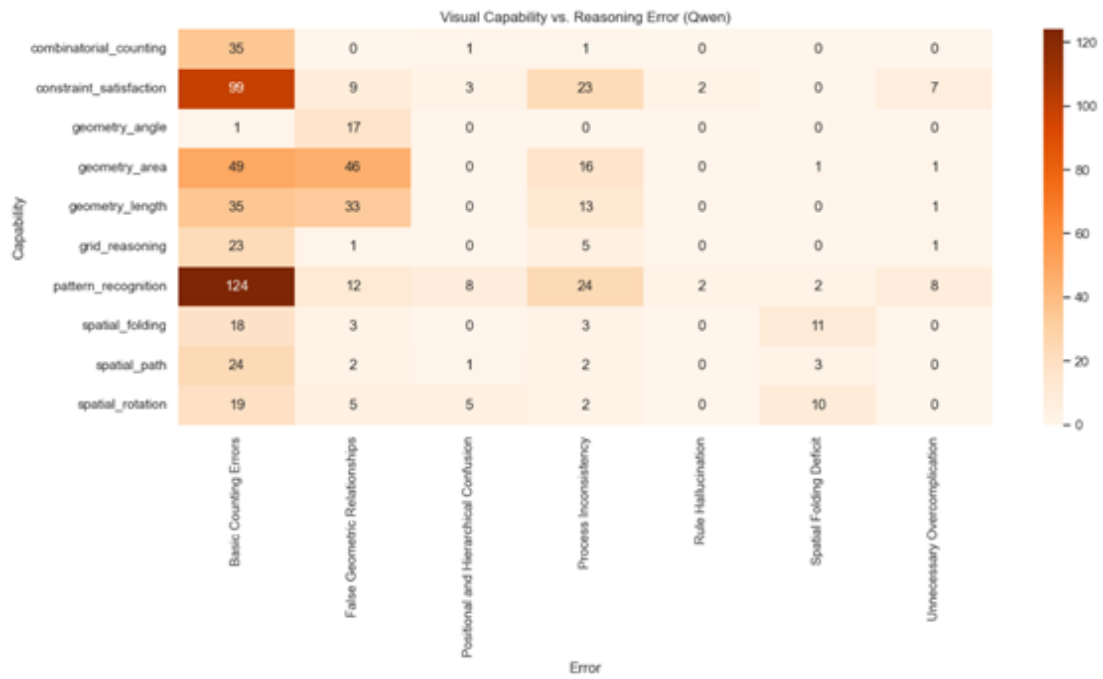


Figure 20 (Appendix).

C Pipeline appendix

C.1 Extended results

Topic	n	Accuracy	Coverage	Contradiction
Geometry and Figures	379	34.04%	37.98%	78.10%
Algebra and Visual Arithmetic	165	34.55%	40.74%	84.85%
Logic and Visual Reasoning	125	30.40%	54.92%	85.60%
Patterns and Sequences	101	28.71%	44.33%	85.15%
Combinatorics and Probability	93	27.96%	35.56%	81.72%

Table 13: Results by topic. Coverage indicates whether the correct answer appears among the options accepted by the pipeline.

Language	n	Accuracy	Coverage	Average accepted options
Valencian	407	29.48%	33.50%	1.30
French	170	35.88%	64.88%	2.90
Spanish	159	40.25%	52.23%	2.19
English	127	26.77%	20.66%	0.87

Table 14: Results by language.

C.2 Auxiliary figures

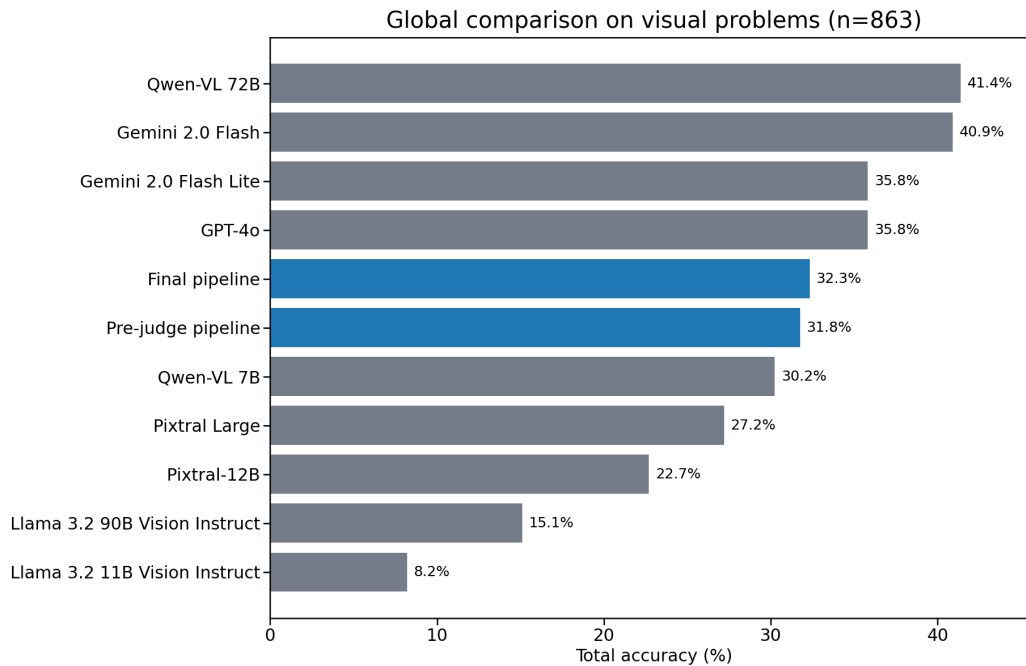


Figure 13: Extended global comparison generated from the analysis notebook.

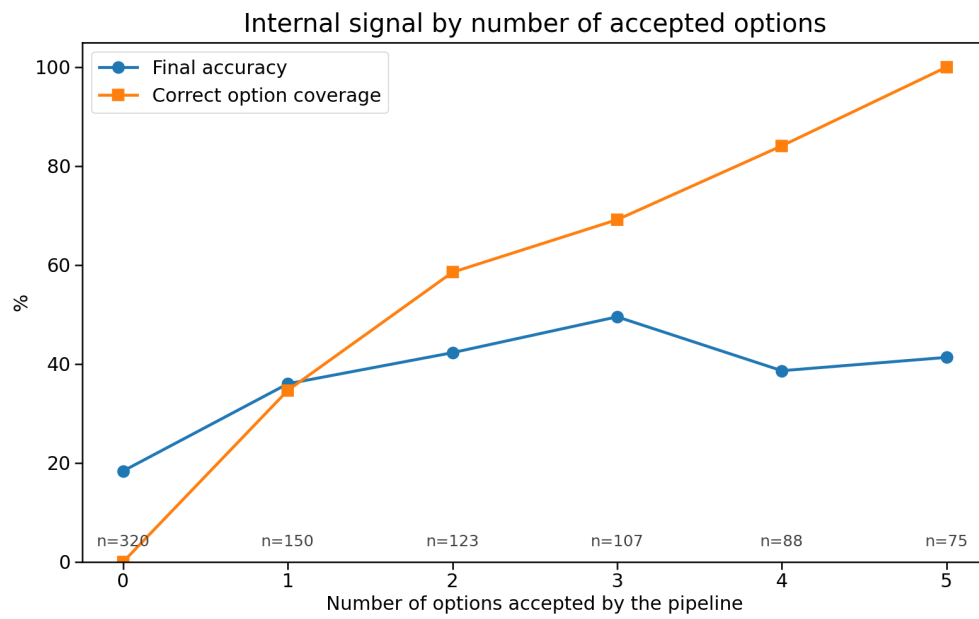


Figure 14: Relation between accepted options, final accuracy and coverage of the correct option.

D App Screenshots



Figure 15: Final answer and option-by-option justification panel.

Pasos del razonamiento

- 1 Doblar una imagen por una línea discontinua y verificar cuál de las opciones es igual a la imagen doblada.
- 2 Propiedad de simetría: una figura es simétrica si se puede doblar por una línea y coincidir con su imagen.
- 3 Propiedad de congruencia: dos figuras son congruentes si pueden superponerse exactamente.
- 4 Extraer la imagen original y la línea discontinua del enunciado.
- 5 Observar cómo se doblaría la imagen original por la línea discontinua.
- 6 Comprobar si la opción proporcionada es congruente con la imagen doblada.
- 7 Verificar si la opción es simétrica con respecto a la línea discontinua.
- 8 Decidir si la opción es correcta, incorrecta o incierta basándose en la congruencia y la simetría.
- 9 El razonamiento de la opción B indica que es congruente con la imagen doblada por la línea discontinua, con la misma forma, orientación y correspondencias de elementos. Aunque la opción C también es marcada como correcta por el modelo, el enunciado requiere una única respuesta. Al analizar la consistencia con el enunciado, la opción B describe explícitamente una figura simétrica que coincide con la imagen doblada, mientras que la opción C menciona congruencia sin especificar detalles críticos de simetría. Además, en problemas de doblado de imágenes, la simetría respecto a la línea discontinua es el criterio clave, lo que hace a B la opción más consistente y precisa según los razonamientos proporcionados.
- 10 El razonamiento de la opción B indica que es congruente con la imagen doblada por la línea discontinua, con la misma forma, orientación y correspondencias de elementos. Aunque la opción C también es marcada como correcta por el modelo, el enunciado requiere una única respuesta. Al analizar la consistencia con el enunciado, la opción B describe explícitamente una figura simétrica que coincide con la imagen doblada, mientras que la opción C menciona congruencia sin especificar detalles críticos de simetría. Además, en problemas de doblado de imágenes, la simetría respecto a la línea discontinua es el criterio clave, lo que hace a B la opción más consistente y precisa según los razonamientos proporcionados.

Clasificación y recortes

MathVision AI puede cometer errores. Verifica siempre los pasos importantes y consulta a tu profesor si lo necesitas.

Figure 16: Central panel showing the reasoning steps.

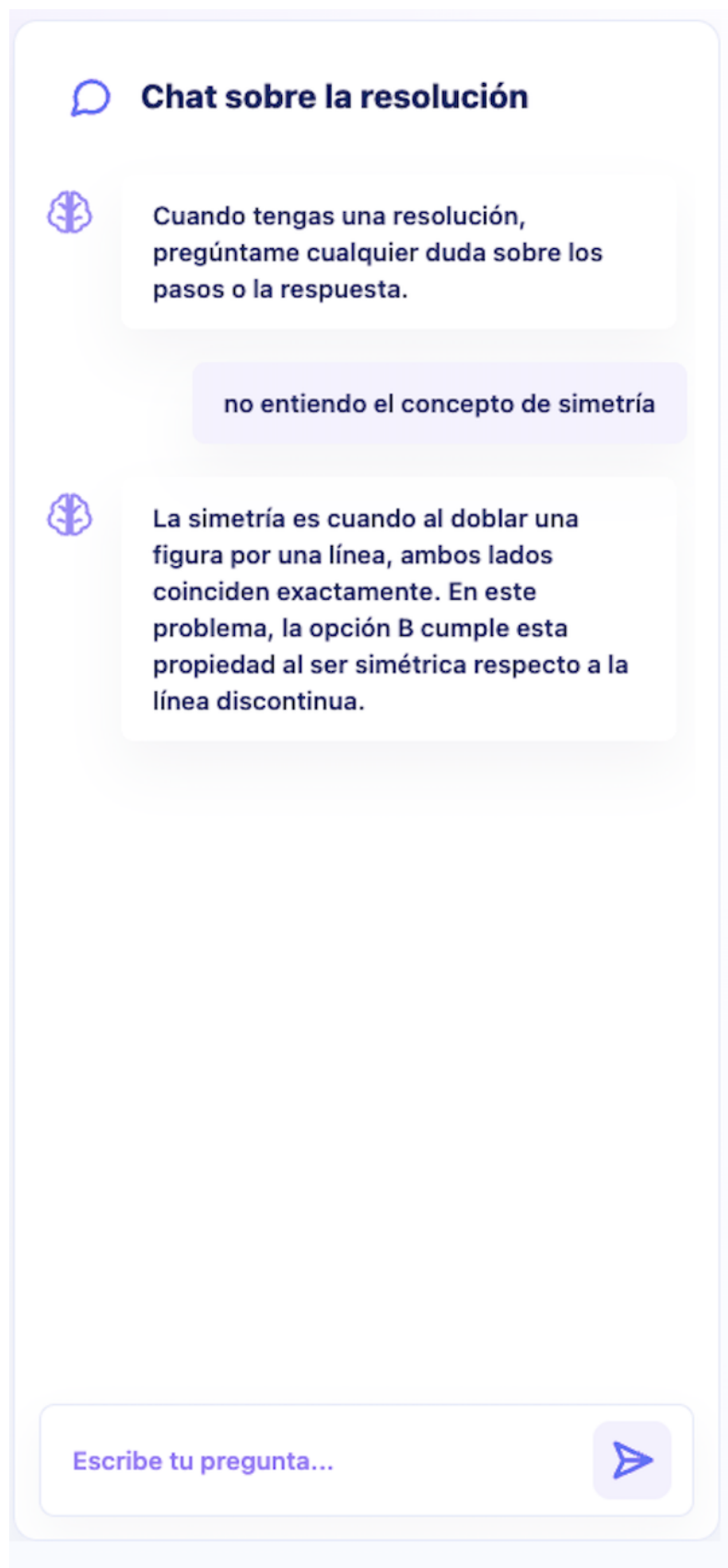


Figure 17: Interactive chat used for follow-up questions about the generated solution.