

# Your Reasoning Benchmark May Not Test Reasoning: Revealing Perception Bottleneck in Abstract Reasoning Benchmarks

**Xinhe Wang**  
Carnegie Mellon University  
xinhew@andrew.cmu.edu

**Jin Huang**  
University of Michigan  
huangjin@umich.edu

**Xingjian Zhang**  
University of Michigan  
jimmyzsj@umich.edu

**Tianhao Wang**  
University of California San Diego  
tianhaowang@ucsd.edu

**Jiaqi W. Ma**  
University of Illinois Urbana-Champaign  
jiaqima@illinois.edu

## Abstract

Reasoning benchmarks such as the Abstraction and Reasoning Corpus (ARC) and ARC-AGI are widely used to assess progress in artificial intelligence and are often interpreted as probes of core, so-called “fluid” reasoning abilities. Despite their apparent simplicity for humans, these tasks remain challenging for frontier vision-language models (VLMs), a gap commonly attributed to deficiencies in machine reasoning. We challenge this interpretation and hypothesize that the gap arises primarily from limitations in visual perception rather than from shortcomings in inductive reasoning.

To verify this hypothesis, we introduce a two-stage experimental pipeline that explicitly separates perception and reasoning. In the perception stage, each image is independently converted into a natural-language description, while in the reasoning stage a model induces and applies rules using these descriptions. This design prevents leakage of cross-image inductive signals and isolates reasoning from perception bottlenecks. Across three ARC-style datasets, Mini-ARC, ACRE, and Bongard-LOGO, we show that the perception capability is the dominant factor underlying the observed performance gap by comparing the two-stage pipeline with against standard end-to-end one-stage evaluation. Manual inspection of reasoning traces in the VLM outputs further reveals that approximately 80 percent of model failures stem from perception errors. Together, these results demonstrate that ARC-style benchmarks conflate perceptual and reasoning challenges and that observed performance gaps may overstate deficiencies in machine reasoning. Our findings underscore the need for evaluation protocols that disentangle perception from reasoning when assessing progress in machine intelligence.

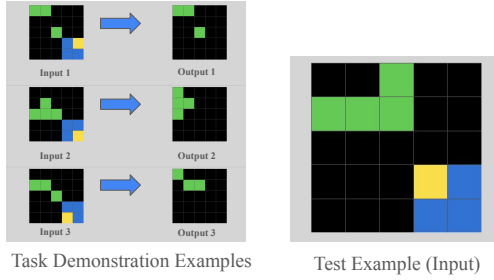
## 1 Introduction

Reasoning capability has increasingly become a central criterion for evaluating the progress of fron-

tier artificial intelligence (AI) models (Google, 2025; OpenAI, 2024). Correspondingly, a wide range of reasoning benchmarks have emerged to assess different dimensions of reasoning performance (ARC Prize Foundation, 2025; Yue et al., 2025; Chen et al., 2021; Rein et al., 2024). Among these, the Abstraction and Reasoning Corpus (ARC) (Chollet, 2019) and its successor, ARC-AGI (ARC Prize Foundation, 2025; Chollet et al., 2025), have attracted particular attention, as they are widely viewed as tests of core reasoning abilities and, in some views, as indicators of progress toward general intelligence (Chollet et al., 2024). Reflecting their influence, frontier AI labs, including OpenAI and Google, routinely highlight their models’ performance on ARC as a key indicator of advances in general reasoning capabilities (Google, 2025; ARC Prize Foundation, 2025; OpenAI, 2024).

The core design principle of ARC and ARC-AGI is to evaluate “fluid intelligence (the ability to reason, solve novel problems, and adapt to new situations) rather than *crystallized* intelligence, which relies on accumulated knowledge and skills,” (ARC Prize Foundation, 2025) where the former is considered a core reasoning capability at which current AI models still fall short. Concretely, ARC and ARC-AGI take the form of grid-based puzzles, as shown in Figure 1a. Each problem consists of several input-output demonstration pairs and a test input. The solver—human or AI—must induce a common rule from the demonstrations and then apply it to the test input. In the specific example in Figure 1a, the correct solution requires recognizing the pattern that the  $2 \times 2$  block on the bottom-right corner encodes the rotation rule for the  $3 \times 3$  green object in the upper-left, and the position of the yellow pixel indicates degrees of rotation. The solver must then apply this inferred rule to the test input to generate the correct test output.

Empirically, these problems are found to be easy



(a) An example ARC problem shown in 2-D visually.

Task Demonstration Examples:

Input:  $[[0, 0, 0, 0, 0], [3, 3, 0, 0, 0], [0, 0, 3, 0, 0], [0, 0, 0, 1, 1], [0, 0, 0, 4, 1]]$   
 Output:  $[[3, 0, 0, 0, 0], [0, 3, 3, 0, 0], [0, 0, 0, 0, 0], [0, 0, 0, 0, 0], [0, 0, 0, 0, 0]]$

(Omitting two demonstration examples.)

Test Example:

Input:  $[[0, 0, 3, 0, 0], [3, 3, 3, 0, 0], [0, 0, 0, 0, 0], [0, 0, 0, 4, 1], [0, 0, 0, 1, 1]]$

(b) The same ARC problem shown in a serialized format.

Figure 1: An example ARC problem. One is asked to induce a common rule from the input-output pairs in the task demonstration examples, and then apply this rule to the test input to generate the test output. The difficulty of this task to human players critically depends on how the problem is presented.

for humans yet surprisingly hard for even state-of-the-art frontier AI models<sup>1</sup> (ARC Prize Foundation, 2025). A common belief is that this persistent gap between AI and human reflects a fundamental advantage of humans’ reasoning capabilities (Chollet et al., 2024). In this work, we critically examine this belief, and *hypothesize* that the gap arises primarily because ARC problems particularly favor humans’ innate visual **perception**, rather than reflecting a genuine difference in the **reasoning** capability (or “fluid intelligence”). This hypothesis is motivated by the observation that the difficulty of the ARC problem to human players depends strongly on how the problem is presented. As shown in Figure 1, presenting the same problem in a serialized format (Figure 1b) makes it far more difficult for humans to solve than when shown in its original 2-D visual format (Figure 1a).

Our hypothesis involves the explicit separation of **perception** and **reasoning** capabilities required to solve ARC-style tasks. Conceptually, perception refers to the ability to *recognize meaningful objects from raw visual inputs*, whereas reasoning refers to the ability to *induce patterns among the recognized objects in demonstrations*. A key challenge in verifying our hypothesis lies in the fact that the success of perception is a prerequisite for the success of reasoning in solving an ARC-style problem, which makes it difficult to have a clean measure and comparison of the perception and reasoning capabilities of a given model.

To address this challenge and verify our hypothesis, we design a two-stage experimental pipeline.

<sup>1</sup>Throughout this paper, we focus on vision-language models (VLMs) unless stated otherwise, as most frontier models are evaluated on these benchmarks in VLM form. We note that other kinds of models specialized for ARC-style tasks fall outside the scope of our discussion.

The first stage (*perception stage*) transforms the raw image inputs into natural language descriptions. Crucially, this transformation is *applied to each images in isolation*, ensuring that no cross-image inductive signals are leaked during this stage. This atomistic approach guarantees that inductive reasoning occurs exclusively in the second stage (*reasoning stage*), where an AI model is tasked with solving the problem using the natural language descriptions obtained from the perception stage. The natural-language-based representation is supposed to alleviate the perception challenge for the model while preserving the inductive structure of the problem. This pipeline allows us to isolate the model’s inductive reasoning performance from its perceptual bottlenecks, providing a clearer picture of where the “reasoning gap” truly resides.

We conduct experiments on three ARC-style visual reasoning datasets, Mini-ARC (Kim et al., 2022), ACRE (Zhang et al., 2021), and Bongard-LOGO (Nie et al., 2020). We compare the performance of vision-language models (VLMs) under both the standard end-to-end one-stage setting and the proposed two-stage pipeline. Our empirical results provide three key findings that support our hypothesis. First, we demonstrate that the two-stage pipeline, in which a dedicated perception stage transforms the raw image inputs into natural language descriptions, significantly outperforms the end-to-end one-stage application of the same VLM. Second, we find that a hybrid two-stage pipeline combining a strong VLM for the perception stage with a weaker VLM for the reasoning stage yields performance close to that of an end-to-end strong VLM, while substantially outperforming an end-to-end weak VLM. This result suggests that perceptual capability, rather than reasoning strength, is the

primary bottleneck in these tasks. Finally, through manual inspection of model reasoning traces, we observe that approximately 80% of failure cases stem from perception errors, i.e., failing to properly identify visual objects. Furthermore, the majority of the performance gains achieved by the two-stage pipeline can be attributed to a reduction in such perception errors.

In summary, our study reveals an important perception bottleneck in an influential class of reasoning benchmarks, abstract reasoning benchmarks, through carefully controlled experiments. Our results suggest that performance gaps on ARC-style tasks may conflate limitations in visual perception with deficiencies in inductive reasoning. This finding calls for caution in interpreting these benchmarks as direct measures of reasoning or fluid intelligence in frontier AI models, and highlights the importance of disentangling perceptual and reasoning components when evaluating progress in machine reasoning.

## 2 Related Work

**Reasoning Benchmarks.** Reasoning benchmarks can be categorized by (i) the input modality and (ii) how much they rely on external knowledge versus in-context rule induction. Text-only benchmarks probe reasoning entirely in language, spanning broad knowledge and multi-disciplinary question answering (e.g., MMLU (Hendrycks et al., 2020)), multi-step mathematical problem solving (e.g., GSM8K (Cobbe et al., 2021)), functional code generation (e.g., HUMAN-EVAL (Chen et al., 2021)), and harder challenge suites designed to reduce superficial heuristics (e.g., GPQA (Rein et al., 2024), BIG-BENCH HARD (Suzgun et al., 2023)). Vision-language benchmarks explicitly couple perception (e.g., reading, grounding, extracting structure from images) with knowledge-based reasoning, including scientific and mathematical reasoning over diagrams (e.g., SCIENCEQA (Lu et al., 2022), MATHVISTA (Lu et al., 2023)), text-rich visual understanding (e.g., DOCVQA (Mathew et al., 2021), TEXTVQA (Singh et al., 2019)), and comprehensive multi-domain suites (e.g., MMMU (Yue et al., 2024), MMBENCH (Liu et al., 2024b)). Finally, knowledge-light visual abstraction benchmarks aim to minimize human priors by emphasizing pattern recognition, abstraction, and generalization from minimal visual examples (e.g., ARC (ARC

Prize Foundation, 2025), ACRE (Zhang et al., 2021) and BONGARD-LOGO (Nie et al., 2020)). These knowledge-light abstraction benchmarks are the primary focus of our work: although they are widely viewed as reasoning-centric tests that require minimal perceptual effort, we demonstrate that their performance can be strongly limited by perceptual bottlenecks, and that substantial gains can arise from improving perception rather than advancing reasoning.

**State-of-the-Art Performance on the ARC Benchmarks.** Recent progress on ARC/ARC-AGI can be broadly grouped into (i) general-purpose foundation models, including LLMs that serialize grids into text VLMs that operate directly on the visual grid representation; and (ii) tailored ARC solvers that introduce task-specific search, program synthesis, or specialized architectures. For general-purpose foundation models, GPT-5.2 Pro (High) and Gemini 3 Pro (Deep Think) have established a new ceiling, achieving scores of 54.2% and 45.1% respectively in ARC-AGI-2 primarily due to their extended thinking ability (ARC Prize, Inc., 2025). This shift towards scaling test-time computation has crystallized into a new paradigm named “refinement loops,” where systems iteratively optimize solutions against feedback rather than relying on single-shot inference.

Among tailored approaches, test-time training (TTT) has emerged as a strong general mechanism for few-shot adaptation on ARC, substantially improving over fine-tuned baselines and often combining well with other solver components (Li et al., 2024; Akyürek et al., 2025). Alternative neural approaches include the Hierarchical Reasoning Model (HRM) and its recursive variants, which utilize small recurrent architectures (Wang et al., 2025; Jolicoeur-Martineau, 2025), and masked diffusion models, which refine the grid globally to capture structural constraints (Franzen et al., 2025). Additionally, CompressARC explores test-time learning by minimizing description length on the target puzzle (Liao and Gu, 2025). Finally, a very recent concurrent work argues that “ARC is a vision problem” (Hu et al., 2025), reframing ARC as image-to-image translation with a ViT-style backbone and test-time adaptation, which gives an example of a vision-centric route to ARC performance. While this recent trend suggests that enhancing perception boosts performance, our primary goal is to challenge the interpretation of ARC

performance as a proxy for general intelligence. Therefore, in this paper, we focus specifically on general-purpose VLMs, aiming to understand how much of their ARC performance is limited by perception rather than reasoning.

**Limitations of the ARC Benchmarks.** The research community has suggested several limitations that complicate treating ARC scores as a clean measure of general reasoning. First, strong performance can result from increased compute, either through larger training budgets or extensive test-time training or searching (Mitchell, 2024). Therefore, gains in accuracy may reflect better fitting to the ARC task distribution rather than improved reasoning. Second, the small hidden evaluation set increases the risk of implicit overfitting to benchmark-specific patterns (Chollet et al., 2024). Third, ARC tasks are inherently *visual*, yet many recent solutions are language-centric (Li et al., 2024; Akyürek et al., 2025). Relatedly, prior work on VLM visual reasoning shows that caption-based vision-text decoupling can improve performance and reveal bottlenecks in combining perception with reasoning (Weng et al., 2025). A concurrent work shows that reframing ARC as an image-to-image translation problem and applying standard vision architectures can achieve near-human performance on ARC-1 (Hu et al., 2025). This result suggests that high ARC scores can be obtained by improving visual representations, without necessarily requiring more general reasoning mechanisms. Our work builds on these studies and examines how perception and reasoning interact when using VLMs to solving ARC tasks.

### 3 Verifying the Hypothesis with a Two-Stage Pipeline

In this section, we formally introduce the design of a two-stage pipeline to verify the hypothesis that the performance of existing VLMs is constrained more by perception than by reasoning.

#### 3.1 Abstract Reasoning Tasks

We start by formalizing the (visual) abstract reasoning tasks considered in this study. Let  $\mathcal{T}$  denote a set of tasks, where each task  $T \in \mathcal{T}$  consists of a small number  $n$  (typically  $n \leq 10$ ) of input-output pairs  $(x_i, y_i) \in \mathcal{X} \times \mathcal{Y}$  serving as demonstration examples, together with a test example for which only the input  $(x_{n+1})$  is observed. Formally,

$$T = \{(x_1, y_1), \dots, (x_n, y_n), x_{n+1}\}.$$

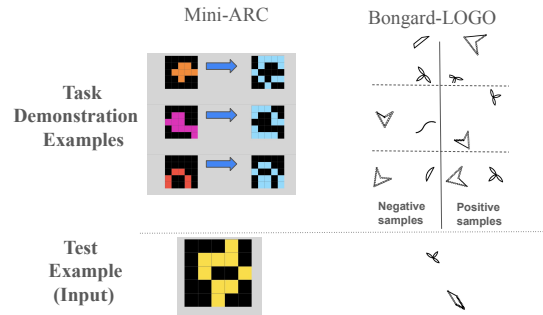


Figure 2: Example tasks of two ARC-style benchmarks, Mini-ARC (Kim et al., 2022) and Bongard-LOGO (Nie et al., 2020). The task in Mini-ARC maps image inputs to image outputs. The task in Bongard-LOGO maps image inputs to binary outputs (positive or negative). In both cases, visual objects that are immediately recognizable by humans serve as the critical basis for the latent mapping rules.

The goal of the task is to predict the corresponding output  $y_{n+1}$  by inductively inferring the underlying relationship between inputs and outputs from the demonstration examples. Figure 2 illustrates example tasks from two ARC-style benchmarks, where each task involves images as inputs and, in some cases, as both inputs and outputs. A shared characteristic of these benchmarks is that the images contain objects that are immediately and unambiguously recognizable to humans, which serve as the critical basis for the latent mapping rules.

When evaluated on these benchmarks, a frontier VLM model is typically used as a mapping  $f : \mathcal{T} \rightarrow \mathcal{Y}$  that directly predicts  $y_{n+1}$  with  $f(T)$ , given the task  $T$  with raw image inputs. However, the model can significantly underperform humans in recognizing the objects visually salient to humans. In this end-to-end evaluation paradigm, it is difficult to quantify the extent to which the model performance is limited by the model’s perception capabilities compared to its reasoning capabilities.

#### 3.2 Quantifying the Perception Bottleneck with a Two-Stage Pipeline

To verify our hypothesis and quantify the perception bottleneck of VLMs, we evaluate the models with a two-stage pipeline that explicitly separates **perception** and **reasoning**. In the first stage (the perception stage), images are transformed into natural language descriptions. In the second stage (the reasoning stage), the model takes these descriptions to enrich the task representation  $T$ , and predicts the output  $y_{n+1}$  on the test example.

**Design Principles of the Perception Stage.** The perception stage is designed according to two key

principles:

1. **No cross-image inductive signal leakage.** The transformation is applied independently to each image in isolation, ensuring that it reduces perceptual difficulty without introducing inductive cues that could alter the intrinsic reasoning difficulty of the task.
2. **Generic human perceptual priors.** The transformation incorporates generic human perceptual priors, such as the identification of objects and recognition of their colors or shapes. This choice directly reflects our hypothesis that the human–AI performance gap is largely driven by the fact that these benchmarks favor human’s innate visual perception.

### Formal Description of the Two-Stage Pipeline.

Formally, for each benchmark, we construct two *uniform* transformations,  $g_{\mathcal{X}} : \mathcal{X} \rightarrow \tilde{\mathcal{X}}$  and  $g_{\mathcal{Y}} : \mathcal{Y} \rightarrow \tilde{\mathcal{Y}}$ , which are respectively applied to the input and output spaces, consistently across all tasks within the benchmark. In practice, these transformations are implemented by prompting a VLM model to convert each image into a corresponding natural language description, with prompts that explicitly instruct the model to attend to features aligned with generic human visual priors. When the output is not an image, the transformation  $g_{\mathcal{Y}}$  is defined as the identity mapping. With these transformations, a task  $T = \{(x_1, y_1), \dots, (x_n, y_n), x_{n+1}\}$  will be enriched as

$$\tilde{T} = \left\{ (x_1, g_{\mathcal{X}}(x_1), y_1, g_{\mathcal{Y}}(y_1)), \dots, \right. \\ \left. (x_n, g_{\mathcal{X}}(x_n), y_n, g_{\mathcal{Y}}(y_n)), \right. \\ \left. (x_{n+1}, g_{\mathcal{X}}(x_{n+1})) \right\} \in \tilde{\mathcal{T}},$$

which will be further fed into a VLM,  $h : \tilde{\mathcal{T}} \rightarrow \mathcal{Y}$ , during the reasoning stage to complete the prediction  $h(\tilde{T})$ .

### 3.3 Two Evaluation Settings

We consider two evaluation settings that differ in how VLMs are instantiated within the two-stage pipeline.

**Setting 1: Same-Model Perception.** In the first setting, we use the same VLM for the transformations  $g_{\bullet}$ , the reasoning stage  $h$ , as well as the baseline end-to-end one-stage predictor  $f$ . In this

case, we expect the two-stage prediction  $h(\tilde{T})$  to outperform the one-stage prediction  $f(T)$ , as the perception stage explicitly incorporates additional human perceptual priors. **Importantly, any performance improvement in this comparison can be attributed solely to the mitigation of the perception bottleneck**, since the two-stage pipeline does not reduce the inductive reasoning difficulty by design<sup>2</sup>.

**Setting 2: Stronger-Model Perception.** In the second setting, we use a stronger VLM for the transformations  $g_{\bullet}$ , and a weaker VLM for the reasoning stage  $h$ . In this case, we compare this hybrid two-stage pipeline against end-to-end one-stage predictions using the strong and weak models, denoted by  $f_S(T)$  and  $f_W(T)$ , respectively. First, we expect  $h(\tilde{T})$  to substantially outperform  $f_W(T)$ , since the perception stage is further enhanced. Second, if the performance of  $h(\tilde{T})$  approaches that of  $f_S(T)$ , this suggests that the performance gap between the weak and strong models is driven primarily by differences in perception capability rather than reasoning capability.

### 3.4 Fine-Grained Error Attribution with Four Categories

**Conceptual Decomposition of the Task-Solving Process.** Given the similar task structure of the abstract reasoning benchmarks, we conceptually decompose the task-solving process into four steps:

1. **Perception (Demonstration).** The system must correctly perceive each input pair  $(x_i, y_i)$ .
2. **Reasoning (Inductive).** From the correctly perceived demonstrations, the system must infer the underlying latent mapping rule that governs the task.
3. **Perception (Testing).** The system must also accurately perceive the test input.
4. **Reasoning (Deductive).** Given the inferred rule, the system applies it to the test input to make the correct prediction.

This decomposition distinguishes steps (1) and (3), which primarily depend on perception capabilities, from steps (2) and (4), which primarily depend on reasoning capabilities. It is worth noting that

<sup>2</sup>In fact, the two-stage pipeline may slightly increase the inductive reasoning difficulty since the model must perform induction with more information.

the four steps are not independent. For example, an error in the step 1 will propagate to and cause failures in steps (2) and (4). An illustration of the dependency of these steps is shown in 3.

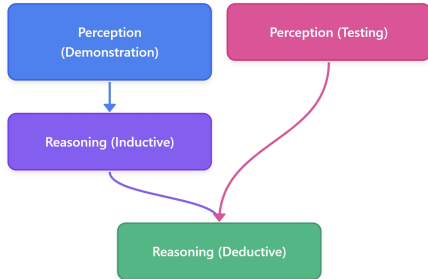


Figure 3: Dependency graph illustrating the four-step task-solving process. Errors in earlier stages propagate to subsequent stages, affecting final predictions.

**Error Attribution with Four Categories.** We observe that the failure cases in our experiments can almost always be attributed to one of four error categories corresponding to the four steps above. In practice, this error attribution can be done by manually inspecting the reasoning traces in the VLM outputs. This procedure is applicable to both one-stage and two-stage predictions. To enable a fair comparison between one-stage and two-stage predictions on the same benchmark, we assign the prediction on each task to one of five categories: the four error categories and a *Correct* category, ensuring that total counts match across the two prediction settings. Given the largely sequential dependency among the four steps, errors are attributed to the earliest step at which a failure occurs. In Section 4.4, we conduct large-scale error attribution across multiple benchmarks for both one-stage and two-stage predictions, and demonstrate that the performance gains of the two-stage pipeline over the one-stage baseline arise almost exclusively from a reduction in perception errors, i.e., errors in steps (1) and (3).

## 4 Experiments

### 4.1 Datasets

We conduct experiments on three ARC-style visual abstract reasoning benchmarks. These datasets are designed to assess compositional and inductive reasoning from visual inputs, making them well suited for studying the interplay between perception and reasoning in VLMs.

**Mini-ARC.** Mini-ARC (Kim et al., 2022) is a reduced-scale variant of ARC with fewer colors, smaller grids, and simplified transformations. It maintains the same format as ARC but with lower perceptual complexity, making it useful for isolating reasoning performance under easier perception.

**ACRE (Abstract Causal REasoning Beyond Covariation).** ACRE (Zhang et al., 2021) is a symbolic visual reasoning benchmark designed for causal induction. Each task is formulated as a classification problem conditioned on demonstrations. Specifically, a task consists of six demonstrations followed by four prediction queries. The demonstrations illustrate the presence or absence of certain objects and the corresponding state of a pink board, which may blink (activated), remains unlit (deactivated), or be underdetermined. Each query presents a new configuration of objects, and the objective is to predict the status of the pink board.

**Bongard-LOGO.** Bongard-LOGO (Nie et al., 2020) is a visual reasoning benchmark inspired by the classic Bongard problems, adapted to the domain of programmatic graphics. Each task presents two sets of images: positive examples and negative examples. The images are generated from programs in the LOGO language, with the positive set sharing an underlying semantic concept that is absent from the negative set. Given these examples, the model must infer the underlying concept and correctly classify novel test instances as belonging to the positive or negative set.

### 4.2 Setting 1: Same-Model Perception

We first report the experiments corresponding to the first evaluation setting (Same-Model Perception) described in Section 3.3, where we compare the two-stage pipeline and the one-stage baseline using the same VLM. We use GPT-4o for Mini-ARC and Bongard-LOGO, while using LLaVA-1.5 (Liu et al., 2024a) for ACRE. Table 1 summarizes concrete settings for each dataset.

**Results.** We report the experiment results in Table 2. Across three datasets, enhancing perception through natural language descriptions consistently improves success rates by 11–13 percentage points. Notably, on Mini-ARC this corresponds to a  $2.5\times$  relative improvement, increasing performance from 8.05% to 20.13%. These results are consistent with our expectation for Setting 1 described in Section 3.3, which supports our hypothesis that per-

Table 1: Experimental setups for each dataset in Setting 1 (Same-Model Perception). We have two configurations (a) and (b) for each dataset. Stage: ‘‘S’’ refers to standard one-stage pipeline; ‘‘P+R’’ refers to two-stage pipeline with separate perception (‘‘P’’) and reasoning (‘‘R’’).

ID	Dataset	Config	Stage	Model
1	Mini-ARC	(a)	S	GPT-4o
		(b)	P+R	GPT-4o (P) + GPT-4o (R)
2	Bongard-LOGO	(a)	S	GPT-4o
		(b)	P+R	GPT-4o (P) + GPT-4o (R)
3	ACRE	(a)	S	LLaVA-1.5
		(b)	P+R	LLaVA-1.5 (P) + LLaVA-1.5 (R)

ception plays a more important role for the success in these tasks.

Table 2: Success rates (%) in Setting 1 (Same-Model Perception). The columns (a) and (b) correspond to the configurations defined in Table 1.  $\Delta$  refers to the absolute improvement from (a) to (b) in percentage points.

ID	Dataset	(a)	(b)	$\Delta$
1	Mini-ARC	8.05	20.13	+12.08
2	Bongard-LOGO	62.00	73.00	+11.00
3	ACRE	22.00	34.50	+12.50

One potential concern is that the reasoning stage in the two-stage pipeline receives both the raw images and the generated textual descriptions, so improvements over the one-stage baseline might arise from multimodal fusion or increased input context rather than from the language representation itself. To test this, we compare three variants for the reasoning stage on Bongard-LOGO and ACRE: image-only, text-only, and image+text. As shown in Table 3, text-only already recovers most of the gain over image-only, while adding the raw image on top of the text provides only marginal additional improvement. If the gains were mainly due to increased context length or multimodal fusion, image+text would be expected to substantially outperform text-only. Instead, the results suggest that the dominant benefit comes from the structured language representation, which mitigates perceptual difficulty and supports our claim that perception, rather than reasoning, is the main bottleneck.

### 4.3 Setting 2: Stronger-Model Perception

We further conduct experiments corresponding to the second evaluation setting (Stronger-Model Perception) described in Section 3.3, where we use a stronger VLM for the perception stage in the two-

Data	Model	Img.	Text	Img.+Text
Bongard-LOGO	GPT-4o	62.0	70.0	73.0
ACRE	LLaVA-1.5	22.0	32.0	34.5

Table 3: Ablation on the reasoning-stage input format. Text-only recovers most of the gain over image-only, while adding raw images yields only marginal further improvement.

stage pipeline. We conduct these experiments on the three datasets mentioned above, where we replace the weaker model used in the perception stage with a stronger model. We also compare against standard one-stage pipelines that use a weak model or a strong model. For Mini-ARC, we consider two different strong models: o1 and Claude-Sonnet-4.5. The concrete configurations are summarized in Table 4.

Table 4: Experimental setups in Setting 2 (Stronger-Model Perception). Configurations (a) and (b) are identical to those in Table 1. Configurations (c) and (d) introduce stronger models either in the perception stage or in a unified one-stage pipeline.

ID	Dataset	Config	Model(s)
1	Mini-ARC	(a)	GPT-4o (S)
		(b)	GPT-4o (P) + GPT-4o (R)
		(c1)	o1 (P) + GPT-4o (R)
		(c2)	Claude-Sonnet-4.5 (P) + GPT-4o (R)
		(d1)	o1 (S)
		(d2)	Claude-Sonnet-4.5 (S)
2	Bongard-LOGO	(a)	GPT-4o (S)
		(b)	GPT-4o (P) + GPT-4o (R)
		(c)	o1 (P) + GPT-4o (R)
		(d)	o1 (S)
3	ACRE	(a)	LLaVA-1.5 (S)
		(b)	LLaVA-1.5 (P) + LLaVA-1.5 (R)
		(c)	GPT-4o (P) + LLaVA-1.5 (R)
		(d)	GPT-4o (S)

Table 5: Success rates (%) under Setting 2 (Stronger-Model Perception). Configurations (a)–(d) correspond to those defined in Table 4.

Dataset	(a)	(b)	(c*)	(d*)
Mini-ARC	8.05	20.13	(c1) 31.54 (c2) 32.89	(d1) 52.03 (d2) 34.22
Bongard-LOGO	62.00	73.00	80.00	78.00
ACRE	22.00	34.50	82.50	93.00

**Results.** As can be seen in Table 5, strengthening the perception module ((b)  $\rightarrow$  (c\*)<sup>3</sup>) yields consistent and substantial improvement, indicating that

<sup>3</sup>We use (c\*) as a general reference to (c), (c1), and (c2). Same for (d\*) below.

Table 6: Error attribution across datasets and experiment configurations. Regarding the settings, for example, “1(a)” refers to the configuration (a) on Mini-ARC, while “3(c)” refers to the configuration (c) on ACRE.

(a) Mini-ARC			(b) Bongard-LOGO		
Setting	1(a)	1(b)	Setting	2(a)	a(b)
<b>Total Errors</b>	44	37	<b>Total Errors</b>	38	27
Perception (Demo)	38 (86.4%)	22 (59.5%)	Perception (Demo)	25 (65.8%)	10 (37.0%)
Reasoning (Inductive)	4 (9.1%)	9 (24.3%)	Reasoning (Inductive)	5 (13.2%)	12 (44.4%)
Perception (Test)	1 (2.3%)	2 (5.4%)	Perception (Test)	7 (18.4%)	3 (11.1%)
Reasoning (Deductive)	1 (2.3%)	4 (10.8%)	Reasoning (Deductive)	1 (2.6%)	2 (7.4%)

(c) ACRE			(d) ACRE (varying P)		
Setting	3(a)	3(b)	Setting	3(b)	3(c)
<b>Total Errors</b>	38	32	<b>Total Errors</b>	32	9
Perception (Demo)	29 (76.3%)	22 (68.8%)	Perception (Demo)	22 (68.8%)	0 (0%)
Reasoning (Inductive)	6 (15.8%)	7 (21.9%)	Reasoning (Inductive)	7 (21.9%)	9 (100%)
Perception (Test)	3 (7.9%)	3 (9.4%)	Perception (Test)	3 (9.4%)	0 (0%)
Reasoning (Deductive)	0 (0%)	0 (0%)	Reasoning (Deductive)	0 (0%)	0 (0%)

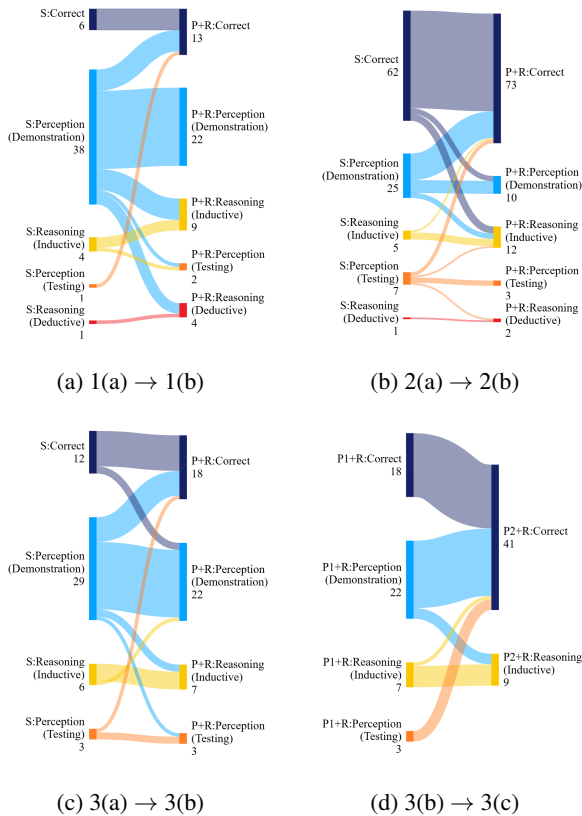


Figure 4: Visualization of how errors evolve from one configuration to another.

perception is indeed the dominant bottleneck in this setting. Furthermore, the performance in (c\*) and that in (d\*), the standard one-stage pipeline with the strong model, are mostly close<sup>4</sup>, suggest-

<sup>4</sup>One exception is Mini-ARC with o1, where the gap between (c1) and (d1) remains large. A plausible explanation is that Mini-ARC may have been included in o1’s training data. This possibility is supported by the observation that o1’s performance on Mini-ARC (d1) is markedly higher than that of Claude-Sonnet-4.5 (d2), while its performance on ARC-AGI

ing that perception constitutes the main bottleneck explaining the performance difference between the strong model (d\*) and the weak model (a).

#### 4.4 Error Attribution

In order to further validate the hypothesis and gain insights into the causes of errors, we conduct error attribution on the failure cases for different configurations<sup>5</sup> across datasets, following the protocol introduced in Section 3.4. Specifically, for configurations on Mini-ARC and ACRE, we randomly select 50 tasks, and for those on Bongard-LOGO, we randomly select 100 tasks. Among these, some tasks may have been correctly solved by the model, while for the remaining ones we perform detailed error attribution.

**Results.** As shown in Table 6, perception errors consistently dominate across all setups, indicating that the visual understanding stage is the primary bottleneck. On Mini-ARC, perception errors account for 86.4% of all errors in setting 1(a) (standard one-stage) and 59.5% in setting 1(b) (two-stage). A similar trend is observed on Bongard-LOGO, where perception errors represent 65.8% (2(a)) and 37.0% (2(b)) of total errors, and on ACRE, where perception errors reach 76.3% (3(a)) and 68.8% (3(b)). On ACRE with varying P, where different perception modules are compared, the model with LLaVA1.5 still shows 68.8% perception errors, while GPT-4o eliminates perception errors, with all remaining errors in reasoning stages.

is inferior to Claude-Sonnet-4.5.

<sup>5</sup>We omitted configurations with o1 or Claude-Sonnet-4.5 as we did not have full reasoning traces in our results.

These results clearly demonstrate that the perception stage remains the dominant source of model failure, overshadowing reasoning-related errors (both inductive and deductive). Moreover, this fine-grained error attribution provides more direct evidence that the performance improvement in both Setting 1 and Setting 2 are achieved through the mitigation of perception errors. This result highlights the importance of perception in tasks that are often regarded as reasoning tasks.

To gain deeper insight into this attribution, we further analyze how error types evolve across experimental configurations, specifically, how perception-related errors transition into reasoning errors (and vice versa) under different configurations, shown in Figure 4. From the analysis, we observe that a substantial portion of perception errors are either eliminated or transformed into downstream reasoning errors, and that the majority of performance gains arise from resolving perception-related errors rather than improving reasoning.

## 5 Conclusion

In this work, we critically re-examine the interpretation of ARC-style benchmarks as direct measures of machine reasoning ability and show that a substantial portion of the observed human–AI performance gap is driven instead by limitations in visual perception. By introducing a carefully controlled two-stage pipeline that explicitly separates perception from reasoning, and by conducting fine-grained error attribution across multiple benchmarks, we demonstrate that mitigating perceptual bottlenecks alone leads to large performance gains, while improvements in reasoning play a secondary role. These findings suggest that ARC-style benchmarks conflate perceptual and inductive challenges, potentially overstating deficiencies in model reasoning. More broadly, our results underscore the importance of evaluation protocols that disentangle perception from reasoning when assessing progress toward general intelligence, and point to the need for future benchmarks that more cleanly isolate the cognitive capabilities they aim to measure.

## Limitations

Our study has several limitations. First, the natural-language descriptions used in the perception stage should not be interpreted as an optimal or canonical intermediate representation for abstract reasoning. We adopt language as a convenient and inter-

pretable way to inject generic human perceptual priors and to demonstrate the existence and impact of perceptual bottlenecks in ARC-style benchmarks. We do not claim that natural language is the best representation for isolating reasoning, nor that it preserves all aspects of the original task structure without distortion.

Second, our fine-grained error attribution relies on manual inspection of model outputs and reasoning traces. While this attribution follows a clearly defined, stage-wise protocol and yields consistent patterns across datasets and experimental settings, it involves some subjectivity. We leave the development of scalable and fully automated attribution methods to future work.

Finally, our conclusions are limited to ARC-style, knowledge-light visual abstraction benchmarks and to the vision-language models evaluated in this study. We do not claim that perception is the dominant bottleneck for all multimodal or textual reasoning tasks. Nevertheless, we hope that the diagnostic perspective and methodology introduced in this work can inspire future research on disentangling perception and reasoning in a broader range of reasoning benchmarks and evaluation settings.

## References

- Ekin Akyürek, Mehul Damani, Adam Zweiger, Linlu Qiu, Han Guo, Jyothish Pari, Yoon Kim, and Jacob Andreas. 2025. [The surprising effectiveness of test-time training for few-shot learning](#). In *Proceedings of the 42nd International Conference on Machine Learning*, volume 267 of *Proceedings of Machine Learning Research*, pages 942–963. PMLR.
- ARC Prize Foundation. 2025. [Arc-agi-1 benchmark](#). Web page. Accessed 2025-11-22.
- ARC Prize, Inc. 2025. [Arc-agi-1 leaderboard](#).
- Mark Chen, Jerry Tworek, Heewoo Jun, Qiming Yuan, Henrique de Oliveira Pinto, Jared Kaplan, Harri Edwards, Yuri Burda, Nicholas Joseph, Greg Brockman, Alex Ray, Gaurav Puri, Gretchen Krueger, Michael Petrov, Heidy Khlaaf, Girish Sastry, Pamela Mishkin, Brooke Chan, Scott Gray, and 34 others. 2021. Evaluating large language models trained on code. *arXiv preprint arXiv:2107.03374*.
- Francois Chollet, Mike Knoop, Gregory Kamradt, Bryan Landers, and Henry Pinkard. 2025. Arc-agi-2: A new challenge for frontier ai reasoning systems. *arXiv preprint arXiv:2505.11831*.
- François Chollet. 2019. On the measure of intelligence. *arXiv preprint arXiv:1911.01547*.
- François Chollet, Mike Knoop, Gregory Kamradt, and Bryan Landers. 2024. [Arc prize 2024: Technical report](#). *arXiv preprint arXiv:2412.04604*.
- Karl Cobbe, Vineet Kosaraju, Mohammad Bavarian, Mark Chen, Heewoo Jun, Lukasz Kaiser, Matthias Plappert, Jerry Tworek, Jacob Hilton, Reiichiro Nakano, and 1 others. 2021. Training verifiers to solve math word problems. *arXiv preprint arXiv:2110.14168*.
- Daniel Franzen, Jan Disselhoff, and David Hartmann. 2025. [The architects - technical report: Arc prize 2025 solution](#). Accessed: 2026-01-04.
- DeepMind / Google. 2025. [Gemini — our most intelligent ai models](#). Web page. Accessed 2025-11-22.
- Dan Hendrycks, Collin Burns, Steven Basart, Andy Zou, Mantas Mazeika, Dawn Song, and Jacob Steinhardt. 2020. Measuring massive multitask language understanding. *arXiv preprint arXiv:2009.03300*.
- Keya Hu, Ali Cy, Linlu Qiu, Xiaoman Delores Ding, Runqian Wang, Yeyin Eva Zhu, Jacob Andreas, and Kaiming He. 2025. Arc is a vision problem! *arXiv preprint arXiv:2511.14761*.
- Alexia Jolicoeur-Martineau. 2025. Less is more: Recursive reasoning with tiny networks. *arXiv preprint arXiv:2510.04871*.
- Subin Kim, Prin Phunyahibarn, Donghyun Ahn, and Sundong Kim. 2022. Playgrounds for abstraction and reasoning. In *NeurIPS 2022 Workshop on Neuro Causal and Symbolic AI (nCSI)*.
- Wen-Ding Li, Keya Hu, Carter Larsen, Yuqing Wu, Simon Alford, Caleb Woo, Spencer M Dunn, Hao Tang, Michelangelo Naim, Dat Nguyen, and 1 others. 2024. Combining induction and transduction for abstract reasoning. *arXiv preprint arXiv:2411.02272*.
- Isaac Liao and Albert Gu. 2025. Arc-agi without pre-training. *arXiv preprint arXiv:2512.06104*.
- Haotian Liu, Chunyuan Li, Yuheng Li, and Yong Jae Lee. 2024a. Improved baselines with visual instruction tuning. In *Proceedings of the IEEE/CVF conference on computer vision and pattern recognition*, pages 26296–26306.
- Yuan Liu, Haodong Duan, Yuanhan Zhang, Bo Li, Songyang Zhang, Wangbo Zhao, Yike Yuan, Jiaqi Wang, Conghui He, Ziwei Liu, and 1 others. 2024b. Mmbench: Is your multi-modal model an all-around player? In *European conference on computer vision*, pages 216–233. Springer.
- Pan Lu, Hritik Bansal, Tony Xia, Jiacheng Liu, Chunyuan Li, Hannaneh Hajishirzi, Hao Cheng, Kai-Wei Chang, Michel Galley, and Jianfeng Gao. 2023. Mathvista: Evaluating mathematical reasoning of foundation models in visual contexts. *arXiv preprint arXiv:2310.02255*.
- Pan Lu, Swaroop Mishra, Tanglin Xia, Liang Qiu, Kai-Wei Chang, Song-Chun Zhu, Oyvind Tafjord, Peter Clark, and Ashwin Kalyan. 2022. Learn to explain: Multimodal reasoning via thought chains for science question answering. *Advances in Neural Information Processing Systems*, 35:2507–2521.
- Minesh Mathew, Dimosthenis Karatzas, and CV Jawahar. 2021. Docvqa: A dataset for vqa on document images. In *Proceedings of the IEEE/CVF winter conference on applications of computer vision*, pages 2200–2209.
- Melanie Mitchell. 2024. [Did openai just solve abstract reasoning?](#) AI: A Guide for Thinking Humans (Substack). Accessed 2025-01-04.
- Weili Nie, Zhiding Yu, Lei Mao, Ankit B Patel, Yuke Zhu, and Anima Anandkumar. 2020. Bongard-logo: A new benchmark for human-level concept learning and reasoning. *Advances in Neural Information Processing Systems*, 33:16468–16480.
- OpenAI. 2024. [Introducing o3 and o3-mini \(announcement video\)](#). YouTube video. Accessed 2025-11-22.
- David Rein, Betty Li Hou, Asa Cooper Stickland, Jackson Petty, Richard Yuanzhe Pang, Julien Dirani, Julian Michael, and Samuel R Bowman. 2024. Gpqa: A graduate-level google-proof q&a benchmark. In *First Conference on Language Modeling*.
- Amanpreet Singh, Vivek Natarajan, Meet Shah, Yu Jiang, Xinlei Chen, Dhruv Batra, Devi Parikh, and Marcus Rohrbach. 2019. Towards vqa models that can read. In *Proceedings of the IEEE/CVF conference on computer vision and pattern recognition*, pages 8317–8326.

- Mirac Suzgun, Nathan Scales, Nathanael Schärli, Sebastian Gehrmann, Yi Tay, Hyung Won Chung, Aakanksha Chowdhery, Quoc Le, Ed Chi, Denny Zhou, and 1 others. 2023. Challenging big-bench tasks and whether chain-of-thought can solve them. In *Findings of the Association for Computational Linguistics: ACL 2023*, pages 13003–13051.
- Guan Wang, Jin Li, Yuhao Sun, Xing Chen, Changling Liu, Yue Wu, Meng Lu, Sen Song, and Yasin Abbasi Yadkori. 2025. Hierarchical reasoning model. *arXiv preprint arXiv:2506.21734*.
- Zihan Weng, Lucas Gomez, Taylor Whittington Webb, and Pouya Bashivan. 2025. Caption this, reason that: Vlms caught in the middle. *arXiv preprint arXiv:2505.21538*.
- Xiang Yue, Yuansheng Ni, Kai Zhang, Tianyu Zheng, Ruoqi Liu, Ge Zhang, Samuel Stevens, Dongfu Jiang, Weiming Ren, Yuxuan Sun, and 1 others. 2024. Mmmu: A massive multi-discipline multimodal understanding and reasoning benchmark for expert agi. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, pages 9556–9567.
- Xiang Yue, Tianyu Zheng, Yuansheng Ni, Yubo Wang, Kai Zhang, Shengbang Tong, Yuxuan Sun, Botao Yu, Ge Zhang, Huan Sun, and 1 others. 2025. Mmmu-pro: A more robust multi-discipline multimodal understanding benchmark. In *Proceedings of the 63rd Annual Meeting of the Association for Computational Linguistics (Volume 1: Long Papers)*, pages 15134–15186.
- Chi Zhang, Baoxiong Jia, Mark Edmonds, Song-Chun Zhu, and Yixin Zhu. 2021. Acre: Abstract causal reasoning beyond covariation. In *Proceedings of the IEEE/CVF conference on computer vision and pattern recognition*, pages 10643–10653.

## Appendix

### 5.1 Examples

#### 5.1.1 Example of Prompt

In this section, we provide an example prompt for the perception stage of the Mini-ARC problem.

You are a precise visual perception system for ARC puzzles. Your task is to provide objective, structured observations of a single grid to help downstream reasoning models discover transformation rules.

Note that 0 corresponds to the background color (and does not lead to objects), and different values represent different colors. Write the following in natural language:

- Write about the shape and location of the objects in the grid in natural language. Some objects can be large or/and overlapping with each other.
- Write about any additional patterns in the grid in natural language.
- Write about the dominant colors in the grid in natural language

#### 5.1.2 Example of Generated Natural Language Description

Below we show an example output of the perception stage for the first input in 1a.

I'll analyze this grid systematically.\n\n\*\*Shape and Location of Objects:\*\*\n\nThe grid contains several small objects:\n- In the top-left corner (row 0, columns 0-1), there are two cells with value 3 forming a horizontal pair\n- A single isolated cell with value 3 appears at row 2, column 2, positioned roughly in the center of the grid\n- In the bottom-right area, there's an L-shaped or angular object formed by values 1 and 4: value 1 appears at positions (3,3) and forms a small vertical line with another 1 at (4,3) and (4,4), while value 4 appears once at position (3,4)\n- The objects are distributed across different regions of the grid rather than clustered together\n\n\*\*Additional Patterns:\*\*\n\nThe grid shows a diagonal-like distribution of non-zero values from top-left to bottom-right. The value 3 appears in two separate locations (top-left and center), suggesting possible grouping or repetition. The bottom-right cluster combines two different non-zero values (1 and 4) that are adjacent to each other, forming a connected component.\n\n\*\*Dominant Colors:\*\*\n\nThe dominant non-zero values in the grid are:\n- Value 3,

which appears in multiple locations\n- Value 1, which appears multiple times in the bottom-right\n- Value 4, which appears once\n\nValue 3 and value 1 are the most prevalent non-zero colors, with value 4 being the least common.\n\n\*\*Number of Cells for Each Color:\*\*\n- Value 3: 3 cells total\n- Value 1: 3 cells total\n- Value 4: 1 cell total\n- Value 0 (background): 18 cells total

#### 5.1.3 Example of perception and reasoning error

In this section we provide a sample perception and a sample reasoning error when models are trying to solve problem described in 1a. A sample perception error:

Training Example 1:

Input: ....

Output: All numbers except the ones with value 3 are turned to 0. In this specific case, only the original positions of 3 remained; all else became 0.

Training Example 2:

Input: .....

Output: Only the positions originally containing 3 have retained their value, but others, including numbers 1 and 4, turned to 0.

Here, the model incorrectly describes the output of training example 2 since it wrongly perceives that all 3s remain in the original position.

A sample reasoning error:

Test: Since the yellow square is located in line 4, the input 3s should remain as in Ex1

In this example, the model correctly perceives the training demonstrations and the test input. However, when inferring the rule, instead of linking the position of the yellow square relative to the blue squares to the rotation rule, it simply asserts that because the yellow square is on line 4 the 3s should remain as in Training Example 1. This indicates a reasoning error (incorrect rule induction) rather than a perception error.