HOW FAR ARE WE FROM INTELLIGENT VISUAL DEDUCTIVE REASONING?

Anonymous authors
Paper under double-blind review

ABSTRACT

Vision-Language Models (VLMs) such as GPT-4V have recently demonstrated incredible strides on diverse vision language tasks. We dig into vision-based deductive reasoning, a more sophisticated but less explored realm, and find previously unexposed **blindspots** in the current SOTA VLMs. Specifically, we leverage Raven's Progressive Matrices (RPMs), to assess VLMs' abilities to perform multi-hop relational and deductive reasoning relying solely on visual clues. We perform comprehensive evaluations of several popular VLMs employing standard strategies such as in-context learning, self-consistency, and Chain-of-thoughts (CoT) on three diverse datasets, including the Mensa IQ test, IntelligenceTest, and RAVEN. The results reveal that despite the impressive capabilities of LLMs in text-based reasoning, we are still far from achieving comparable proficiency in visual deductive reasoning. We found that certain standard strategies that are effective when applied to LLMs do not seamlessly translate to the challenges presented by visual reasoning tasks. Moreover, a detailed analysis reveals that VLMs struggle to solve these tasks mainly because they are unable to perceive and comprehend multiple, confounding abstract patterns in RPM examples.

1 INTRODUCTION

Recent advancements in Vision-Language Models (VLMs) have showcased the success of models such as GPT4-V (OpenAI, 2023) and Gemini (Team et al., 2023) across various vision language tasks. These tasks include captioning, object localization, multimodal world knowledge and commonsense, visual question answering (VQA), and vision-based coding (Yang et al., 2023). Previous evaluations of these models have proven that state-of-the-art (SOTA) VLMs are capable of performing well in numerous vision-based reasoning and understanding tasks (OpenAI, 2023; Team et al., 2023). Notably, prior works have demonstrated that strong VLMs can accurately extract text from images, understand and reason with charts and tables, and solve simple visual math problems (Yang et al., 2023; Nahida Akter et al., 2023).

In this study, we aim to evaluate the limitations of VLMs on challenging tasks that demand sophisticated vision-based deduction abilities, an area that has been relatively unexplored. Specifically, we ask the models to complete a set of Raven's Progressive Matrices (RPMs) problems (Kunda et al., 2013; Zhang et al., 2019), which are frequently used to measure human intelligence, by identifying the correct pattern to fill in the blank from multiple options. See Figure 1 for illustration. This requires the models to 1) comprehend each given pattern including the choices, 2) deduce underlying rules and identify any trend that can explain the evolution of these patterns, and 3) employ the learned rules to choose the missing pattern from the given options. The model's capacity to handle each aspect must be effectively coordinated to provide the correct answer. Our findings reveal that although some problems may seem intuitive to humans, they might not be as intuitive to VLMs.

Compared to standard image reasoning tasks like VQA (Antol et al., 2015), RPMs pose several unique challenges: 1) RPMs require sophisticated deductive capabilities that involve multi-hop comparative reasoning, such as discrimination, relation, and analogy, while VQA typically requires only few steps of reasoning, 2) RPMs rely solely on visual clues to generate hypotheses and verify them, while VQA often involves using natural language to infer the objective and determine which parts to focus on, 3) RPMs are inherently few-shot (mostly 2-shot) learning tasks. Each RPM problem may have different underlying rules, which demands strong generalization abilities to solve them.

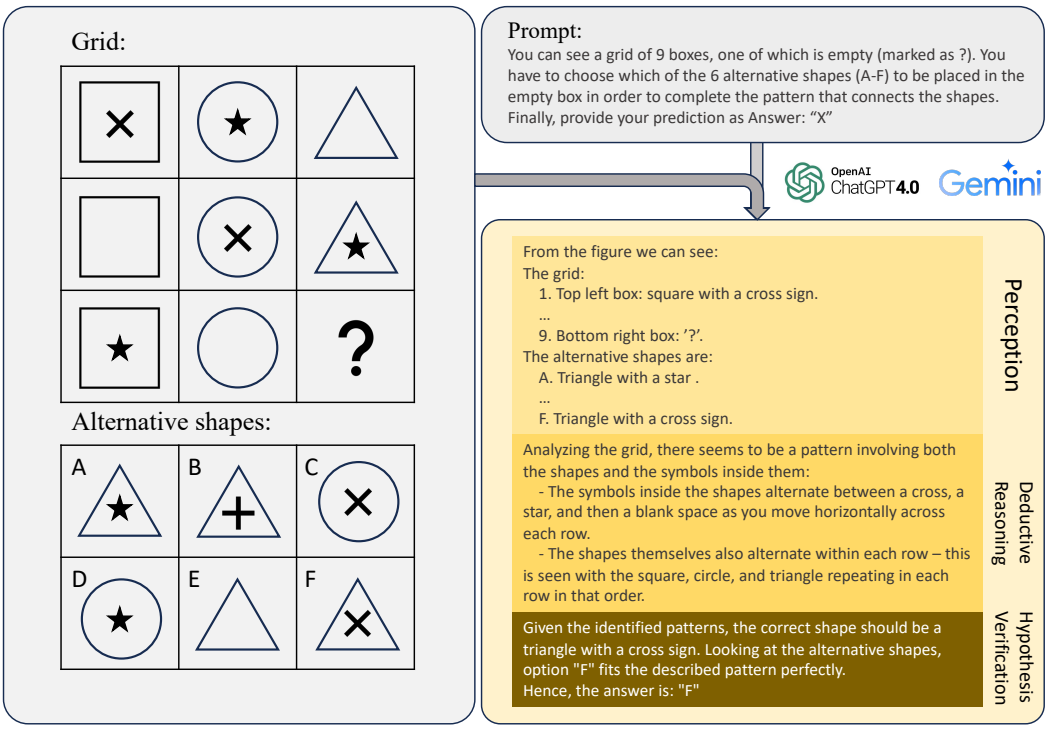


Figure 1: Illustration of the visual deductive reasoning for Raven's Progressive Matrices. The task requires intricate coordination among perception, deductive reasoning, and hypothesis verification capabilities exhibited by Vision-Language Models.

Humans have a remarkable ability to learn from just a few examples, and powerful language models like LLMs have demonstrated this ability in text-based tasks. However, the ability of strong VLMs to solve few-shot reasoning tasks by relying solely on visual cues has not been well studied.

As an emerging field, it is crucial to establish benchmarks and systematic evaluations in order to push the limits of the visual deductive ability of VLMs. Our contributions include:

1. We set up a framework for systematically evaluating VLMs on RPM problems. We evaluated several SOTA open-source and closed-source VLMs on three diverse datasets, including the Mensa IQ test, IntelligenceTest, and RAVEN, providing a comprehensive assessment of their performance. The results indicate that although LLMs exhibit impressive capabilities in text-based reasoning, such proficiency has not been achieved in image-based reasoning.

2. We employed standard inference-time strategies in LLMs such as in-context learning (Brown et al., 2020) and self-consistency (Wang et al., 2022) to probe the potential of VLMs. We found that some standard strategies that are effective in LLMs do not seamlessly translate to the VLMs we used.

3. We finely diagnose the performance bottleneck of VLMs by breaking down their capability into *perception*, *deductive reasoning*, and *hypothesis verification*. Our analysis reveals that perception is the limiting factor in current VLMs. To scrutinize this specific "blind spot" in strong VLMs such as GPT-4V, we provide a case study highlighting where issues occur.

4. We identified and examined several issues associated with the current VLMs in this task. These issues include overconfidence, sensitivity to prompt design and an inability to effectively leverage in-context examples. We ablated the effects of different prompts on the overall performance of the model and found models can benefit from more *structured* prompts.

2 RELATED WORK

General LLM Reasoning benchmarks Many text-based reasoning tasks and benchmarks have been introduced to evaluate LLMs in various domains (Huang & Chang, 2022) such as general knowledge (Hendrycks et al., 2020), math reasoning (Cobbe et al., 2021), commonsense reasoning (Geva et al., 2021; Clark et al., 2018), factual reasoning (Laban et al., 2023), and coding (Chen et al., 2021). Some noteworthy examples of these works are BIG-bench (Srivastava et al., 2022), HELM (Liang et al., 2022), SuperGLUE (Sarlin et al., 2020), and LAMA (Petroni et al., 2019).

Visual reasoning evaluation Previous work on visual reasoning tasks has primarily focused on tasks such as visual question answering (VQA) Antol et al. (2015) and image captioning. These tasks involve answering questions about images or generating natural language descriptions of visual content. Researchers have also examined the ability of models to understand the relational and compositional aspects of objects in images. Datasets like CLEVR (Johnson et al., 2017) and SHAPES (Andreas et al., 2016) assess visual reasoning abilities such as counting, comparing, logical reasoning, and storing information in memory. As the VLMs abilities to perform visual reasoning have evolved so have the benchmarks. New benchmarks, like MMMU (Yue et al., 2023) and MathVista (Lu et al., 2023) have been developed that test the models' ability to emulate human-like understanding of scenes and objects in images and videos. These benchmarks include areas such as scene text understanding (Sidorov et al., 2020), formulation (Lu et al., 2024), table and chart interpretation (Lu et al., 2024), the comprehension of visual stimuli (Yang et al., 2023), geometric reasoning (Ahrabian et al., 2024), spatial reasoning (Chen et al., 2024), and facial expression comprehension and reasoning (Yang et al., 2023).

Deductive reasoning Deductive reasoning evaluation and benchmarks have been conducted for both textual and visual domains. Two notable examples are GuessWhat?! (De Vries et al., 2017) and ReferIt (Kazemzadeh et al., 2014), which assess the visual reasoning abilities of the models being tested. More recently, LMRL Gym (Abdulhai et al., 2023) and Entity Deduction Arena (Zhang et al., 2023) have been introduced as methods to evaluate the ability of LLMs to perform multi-turn deductive reasoning tasks. Another relevant task is ARC (Acquaviva et al., 2022) which shares similarities with RPMs, as they both require correctly inferring unseen outputs based on given examples. Comparing with ARC, RPMs are abstract and requires intricate analogical and relational reasoning.

3 EXPERIMENT SETTING

3.1 DATASET

In our paper, we employed three RPMs datasets. The **Mensa Test**[1] consists of 35 questions with progressive levels of difficulty. For the purpose of 1-shot learning, we used the first question as an in-context example and reserved the remaining 34 questions for evaluation. The **IntelligenceTest** (IT)[2] provided an IQ test encompassing verbal, pattern recognition, math, and structural components. We specifically focused on pattern recognition, which solely comprised RPMs problems and included 66 examples. Additionally, we incorporated the **RAVEN** dataset (Zhang et al., 2019) for evaluation. The RAVEN dataset employs a generative model to create RPMs problems using a hierarchical pipeline. The test dataset of RAVEN contains 14,000 examples, covering 7 types of distinct figural configurations that incorporate different layouts, shapes, and relational structures. In this work, we generate 140 new samples, 20 samples for each figural configuration.

3.2 MODELS

We compared various VLMs that represent the state-of-the-art for both closed-source and open-source models, including GPT4-V (gpt-4-vision-preview) (OpenAI, 2023), Gemini-pro (Team et al., 2023), Qwen-VL-Max (Bai et al., 2023) and LLaVA-1.5-13B (Liu et al., 2023). We use the default sampling method for each of the tested VLMs in our generation process, show in Appendix A.2.

[1] https://www.mensa.org/public/mensa-iq-challenge

[2] https://www.intelligencetest.com/questions

3.3 PROMPTS

We prompt the model with the instruction followed by the query image. For the Mensa set we used the following prompt:

> *You can see a grid of 9 boxed, one of which is empty (marked as ?). You have to choose which of the 6 alternative shapes (A-F) should be placed in the empty box in order to complete the pattern that connects the shapes. Finally, provide your prediction as Answer: "X".* {*query image*}

Appendix A.1 contains the prompts used for all three datasets for different settings. In Section 5.4, we explain why we chose the particular order of the prompt and query image order.

4 EVALUATION RESULTS

4.1 EVALUATION OF VLMS ON VISUAL DEDUCTIVE REASONING

	Mensa		**IntelligenceTest (IT)**		**RAVEN**	
	Entropy	**Accuracy↑**	**Entropy**	**Accuracy↑**	**Entropy**	**Accuracy↑**
GPT-4V	1.49	0.24 ± 0.05	1.40	0.16 ± 0.04	2.07	0.12 ± 0.04
Gemini Pro	1.24	0.15 ± 0.04	1.18	0.18 ± 0.03	1.37	0.11 ± 0.04
QWen-VL-Max	1.13	0.17 ± 0.01	0.97	0.13 ± 0.02	0.48	0.10 ± 0.03
LLaVA-1.5-13B	0.72	0.23 ± 0.01	0.64	0.09 ± 0.01	0.25	0.10 ± 0.03
Random Guess	2.58	0.16	2.58	0.16	3.00	0.12

Table 1: Benchmark of VLMs on three different datasets. "Entropy" denotes uncertainty of the prediction, and "Accuracy" indicates the percentage of accurately answered questions.

In Table 1 we show how different VLMs performed on each dataset. For each model and dataset, we computed the statistics by averaging them over 10 repetitions. From the table, it is evident that GPT-4 either slightly surpasses or is on par with the other models across all benchmarks. However, the accuracy gap between the models is not substantial in terms of their ability to solve RPM puzzles. It is interesting to note that the performance of these models is comparable to random guessing (last row), indicating their limited effectiveness in this area. Converting the accuracy on the questions to human ranking scale, we find that the models rank in the 2-8 percentile on the Mensa tests. On the IT dataset humans demonstrate a wide range of success rates per question, spanning from 30% to 93.4%, which is much higher than the highest accuracy of a mere 18% observed for Gemini Pro. Similarly, on the Raven dataset humans attain an impressive success rate of 84.67% (Zhang et al., 2019), starkly outperforming VLMs, which consistently yield results akin to random guessing.

Uncertainty of the prediction We analyze the entropy of model predictions in order to assess the uncertainty inherent in their predictive distribution. For the choices set C, the Entropy is defined as $S = -\sum_{i \in C} p_i \log p_i$. If the model consistently predicts a single answer, it is has an entropy of 0. If it randomly guesses, the entropy reaches the upper bound shown in the Table 1. We see that GPT-4 and Gemini Pro exhibit a greater diversity of answers, which is also reflected in the greater diversity in recognizing and attempting to identify various patterns. On the other hand, LLaVA and QWen-VL produce more deterministic predictions, resulting in lower entropy.

We observed that all the VLMs tend to be confident while being wrong. Interestingly, we observed that even when the entropy was high, models tried to provide a nonsensical rational, instead of acknowledging their inability to perform the task; this was observed to happen more often with models that had higher entropy. All the tested models never express any level of uncertainty by using words like "likely" or "maybe". This excessive confidence can presumably be attributed to the model pretraining and instruction finetuning steps, which typically do not involve calibrating the model for uncertainty. Instead, the models are encouraged to generate uncertain content, leading to more errors in aggregating in the generated output.

4.2 DO SOME STANDARD STRATEGIES USED IN LLMS TRANSLATE EFFECTIVELY TO VISUAL DEDUCTIVE REASONING?

We tried two strategies that have been shown to be effective in various tasks related to text LLMs: 1) **One-shot** (1-shot) (Brown et al., 2020) involves providing a single in-context RPM example and its solution to the VLMs. 2) **Self-consistency** (SC) (Wang et al., 2022) entails sampling multiple responses and selecting the answer that receives the majority vote. The results are shown in Table 2.

	Mensa		**IntelligenceTest**		**RAVEN**	
	Entropy	**Accuracy↑**	**Entropy**	**Accuracy↑**	**Entropy**	**Accuracy↑**
GPT-4V (0-shot)	1.49	0.24 ± 0.05	1.40	0.16 ± 0.04	2.07	0.12 ± 0.04
GPT-4V (1-shot)	1.41	0.22 ± 0.06	1.31	0.17 ± 0.04	2.03	0.12 ± 0.04
GPT-4V (SC)	0.17	0.31 ± 0.01	0.15	0.19 ± 0.02	0.20	0.10 ± 0.02
Gemini Pro (0-shot)	1.24	0.15 ± 0.04	1.18	0.18 ± 0.03	1.37	0.11 ± 0.04
Gemini Pro (1-shot)	0.69	0.17 ± 0.03	0.54	0.19 ± 0.01	1.35	0.10 ± 0.03
Gemini Pro (SC)	0.03	0.18 ± 0.01	0.03	0.18 ± 0.01	0.08	0.10 ± 0.01

Table 2: Expanded benchmark of VLMs on three different datasets, including the 1-shot and SC variants for both GPT-4 and Gemini models. The prompts are provided in Appendix A.1.

VLMs struggle with reading in-context image The performance of the 1-shot evaluation did not demonstrate a significant improvement compared to the 0-shot evaluation. Specifically, we observed only a marginal 1% enhancement for the IntelligenceTest dataset, while encountering a decrease of 2-4% in accuracy for the Mensa test. Surprisingly, all the tested models, including GPT-4V and Gemini, struggle with a high failure rate *even when the in-context example is identical to the current task being solved*. This is peculiar because powerful LLMs usually exhibit the ability to analogize and copy the in-context example when provided with the same query. We observed accuracy ranging from 10% to 20% for these in-context examples across different datasets, which is comparable to the accuracy when a different example is used as the in-context example.

In order to make this observation concrete we present an ablation experiment with a specific example we created manually in the style of Mensa problems, which we call *M-easy* (See Figure 2a for the problem and Table 3 for a summary of results). Here the same example is used as the in-context example, and as the task being solved, the model only needs to be able to draw a comparison between the in-context example and the query, and copy over the answer from the in-context sample[3].

We first cast the problem as a text-only problem using appropriate descriptions for both the in-context example and the query (row 1). The model demonstrates a perfect accuracy of 100% showing that it is easy for it to solve this problem when it is represented as text. Next, we added the image to the textual description for both the in-context example and the query. The accuracy now decreases to 80%, even though additional visual information has been provided (row 2). Finally, when the text description is removed from the query, the accuracy significantly drops to 20% (row 3). We hypothesize that the drop in accuracy arises because it is much harder for the model to compare image tokens than it is to compare textual tokens and also that the model utilizes text more than it does the images.

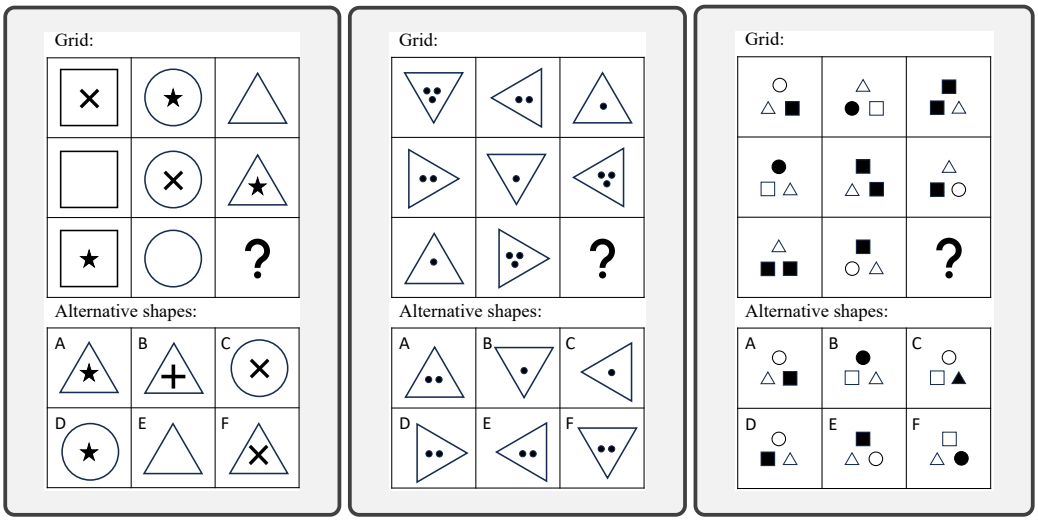
In-context	Query	Accuracy
Desc. + Rat. + Ans.	Desc.	100%
Img. + Desc. + Rat. + Ans.	Img. + Desc.	80%
Img. + Desc. + Rat. + Ans.	Img.	20%
Img.	Img. + Desc.	80%
Img.	Img.	40%

Table 3: GPT-4V analogizes better when solely based on text descriptions. Desc., Rat., Ans. and Img. represents description, rationale, answer and image, respectively

5 WHAT LIMITS THE PERFORMANCE OF THE VLMS?

We investigate why VLMs fail to reach human-level performance in answering even simple questions that are intuitive to humans. For this purpose, as a case study, we manually created three RPMs with varying degrees of difficulty, as depicted in Figure 2. To conduct a fine-grained analysis and

[3]The results are based on 10 repetitions



(a) *M-Easy* RPM Problem (b) *M-Medium* RPM Problem (c) *M-Hard* RPM Problem

Figure 2: Three manually created RPM problems evaluated for text description augmentation, illustrating varying levels of difficulty: easy, medium, and hard. The correct answers are "F, F, F".

Position	Description of *M-Easy* RPM	Description of *M-Medium* RPM	Desc. of segmented *M-Medium* RPM
Top left	A square with an X inside.	Triangle pointing down with three dots forming a vertical line in the center.	Inverted triangle with three dots inside.
Top center	A circle with a star inside.	Triangle pointing right with three dots forming a horizontal line along the center.	Right-pointing triangle with two dots.
Top right	An empty triangle.	Triangle pointing up with four dots forming a vertical line in the center.	Upright triangle with one dot in the center.
Middle left	A square with an X inside.	Triangle pointing down with two dots forming a horizontal line in the middle.	Right-pointing triangle with two dots.
Middle center	A circle with an X inside.	Triangle pointing right with a single dot in the center.	Inverted triangle with one dot in the center.
Middle right	A triangle with an X inside.	Triangle pointing up with two dots forming a vertical line along the center.	Left-pointing triangle with three dots.
Bottom left	A square with a star inside.	Triangle pointing down with one dot in the center.	Upright triangle with one dot in the center.
Bottom center	A circle.	Triangle pointing right with two dots forming a horizontal line in the middle.	Right-pointing triangle with three dots.

Table 4: The M-Easy and M-Medium RPMs descriptions from GPT-4V for the patterns can contain errors, including hallucinations and Chimera descriptions. When the model is provided with segmented RPM images (i.e., when patterns are separated into multiple image inputs), it leads to a reduction in the error. Errors are indicated in red.

diagnosis of the VLM's inability to perform this task of visual deductive reasoning with RPMs, we decompose the evaluation into three consecutive stages:

- **Perception**: assess if the model can understand and describe the patterns in the RPMs.
- **Deductive reasoning**: evaluate if the model can discern and articulate underlying rules.
- **Hypothesis verification**: examine the model's proficiency in formulating a plausible hypothesis for the missing pattern and identifying a matching option among alternatives.

5.1 How good is the VLM's perception on this task?

We first asked the model to describe the RPM figures, to assess if they understood the images that were provided as part of the problem. Surprisingly, even though VLMs are astoundingly accurate in describing commonplace images, they seemed to be quite unsuccessful at accurately describing

even the simpler abstract patterns we gave them. The generated descriptions contained numerous errors *across all the tested models*, as exemplified by results from GPT-4V in Table 4.

We identified two major issues for this *blind spot* of VLMs:

1. *Compounding error*: Models tend to replicate the descriptions of previous patterns, leading to an autoregressive amplification of compounding errors in successive descriptions. This results in an increasingly erroneous narrative throughout the generation process. For example, in Table 4 (M-Medium), When the model first makes a mistake by including "a vertical line" in the description, the subsequent text follows the same error. We think that the autoregressive nature of the VLMs causes it to repeat itself, with the preceding text dictating the entire follow-up text.
2. *Confounding error*: The similarities between patterns cause confusion, as the model struggles to maintain focus on a single pattern. Consequently, we often observe "Chimera descriptions" that erroneously combine elements from multiple patterns. For example, in Table 4 (M-Easy, middle right), the description seems to combine elements in two adjacent patterns (middle center, middle right). This could be attributed to the model's failure to effectively focus its attention on the corresponding pattern when all the patterns appear similar.

These two issues are prevalent across all the tested methods and dataset. When the patterns contain multiple elements and are more detailed, these issues become more severe.

Can decomposing the RPMs into each single pattern from the grid enhance perception? Presumably, by decomposing the patterns into individual components, we can eliminate the *confounding* errors. To investigate this, we first segmented each of the three manual examples shown in Figure 2, into 9 individual question patterns and 6 candidate patterns. We then used a new prompt A.1 for GPT-4V to read both the full image and the segmented patterns to infer the answer. In this way, we found GPT-4V would describe each pattern more accurately. The descriptions of the *M-Medium* RPM can be found in Table 4. We conducted 10 tests for each RPM and report the accuracy comparison with and without segmentation in Figure 3. We also verify the segmentation impact using the Raven dataset (140 examples). We got 16.4% accuracy for segmented RPMs and 11.4% for non-segmented RPMs. The results demonstrate a significant reduction in *confounding* errors, confirming the issues discussed earlier.

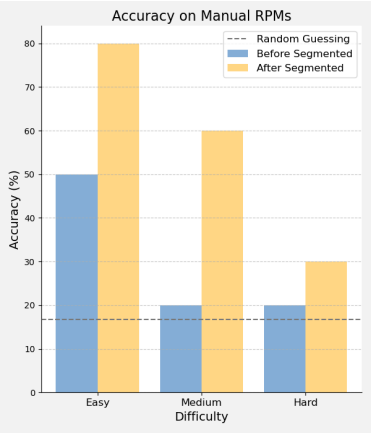


Figure 3: Accuracy of the original RPM as input with that of the segmented RPM as input. Results based on 10 repetitions.

Hallucination We have also observed the model generating hallucinations in the descriptions, particularly when it comes to counting. For instance, in Table 4 (M-Medium, top right), the model erroneously states that there are four dots when, in reality, there are only three.

Data distribution aspect VLMs are presumably trained primarily on real world images, which may cause them to be less sensitive to abstract patterns. We did not try to finetune the model to test this hypothesis because of the limited availability of diverse RPM data. We believe that additional finetuning could potentially improve the performance. However, we hypothesize that finetuning the model with RPMs might not entirely eliminate the *compounding* and *confounding errors*, as they appear to be inherent limitations of the VLMs from training.

5.2 How good is the VLM's deductive reasoning on this task?

Next, we assess the model's ability to perform effective reasoning by conditioning it on the ground truth text description of the RPMs. We provide the prompts in Appendix A.1.

Does the oracle text description improve the model's performance? The original evaluation (Tables 1 and 2) requires the model to directly generate the answer, making it difficult to disentangle

			M-Easy					
Model	**Acc.**	**Ent.**	**A**	**B**	**C**	**D**	**E**	**F**
GPT-4V 1-shot	50%	1.69	0	0	1	1	3	5
GPT-4V 1-shot + Gen. Desc. (CoT)	50%	1.36	0	1	0	0	4	5
GPT-4V 1-shot + Oracle Desc.	60%	1.57	0	1	0	1	2	6
GPT-4V 1-shot + Oracle Desc. - Visual	60%	**0.97**	0	4	0	0	0	6
GPT-4V 1-shot + Oracle Desc. + Rationale	**60%**	1.57	1	0	0	1	2	6
			M-Medium					
Model	**Acc.**	**Ent.**	**A**	**B**	**C**	**D**	**E**	**F**
GPT-4V 1-shot	20%	2.25	0	3	2	2	1	2
GPT-4V 1-shot + Gen. Desc. (CoT)	50%	1.69	0	3	1	1	0	5
GPT-4V 1-shot + Oracle Desc.	**80%**	0.92	1	1	0	0	0	8
GPT-4V 1-shot + Oracle Desc. - Visual	60%	1.57	1	0	0	1	2	6
GPT-4V 1-shot + Oracle Desc. + Rationale	70%	**0.88**	3	0	0	0	0	7
			M-Hard					
Model	**Acc.**	**Ent.**	**A**	**B**	**C**	**D**	**E**	**F**
GPT-4V 1-shot	20%	1.49	0	0	3	0	5	2
GPT-4V 1-shot + Gen. Desc. (CoT)	30%	1.97	0	2	2	0	3	3
GPT-4V 1-shot + Oracle Desc.	40%	1.85	0	3	0	1	2	4
GPT-4V 1-shot + Oracle Desc. - Visual	10%	1.96	2	0	1	5	1	1
GPT-4V 1-shot + Oracle Desc. + Rationale	**50%**	**1.49**	0	2	0	0	3	5

Table 5: Accuracy and breakdown of GPT-4V variants with augmented text description across different RPMs. Each combination is ran for 10 repetitions. The correct answer "F" is marked in color.

the understanding and deductive reasoning aspects. To examine the VLMs more closely, we provided each evaluated model with oracle text descriptions that were manually created by the authors. We then evaluated the models' performance on the three RPM problems and present the results in Table 5 (GPT-4V + Oracle Desc.). The oracle text descriptions can be found in the Appendix A.3. We also provide sampled rationale generated by GPT-4V in the Appendix A.4.

It is evident that the model's performance has been significantly improved with the addition of oracle descriptions for each pattern (Table 5). The models are able to analyze the given patterns and deduce rules for the *M-Easy* and *M-Medium* RPMs, and provide rationale for the problem. For the *M-Hard* RPM, the models demonstrate some capability of reasoning, albeit with some challenges and is far from human parity. We provide additional examples in the Appendix. However, it is not clear whether the models still rely heavily on visual cues or if their reasoning is purely text-based.

Will removing the visual cues harm the model? Next, we examine whether textual information alone is sufficient by removing the visual information. The results, shown in Table 5 (GPT-4V + Oracle Desc. - Visual), are intriguing. Without visual information, the models can maintain a similar level of performance for *M-Easy* and *M-Medium* RPMs. Notably the result solely rely on the textual information of the input is superior to the GPT-4V baseline, which mostly rely on visual information of the input. However, as the tasks become more challenging (*M-Hard* RPM), the models start to struggle. The performance is also worse than GPT-4V baseline. This suggests that for tasks that involve complex spatial layouts and relational reasoning, text alone may be insufficient and potentially confusing, while visual cues may provide additional visual *alignment* and better *comparative* attention. In such cases, visual information and textual clues would complement each other and work in synergy to achieve the optimal performance. Interestingly, when we provide GPT-4V with an incorrect description, there is around an 80% chance that the model recognizes the mismatch between the text and the image and responds with statements such as: "I believe there has been a misinterpretation of the provided image". The model, nevertheless, still generates some rationale which seems adhere more closely to the text description than to the visual cues.

Can the performance be improved by reasoning with noisy text descriptions generated by the model itself? Drawing inspiration from Chain-of-Thoughts (CoT) in the text domain (Wei et al., 2022) and the recent Self-Imagine work (Akter et al., 2024), we further investigate whether VLMs can enhance their performance using noisy text descriptions that they generate on their own. This also helps us understand the extent to which VLM reasoning relies on accurate descriptions of images and the extent to which it can recover from errors in the descriptions. Table 5 (GPT-4V + Gen Desc.) shows that incorrect text descriptions can still produce a gain. The gap between self-generated descriptions and oracle descriptions, however, varies across the different cases.

5.3 HOW GOOD IS THE VLM'S HYPOTHESIS VERIFICATION ON THIS TASK?

Finally, We tested the performance of GPT-4V when it received both an oracle description and an oracle rationale. The oracle rationale, which can be found in Appendix A.1, only includes the explanation of the underlying rule without predicting the final pattern or answer. The results for 10 repetitions on manual examples are shown in Table 5 (GPT-4V + Oracle Desc. + Rationale).

Surprisingly, compared to the row representing GPT-4V + Oracle Desc., the oracle rationale did not significantly improve accuracy. In cases where the model failed, it sometimes directly generated an incorrect answer and at other times extended the rationale but still generated false answers. For example, for M-easy, GPT-4V continued to generate "the third row should have a star, as the first two boxes of the third row (square and circle) already have a star." This indicates that hypothesis generation and verification are closely tied to deductive reasoning, and the model has not yet reached human-level performance in following hints and turning learned rules into future predictions.

Interestingly, strong models like GPT-4V exhibit some strategies similar to humans. For instance, they often use the answer options along with the grid to form and tests hypotheses, rather than generating a hypothesis solely based on the grid and then checking for any matches with the alternative shapes. [4] GPT-4V also sometimes employs a strategy of elimination to rule out incorrect answers (e.g., "the right shape should have a cross sign, which leaves the options to C and F.").

5.4 HOW DOES THE PROMPT FORMAT INFLUENCE THE MODEL PREDICTION?

The format of the prompt can sometimes significantly impact the performance of VLM. For example, we found the arrangement of task instruction and images is crucial to Gemini Pro. We show the results in Table 6. We observed a remarkable 200% increase in prediction accuracy when we simply altered the sequence of these elements. However, we don't observe similar conclusion from other tested models.

Prompting Structure	Mensa
Gemini Pro *Image First*	2.3 ± 1.3
Gemini Pro *Instruction First*	5.4 ± 1.2
GPT4V 1-Shot *w/o Sentinel Token*	6.1 ± 1.5
GPT4V 1-Shot *w/ Sentinel Token*	7.8 ± 1.7

Table 6: Average number of correct predictions made by GPT4-V and Gemini Pro on the Mensa test, demonstrating its sensitivity to the structure of prompts used.

We also delves into the differences in how the model performs under 0-shot and 1-shot evaluation setups. We discovered that using special sentinel tokens, such as `[{BEGIN/END}_OF_EXAMPLE]` , to clearly separate text prompts from images helps the model delineate task instructions from in-context examples. This method of structuring prompts is particularly effective in aiding the model's comprehension across all tested VLMs. For instance, we show the results of GPT-4V in Table 6.

This study underscores that VLMs, unlike their text-only counterparts, can benefit significantly from a more *structured* format in their task prompts. Furthermore, the interaction between different modalities, such as text and image, needs to be carefully considered and evaluated.

6 CONCLUSION

This work is a systematic evaluation of the performance of popular Vision-Language Models (VLMs) in a variety of Raven's Progressive Matrices (RPMs). These tasks serve as a challenging benchmark for assessing the models' ability to reason based on visual clues. We observed that the current state-of-the-art VLMs still fall short of achieving human-level performance on these tasks, with the best-performing models being close-sourced. Our analysis of the models' performance reveals that perceptual understanding may be the main bottleneck, as the models perform better when provided with appropriate textual descriptions. In future work, it would be intriguing to validate our hypothesis concerning the blind spot of VLMs when it comes to describing patterns. This investigation has the potential to enhance the general recognition and attentiveness capabilities of VLMs. Additionally, exploring the development of contrastive learning or reinforcement learning algorithms could further improve the model's visual deductive reasoning abilities.

[4] This generate-then-verify strategy accounts for less than 10% of GPT-4V's behavior in our observation. In such cases the model often rejects the options provided and responds as follows: "Unfortunately, the given options do not correspond with the identified pattern."

REFERENCES

Marwa Abdulhai, Isadora White, Charlie Snell, Charles Sun, Joey Hong, Yuexiang Zhai, Kelvin Xu, and Sergey Levine. Lmrl gym: Benchmarks for multi-turn reinforcement learning with language models. *arXiv preprint arXiv:2311.18232*, 2023.

Sam Acquaviva, Yewen Pu, Marta Kryven, Theodoros Sechopoulos, Catherine Wong, Gabrielle Ecanow, Maxwell Nye, Michael Tessler, and Josh Tenenbaum. Communicating natural programs to humans and machines. *Advances in Neural Information Processing Systems*, 35:3731–3743, 2022.

Kian Ahrabian, Zhivar Sourati, Kexuan Sun, Jiarui Zhang, Yifan Jiang, Fred Morstatter, and Jay Pujara. The curious case of nonverbal abstract reasoning with multi-modal large language models. *arXiv preprint arXiv:2401.12117*, 2024.

Syeda Nahida Akter, Aman Madaan, Sangwu Lee, Yiming Yang, and Eric Nyberg. Self-imagine: Effective unimodal reasoning with multimodal models using self-imagination. *arXiv preprint arXiv:2401.08025*, 2024.

Jacob Andreas, Marcus Rohrbach, Trevor Darrell, and Dan Klein. Neural module networks. In *CVPR*, June 2016.

Stanislaw Antol, Aishwarya Agrawal, Jiasen Lu, Margaret Mitchell, Dhruv Batra, C Lawrence Zitnick, and Devi Parikh. Vqa: Visual question answering. In *Proceedings of the IEEE international conference on computer vision*, pp. 2425–2433, 2015.

Jinze Bai, Shuai Bai, Shusheng Yang, Shijie Wang, Sinan Tan, Peng Wang, Junyang Lin, Chang Zhou, and Jingren Zhou. Qwen-vl: A frontier large vision-language model with versatile abilities. *arXiv preprint arXiv:2308.12966*, 2023.

Tom Brown, Benjamin Mann, Nick Ryder, Melanie Subbiah, Jared D Kaplan, Prafulla Dhariwal, Arvind Neelakantan, Pranav Shyam, Girish Sastry, Amanda Askell, et al. Language models are few-shot learners. *Advances in neural information processing systems*, 33:1877–1901, 2020.

Boyuan Chen, Zhuo Xu, Sean Kirmani, Brian Ichter, Danny Driess, Pete Florence, Dorsa Sadigh, Leonidas Guibas, and Fei Xia. Spatialvlm: Endowing vision-language models with spatial reasoning capabilities. *arXiv preprint arXiv:2401.12168*, 2024. URL `https://arxiv.org/abs/2401.12168`.

Mark Chen, Jerry Tworek, Heewoo Jun, Qiming Yuan, Henrique Ponde de Oliveira Pinto, Jared Kaplan, Harri Edwards, Yuri Burda, Nicholas Joseph, Greg Brockman, et al. Evaluating large language models trained on code. *arXiv preprint arXiv:2107.03374*, 2021.

Peter Clark, Isaac Cowhey, Oren Etzioni, Tushar Khot, Ashish Sabharwal, Carissa Schoenick, and Oyvind Tafjord. Think you have solved question answering? try arc, the ai2 reasoning challenge. *arXiv preprint arXiv:1803.05457*, 2018.

Karl Cobbe, Vineet Kosaraju, Mohammad Bavarian, Mark Chen, Heewoo Jun, Lukasz Kaiser, Matthias Plappert, Jerry Tworek, Jacob Hilton, Reiichiro Nakano, et al. Training verifiers to solve math word problems. *arXiv preprint arXiv:2110.14168*, 2021.

Harm De Vries, Florian Strub, Sarath Chandar, Olivier Pietquin, Hugo Larochelle, and Aaron Courville. Guesswhat?! visual object discovery through multi-modal dialogue. In *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition*, pp. 5503–5512, 2017.

Mor Geva, Daniel Khashabi, Elad Segal, Tushar Khot, Dan Roth, and Jonathan Berant. Did aristotle use a laptop? a question answering benchmark with implicit reasoning strategies. *Transactions of the Association for Computational Linguistics*, 9:346–361, 2021.

Dan Hendrycks, Collin Burns, Steven Basart, Andy Zou, Mantas Mazeika, Dawn Song, and Jacob Steinhardt. Measuring massive multitask language understanding. *arXiv preprint arXiv:2009.03300*, 2020.

Jie Huang and Kevin Chen-Chuan Chang. Towards reasoning in large language models: A survey. *arXiv preprint arXiv:2212.10403*, 2022.

Justin Johnson, Bharath Hariharan, Laurens Van Der Maaten, Li Fei-Fei, C Lawrence Zitnick, and Ross Girshick. Clevr: A diagnostic dataset for compositional language and elementary visual reasoning. In *CVPR*, pp. 2901–2910, 2017.

Sahar Kazemzadeh, Vicente Ordonez, Mark Matten, and Tamara Berg. ReferItGame: Referring to objects in photographs of natural scenes. In *Proceedings of the 2014 Conference on Empirical Methods in Natural Language Processing (EMNLP)*, pp. 787–798, Doha, Qatar, October 2014. Association for Computational Linguistics. doi: 10.3115/v1/D14-1086. URL https://aclanthology.org/D14-1086.

Maithilee Kunda, Keith McGreggor, and Ashok K Goel. A computational model for solving problems from the raven's progressive matrices intelligence test using iconic visual representations. *Cognitive Systems Research*, 22:47–66, 2013.

Philippe Laban, Wojciech Kryściński, Divyansh Agarwal, Alexander R Fabbri, Caiming Xiong, Shafiq Joty, and Chien-Sheng Wu. Llms as factual reasoners: Insights from existing benchmarks and beyond. *arXiv preprint arXiv:2305.14540*, 2023.

Percy Liang, Rishi Bommasani, Tony Lee, Dimitris Tsipras, Dilara Soylu, Michihiro Yasunaga, Yian Zhang, Deepak Narayanan, Yuhuai Wu, Ananya Kumar, et al. Holistic evaluation of language models. *arXiv preprint arXiv:2211.09110*, 2022.

Haotian Liu, Chunyuan Li, Qingyang Wu, and Yong Jae Lee. Visual instruction tuning. *arXiv preprint arXiv:2304.08485*, 2023.

Pan Lu, Hritik Bansal, Tony Xia, Jiacheng Liu, Chunyuan Li, Hannaneh Hajishirzi, Hao Cheng, Kai-Wei Chang, Michel Galley, and Jianfeng Gao. Mathvista: Evaluating mathematical reasoning of foundation models in visual contexts. *arXiv preprint arXiv:2310.02255*, 2023.

Pan Lu, Hritik Bansal, Tony Xia, Jiacheng Liu, Chunyuan Li, Hannaneh Hajishirzi, Hao Cheng, Kai-Wei Chang, Michel Galley, and Jianfeng Gao. Mathvista: Evaluating mathematical reasoning of foundation models in visual contexts. In *ICLR*, 2024.

Syeda Nahida Akter, Zichun Yu, Aashiq Muhamed, Tianyue Ou, Alex Bäuerle, Ángel Alexander Cabrera, Krish Dholakia, Chenyan Xiong, and Graham Neubig. An in-depth look at gemini's language abilities. *arXiv e-prints*, pp. arXiv–2312, 2023.

OpenAI. Gpt-4 technical report, 2023.

Fabio Petroni, Tim Rocktäschel, Patrick Lewis, Anton Bakhtin, Yuxiang Wu, Alexander H Miller, and Sebastian Riedel. Language models as knowledge bases? In *EMNLP*, 2019.

Paul-Edouard Sarlin, Daniel DeTone, Tomasz Malisiewicz, and Andrew Rabinovich. Superglue: Learning feature matching with graph neural networks. In *Proceedings of the IEEE/CVF conference on computer vision and pattern recognition*, pp. 4938–4947, 2020.

Oleksii Sidorov, Ronghang Hu, Marcus Rohrbach, and Amanpreet Singh. Textcaps: a dataset for image captioning with reading comprehension. In *ECCV*, pp. 742–758. Springer, 2020.

Aarohi Srivastava, Abhinav Rastogi, Abhishek Rao, Abu Awal Md Shoeb, Abubakar Abid, Adam Fisch, Adam R Brown, Adam Santoro, Aditya Gupta, Adrià Garriga-Alonso, et al. Beyond the imitation game: Quantifying and extrapolating the capabilities of language models. *arXiv preprint arXiv:2206.04615*, 2022.

Gemini Team, Rohan Anil, Sebastian Borgeaud, Yonghui Wu, Jean-Baptiste Alayrac, Jiahui Yu, Radu Soricut, Johan Schalkwyk, Andrew M Dai, Anja Hauth, et al. Gemini: a family of highly capable multimodal models. *arXiv preprint arXiv:2312.11805*, 2023.

Xuezhi Wang, Jason Wei, Dale Schuurmans, Quoc Le, Ed Chi, Sharan Narang, Aakanksha Chowdhery, and Denny Zhou. Self-consistency improves chain of thought reasoning in language models. *arXiv preprint arXiv:2203.11171*, 2022.

Jason Wei, Xuezhi Wang, Dale Schuurmans, Maarten Bosma, Fei Xia, Ed Chi, Quoc V Le, Denny Zhou, et al. Chain-of-thought prompting elicits reasoning in large language models. *Advances in Neural Information Processing Systems*, 35:24824–24837, 2022.

Zhengyuan Yang, Linjie Li, Kevin Lin, Jianfeng Wang, Chung-Ching Lin, Zicheng Liu, and Lijuan Wang. the Dawn of Lmms: Preliminary Explorations With Gpt-4v(ision), 2023.

Xiang Yue, Yuansheng Ni, Kai Zhang, Tianyu Zheng, Ruoqi Liu, Ge Zhang, Samuel Stevens, Dongfu Jiang, Weiming Ren, Yuxuan Sun, et al. Mmmu: A massive multi-discipline multimodal understanding and reasoning benchmark for expert agi. *arXiv preprint arXiv:2311.16502*, 2023.

Chi Zhang, Feng Gao, Baoxiong Jia, Yixin Zhu, and Song-Chun Zhu. Raven: A dataset for relational and analogical visual reasoning. In *CVPR*, pp. 5317–5327, 2019.

Yizhe Zhang, Jiarui Lu, and Navdeep Jaitly. the Entity-deduction Arena: A Playground for Probing the Conversational Reasoning and Planning Capabilities of Llms. *arXiv preprint arXiv:2310.01468*, 2023.

A APPENDIX

A.1 PROMPTS FOR DIFFERENT DATASET AND TASKS

We listed all the prompts we used to generate the response from different models.

Mensa. (0-shot)

You can see a 3x3 grid of 9 boxes, one of which is empty (marked as ?). You have to choose which of the 6 alternative shapes (A-F) should be placed in the empty box in order to complete the pattern that connects the shapes. Finally, provide your prediction as Answer: "X"
{query image}

Mensa (1-shot)

You can see a 3x3 grid of 9 boxes, one of which is empty (marked as ?). You have to choose which of the 6 alternative shapes (A-F) to be placed in the empty box in order to complete the pattern that connects the shapes. Think step by step by first describe the each box in the 3x3 grid, and each of the alternative shapes as the candidate answers. Then identify the common pattern. Finally, provide your prediction as Answer: "X"
For example, for the following image:
{in-context image}
{in-context description}
{in-context rationale}
{in-context answer}
Now do the following one:
{query image}

Mensa (1-shot) + Oracle Desc.

You can see a 3x3 grid of 9 boxes, one of which is empty (marked as ?). You have to choose which of the 6 alternative shapes (A-F) to be placed in the empty box in order to complete the pattern that connects the shapes. Think step by step by first describe the each box in the 3x3 grid, and each of the alternative shapes as the candidate answers. Then identify the common pattern. Finally, provide your prediction as Answer: "X"
For example, for the following image:
{in-context image}
{in-context description}
{in-context rationale}
{in-context answer}
Now do the following one:
{query image}
{query oracle description}

Mensa (1-shot) + Oracle Desc. + Oracle Rationale

You can see a 3x3 grid of 9 boxes, one of which is empty (marked as ?). You have to choose which of the 6 alternative shapes (A-F) to be placed in the empty box in order to complete the pattern that connects the shapes. Think step by step by first describe the each box in the 3x3 grid, and each of the alternative shapes as the candidate answers. Then identify the common pattern. Finally, provide your prediction as Answer: "X"
For example, for the following image:
{*in-context image*}
{*in-context description*}
{*in-context rationale*}
{*in-context answer*}
Now do the following one:
{*query image*}
{*query oracle description*}
{*query oracle rationale*}

IntelligenceTest (0-shot)

This image is an Intelligence Test Question asking which figure (A-F) belongs in the bottom right box. Please select the correct answer. You must first give your explanation and then output the answer at the end of your response in the format: "The correct answer is: _".
{*query image*}

RAVEN (0-shot)

You can see a 3x3 grid of 9 boxes, one of which is empty (marked as ?). You have to choose which of the 8 alternative shapes (A-H) should be placed in the empty box in order to complete the pattern that connects the shapes. You must first give your explanation and then output the answer at the end of your response in the format: "The correct answer is: _".
{*query image*}

Segmented Prompt (0-shot)

In the first image, you will see a 3x3 grid of 9 boxes. Each row has three images and is to be read from left to right, with the last image in the third row is missing (marked as ?). Your task is to infer the correct pattern that should complete each row based on the sequence observed in the preceding patterns, and finally select the right option (A, B, C, D, E, F) that fits the 3rd row's last image.
{query image}
For your convenience, I provide 15 segmented figures: the question grid is segmented into 9 parts, and the answer options are segmented into 6 parts. q0, q1, and q2 are the first row, q3, q4, and q5 are the second row, and q6, q7, and q8 are the third row. A, B, C, D, E, and F are the answer options. Your task is to find which option should be placed in q8.
q0: {q0 image}
q1: {q1 image}
q2: {q2 image}
q3: {q3 image}
q4: {q4 image}
q5: {q5 image}
q6: {q6 image}
q7: {q7 image}
q8: {q8 image}
A: {A image}
B: {B image}
C: {C image}
D: {D image}
E: {E image}
F: {F image}
For each row, analyze the changes and relationships between the images. Consider the number of shapes, the types of shapes, their positions, the shading, and any other changes that occur from one pattern to the next. Once you have identified the rule or sequence that applies to the rows, select the option (A, B, C, D, E, F) that contains the pattern which correctly completes the third row sequence.
Please first give your explanation and then write the answer at the end of your response in the format: "The correct answer is: _".

A.2 SAMPLING METHODS FOR MODEL EVALUATIONS

In the main text, we present evaluation results obtained by independently running each model 10 times and estimating their respective statistics. For GPT4-V, we found that using a zero temperature setting along with different random seeds effectively balances prediction accuracy with reasoning diversity. In the case of Gemini Pro Vision, the optimal performance is achieved at a temperature setting of 0.4. However, for the self-consistency evaluation, we increase the sampling temperature to a maximum of 0.7. This adjustment facilitates more varied predictions which are beneficial for majority-voting processes. For all other models, we observed that a lower temperature range, between 0 and 0.2, is necessary. This lower temperature helps the models better adhere to instructions and improves their task prediction accuracy, though it also results in somewhat less variability in the models' predictions.

M-Easy (Oracle)

Description

The grid:

1. Top left box: square with a cross sign.
2. Top center box: circle with a star.
3. Top right box: Empty triangle.
4. Middle left box: empty square.
5. Middle center box: circle with a cross sign.
6. Middle right box: triangle with a star.
7. Bottom left box: square with a star.
8. Bottom center box: empty circle.
9. Bottom right box: '?'.

The alternative shapes are:

- A. Triangle with a star.
- B. Triangle with a plus sign.
- C. Circle with a cross sign.
- D. Circle with a star.
- E. Empty triangle.
- F. Triangle with a cross sign.

Rationale

Analyzing the grid, there seems to be a pattern involving both the shapes and the symbols inside them:

- The symbols inside the shapes alternate between a cross, a star, and then a blank space as you move horizontally across each row.
- The shapes themselves also alternate within each row – this is seen with the square, circle, and triangle repeating in each row in that order.

A.3 ORACLE DESCRIPTION AND RATIONALE FOR EACH RPMS

M-Medium (Oracle)

Description

The grid:

1. Top left box: A downward-pointing triangle with three dots in it.
2. Top center box: A leftward-pointing triangle with two dots in it.
3. Top right box: An upward-pointing triangle with one dot in it.
4. Middle left box: A rightward-pointing triangle with two dots in it.
5. Middle center box: A downward-pointing triangle with one dot in it.
6. Middle right box: A leftward-pointing triangle with three dots in it.
7. Bottom left box: An upward-pointing triangle with one dot in it.
8. Bottom center box: A rightward-pointing triangle with three dots in it.
9. Bottom right box: '?'.

The alternative shapes are:

- A. An upward-pointing triangle with two dots in it.
- B. A downward-pointing triangle with one dot in it.
- C. A leftward-pointing triangle with one dot in it.
- D. A rightward-pointing triangle with two dots in it.
- E. A leftward-pointing triangle with two dots in it.
- F. A downward-pointing triangle with two dots in it.

Rationale

Analyzing the grid, it appears that there's a pattern related to the direction the triangle is pointing and the number of dots within the triangles.
First, let's establish the patterns of triangle directions and dots count:

- The first row has the triangles pointing downward, to the left, and then up.
- The second row has the triangles pointing rightward, downward, and then to the left.
- This implies that the direction that the triangle is pointing to is rotating clockwise in each row.

Now let's look at the pattern in the number of dots:

- The first row has 3, 2, 1 dots.
- The second row has 2, 1, 3 dots.
- This implies a pattern of a decreasing sequence from left to right.

Hard (Oracle)

Description

The grid:

1. Top left box: white circle, white triangle, black square.
2. Top center box: white triangle, black circle, white square.
3. Top right box: black square, black square, white triangle.
4. Middle left box: black circle, white square, white triangle.
5. Middle center box: black square, white triangle, black square.
6. Middle right box: white triangle, black square, white circle.
7. Bottom left box: white triangle, black square, black square.
8. Bottom center box: black square, white circle, white triangle.
9. Bottom right box: '?'.

The alternative shapes are:

- A. white circle, white triangle, black square.
- B. black circle, white square, white triangle.
- C. white circle, white square, black triangle.
- D. white circle, black square, white triangle.
- E. black square, white triangle, white circle.
- F. white square, white triangle, black circle.

Rationale

Analyzing the grid, it appears that there's a pattern along the diagonal lines from top right to bottom left direction.

- The three elements in the first diagonal line (top right, middle center, bottom left) consist of a black square, a black square, and a white triangle. As we follow the diagonal line, these elements rotate 120 degrees clockwise.
- The three elements in the second diagonal line (top left, middle right, bottom center) consist of a white circle, a white triangle, and a black square. As we follow the diagonal line, these elements rotate 120 degrees clockwise.
- This implies that the third diagonal line (top center, middle left, bottom right) should also follow this pattern.

A.4 RATIONALE GENERATED BY GPT-4V

In this paper, we present several examples that demonstrate the reasoning capabilities of GPT-4V for solving certain RPM tasks. From the examples, it becomes evident that as the difficulty level of the problem increases, the GPT-4V tends to struggle with its reasoning abilities and provides an explanation that is nonsensical.

Hard (rationale)

Analyzing the grid, it appears that there's a pattern along the diagonal lines from top right to bottom left direction. - The three elements in the first diagonal line (top right, middle center, bottom left) consist of a black square, a black square, and a white triangle. As we follow the diagonal line, these elements rotate 120 degrees clockwise. - The three elements in the second diagonal line (top left, middle right, bottom center) consist of a white circle, a white triangle and a black square. As we follow the diagonal line, these elements rotate 120 degrees clockwise. - This implies that the third diagonal line (top center, middle left, bottom right) should also follow this pattern.

Generated rationale for *M-Medium* RPM

Analyzing the grid, it appears there's a pattern related to the direction the triangle is pointing and the number of dots within the triangles.
First, let's establish the patterns of triangle directions and dots count:

- The first row has the triangles pointing downward, to the left, and then up.
- The second row has the triangles pointing rightward, downward, and then to the left.
- It implies that for the third row, following the same pattern, the triangles should point upward, rightward, and then downward.

Now let's look at the pattern in the number of dots:

- Along the vertical columns, we can observe a pattern in dot counts:
  - The first column has 3, 2, 1 dots.
  - The second column has 2, 1, 3 dots.
  - If we follow this pattern, the third column should have 1, 3, 2 dots from top to bottom.

Given these observed sequences, the third row should have a triangle pointing downward with two dots in it, fitting the patterns of both direction and dot count. Looking at the alternative shapes, option F is the downward-pointing triangle with two dots within it.
Answer: "F"

Generated rationale for *M-Hard* RPM

Analyzing the grid, we can deduce the following pattern:

1. The shapes in each box follow a sequence that changes from one cell to the next horizontally.
2. There are three types of shapes: circle, square, and triangle, and each type also has two variants of color: black or white.

To understand how the sequence works, we can lay out the patterns we see:

- In the top row, we start with a white circle, white triangle, and black square. Moving right, the shapes rotate positions clockwise, and their colors invert (the white circle becomes a black circle and so on).
- The same pattern follows in the middle row: black circle, white square, white triangle becomes black square, white triangle, black square, and again inverting colors and rotating the shapes clockwise.
- Following this pattern in the bottom row, white triangle, black square, black square should change by rotating the shapes clockwise and inverting the colors. The black square would become a white square, the black square would become a white circle, and the white triangle would stay the same, remaining in the bottom position.

Matching this description with the alternatives given, option ”F” (white square, white triangle, black circle) fits the observed sequence.
Answer: ”F”