RIGHT ANSWERS, WRONG REASONS: DISSOCIATING UNDERSTANDING FROM CORRECTNESS IN LLM REASONING

Vimanyu Taneja & Soumya Banerjee
Department of Computer Science and Technology
University of Cambridge
Cambridge, UK
{vt329,sb2333}@cam.ac.uk

ABSTRACT

Accuracy is the standard metric for evaluating LLM reasoning, but it conflates two distinct capabilities: understanding the underlying concept and executing it correctly. We introduce a two-phase evaluation framework that separates these concerns. A solver model attempts ConceptARC tasks requiring inductive reasoning from examples. A separate judge model evaluates only the reasoning trace, scoring conceptual understanding independent of output correctness. Across 480 evaluations (160 tasks $\times$ 3 passes), we find 38% show a mismatch: correct answers from flawed reasoning, or incorrect answers despite sound understanding. We analyze failure patterns across concept types, finding systematic weaknesses in spatial reasoning (Cohen's $d = 1.53$) and 34% inconsistency across repeated attempts. Our results suggest accuracy alone significantly misrepresents reasoning capability, and different failure modes require different interventions.

1 INTRODUCTION

When an LLM solves a reasoning problem correctly, did it actually understand the underlying logic? Standard benchmarks report accuracy, treating correct outputs as evidence of reasoning capability. But a model can arrive at correct answers through pattern matching, memorization, or lucky guesses, without genuine conceptual understanding. Conversely, a model might understand a concept perfectly but make execution errors that produce incorrect outputs.

If we cannot distinguish understanding from correctness, we cannot diagnose *why* models fail, design targeted improvements, or trust models in high-stakes applications where reasoning matters.

We introduce a two-phase evaluation framework that measures understanding and correctness independently. Applied to ConceptARC (Moskvichev et al., 2023), a benchmark requiring induction of transformation rules from examples, we find that 38% of evaluations show a mismatch between understanding and correctness. This has significant implications: accuracy alone would misclassify more than one-third of cases.

Our contributions are: (1) a two-phase evaluation methodology separating reasoning quality from output correctness; (2) evidence that 38% of evaluations show understanding-correctness mismatch; and (3) analysis of systematic failure patterns, including spatial reasoning weakness ($d = 1.53$), 34% inconsistency across attempts, and longer reasoning traces correlating with failure. Code and results are available at https://github.com/vimanyu-taneja/conceptarc-reasoner-eval.

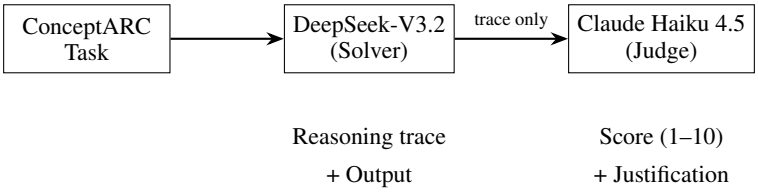


Figure 1: Two-phase evaluation pipeline. The solver generates reasoning traces and outputs. The judge evaluates only the reasoning trace, blind to output correctness, producing a score and justification.

2 METHODOLOGY

2.1 TASK: CONCEPTARC

ConceptARC (Moskvichev et al., 2023) is a benchmark for abstract visual reasoning organized around 16 concepts (e.g., Copy, Center, InsideOutside). Each task provides 2–3 input-output grid examples demonstrating a transformation rule. The model must induce the rule and apply it to new test inputs. We use 160 tasks (10 per concept), requiring inductive reasoning and generalization to novel inputs. Appendix A lists all concepts with their descriptions, and Appendix C shows example tasks from each concept.

2.2 TWO-PHASE EVALUATION PIPELINE

Our evaluation separates reasoning quality from output correctness through two phases (Figure 1).

Phase 1: Solver. DeepSeek-V3.2 in Thinking Mode (DeepSeek-AI, 2025) generates predictions with extended reasoning traces (chain-of-thought). We used `max_tokens=64000` because initial runs with the default 32K limit resulted in 13 truncated responses (2.7%) across 10 tasks and 6 concepts, with the CleanUp concept most affected.

Phase 2: Judge. Claude Haiku 4.5 reads only the reasoning trace and concept name, not the output or whether it was correct. It produces a score (1–10) for conceptual understanding and a short justification explaining the rating.

This separation allows us to detect correct answers from flawed reasoning, and incorrect answers despite sound reasoning.

2.3 SCORING PROMPT DEVELOPMENT

We iteratively refined the scoring rubric to distinguish genuine conceptual understanding from mechanical pattern descriptions. The final rubric scores four independent dimensions: (1) rule-concept alignment (whether the derived rule matches the concept), (2) concept articulation (whether the model explicitly names or describes the concept), (3) application correctness (whether the rule is applied systematically), and (4) explanatory depth (whether the model explains *why* the rule works, not just what it does). The full prompts are provided in Appendix B.

Scores converge to 7–9 for most evaluations, which is expected: the model grasps most concepts adequately. Lower scores (1–4) indicate clear conceptual failures. Perfect scores (10) are rare because flawless articulation is uncommon even for correct solutions. Figure 2 shows the score distribution, with a strong peak at 8 (47%) and a secondary cluster at 2–3 (26%) representing cases of poor reasoning. This bimodal pattern suggests the rubric successfully discriminates between adequate and inadequate understanding.

2.4 EVALUATION PROTOCOL

Each task was evaluated 3 times to reduce sampling variance. We define "high understanding" as score ≥ 5, separating low understanding (1–4) from adequate understanding (5+). Total evaluations: 480 (160 tasks $\times$ 3 passes). Statistical comparisons use Welch's t-test and Cohen's d for effect size.

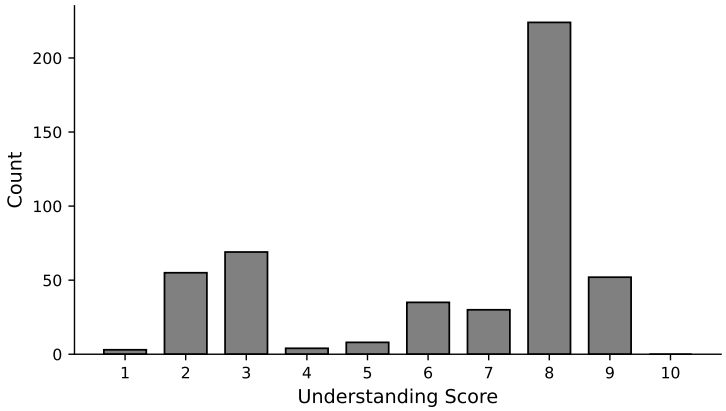


Figure 2: Distribution of understanding scores. Scores cluster at 8 (47%), with scores 2–3 well-populated for poor reasoning, indicating the rubric successfully discriminates.

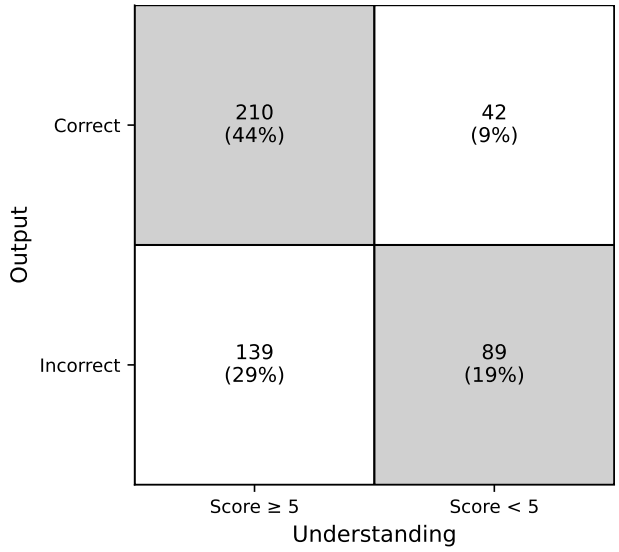


Figure 3: Understanding vs. correctness. 38% of evaluations show a mismatch: correct answers with low understanding (9%) or incorrect answers with high understanding (29%).

3 RESULTS

3.1 THE UNDERSTANDING-CORRECTNESS GAP

Figure 3 shows our central finding. We partition evaluations by correctness (correct/incorrect output) and understanding (score ≥ 5 or < 5).

The overall accuracy is 52.5%, with a mean understanding score of 6.37. The correlation between understanding score and correctness is weak ($r = 0.24$), suggesting these measure partially distinct capabilities. Most importantly, the mismatch rate is 38%: more than one-third of evaluations show a disconnect between understanding and correctness. The 9% of cases with correct answers but low understanding represent potential "lucky guesses" or pattern matching without genuine comprehension. The 29% with incorrect answers but high understanding represent execution failures despite sound reasoning. Accuracy alone would misclassify both categories.

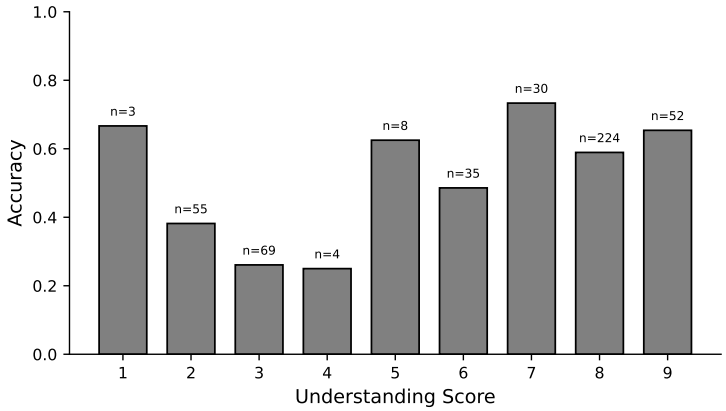


Figure 4: Calibration: accuracy by understanding score. The relationship is non-monotonic; score 7 has higher accuracy than scores 8 or 9.

3.1.1 QUADRANT EXAMPLES

To illustrate what each quadrant represents, we show the judge's justification for representative cases.

Correct + High Understanding (ExtractObjects5, score 9): *"The derived rule precisely captures the core mechanism: systematically identify maximal hollow squares, extract them as independent objects, and arrange them horizontally by column position. This directly implements the concept of identifying and isolating distinct objects."*

Correct + Low Understanding (TopBottom2D8, score 1): *"The model completely failed to understand the TopBottom2D concept. The reasoning derives a rule about finding the last non-zero cell and recoloring, entirely unrelated to how shapes interact or hide each other on a 2D grid."* This case produced a correct output despite fundamentally misunderstanding the task.

Incorrect + High Understanding (CompleteShape1, score 6): *"The model correctly identified the core concept of making shapes symmetric about a vertical axis and derived a plausible rule. It explicitly articulated the symmetry-based approach. However, the application reveals significant issues: the rule overgeneralizes by filling all zeros in boundary rows without properly restricting to the region between leftmost and rightmost non-zero cells (works for some cases but has execution errors)."* The model understood the concept but failed in execution.

Incorrect + Low Understanding (TopBottom3D6, score 1): *"The reasoning shows no understanding of the TopBottom3D concept. Instead, the model applies a completely unrelated rule about bounding boxes and aspect ratios, which has no connection to 3D perspective or depth layering."*

3.2 CALIBRATION

If understanding scores were perfectly calibrated, higher scores should correspond to higher accuracy. Figure 4 tests this by showing accuracy at each score level. The relationship is weak and non-monotonic: score 7 has higher accuracy (73%) than score 8 (59%) or score 9 (65%). This pattern suggests understanding and execution are partially independent skills. A model can understand a concept but fail to execute it correctly, or succeed through pattern matching without understanding.

3.3 CONCEPT-LEVEL ANALYSIS

Figure 5 shows accuracy vs. mean understanding score for each of the 16 concepts. If accuracy and understanding were equivalent, all points would lie along a diagonal. Instead, we observe substantial spread, with several notable patterns.

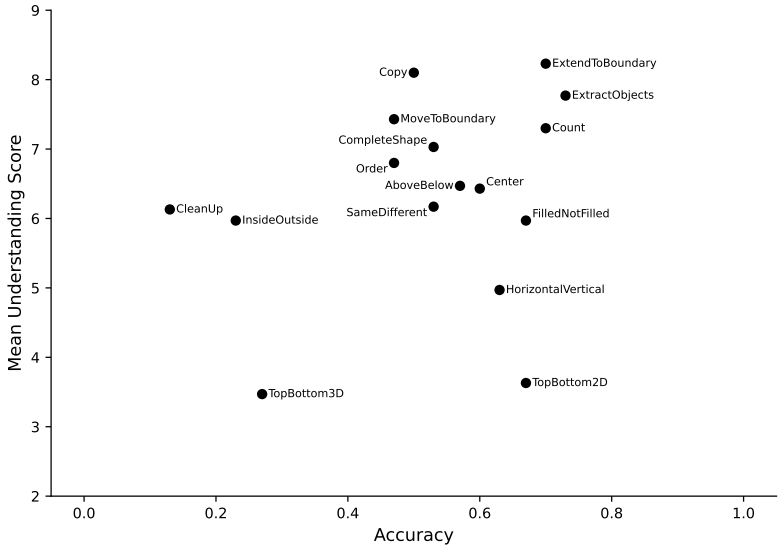


Figure 5: Accuracy vs. understanding by concept. Notable patterns: TopBottom2D has high accuracy (67%) but low understanding (3.6); Copy has high understanding (8.1) but only 50% accuracy.

Table 1: 2D/3D reasoning gap

Group	Mean Score	n
Other concepts	6.77	420
2D/3D concepts	3.55	60

Welch's $t = 10.4$, $p < 10^{-22}$, Cohen's $d = 1.53$ (large effect)

Concepts like ExtractObjects, Count, and ExtendToBoundary cluster in the upper-right, indicating the model genuinely learns these rules. Copy and MoveToBoundary appear in the upper-left region: high understanding but lower accuracy, suggesting execution errors despite sound reasoning. TopBottom2D is an outlier in the lower-right: high accuracy (67%) but low understanding (3.6), indicating pattern matching without genuine comprehension. TopBottom3D, CleanUp, and InsideOutside cluster in the lower-left: the model neither understands nor succeeds at these concepts.

3.4 THE 2D/3D GAP

Concepts requiring reasoning about layering and occlusion (TopBottom2D, TopBottom3D) show dramatically lower understanding scores (Figure 6). The mean score for 2D/3D concepts is 3.55 compared to 6.77 for other concepts. This difference is highly significant ($t = 10.4$, $p < 10^{-22}$) with a large effect size (Cohen's $d = 1.53$).

This is not random variation but a systematic weakness. The model fails to reason about depth, layering, and occlusion even when it produces correct outputs (often through pattern matching). This suggests a fundamental limitation in how the model represents spatial relationships.

3.5 FAILURE PATTERNS BY CATEGORY

To understand whether different types of concepts fail in different ways, we grouped the 16 concepts into five categories: 2D/3D (TopBottom2D, TopBottom3D), Spatial (AboveBelow, InsideOutside, Center), Comparison (FilledNotFilled, HorizontalVertical, SameDifferent), Transform (Copy, ExtendToBoundary, MoveToBoundary, CompleteShape), and Extraction (ExtractObjects, CleanUp, Count, Order).

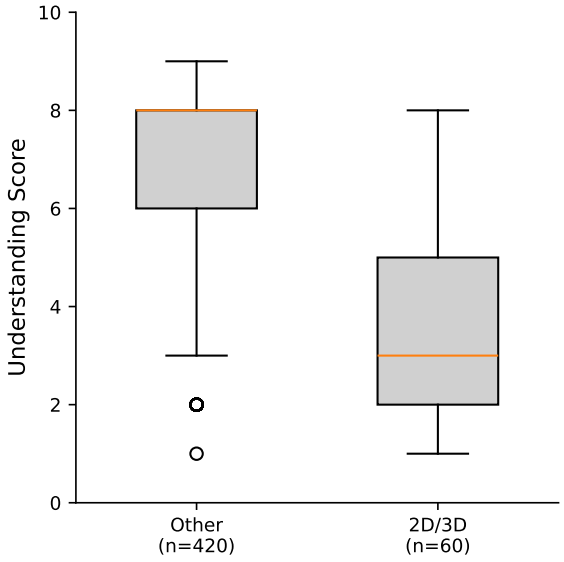


Figure 6: 2D/3D concepts show systematically lower understanding (mean 3.55 vs. 6.77, Cohen's $d = 1.53$).

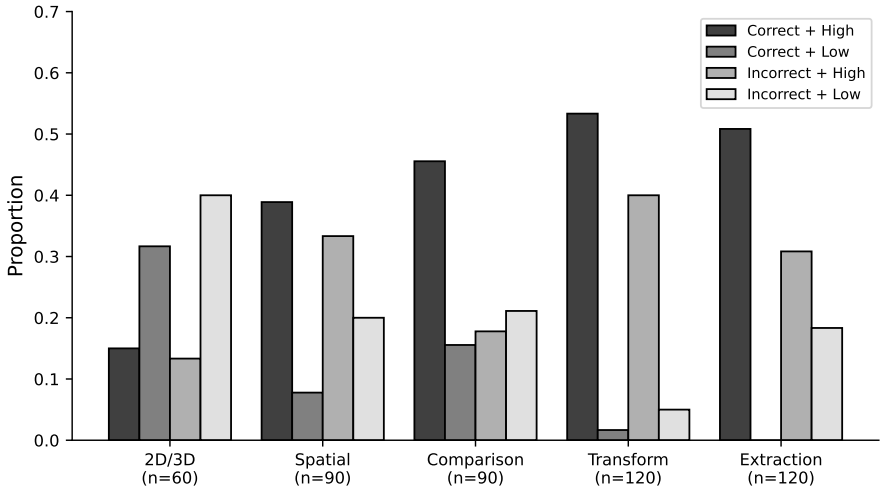


Figure 7: Quadrant distribution by concept category. 2D/3D concepts have the highest rate of correct answers with low understanding (32%); Transform concepts have the highest rate of incorrect answers with high understanding (40%).

Figure 7 breaks down the four quadrants by category. The patterns are striking. 2D/3D concepts have the highest rate of correct answers with low understanding (32%), confirming that success on these tasks often comes from pattern matching rather than genuine comprehension. Transform concepts have the highest rate of incorrect answers with high understanding (40%), indicating that the model understands these transformation rules but frequently makes execution errors. Extraction concepts show zero cases of correct answers with low understanding: when the model succeeds, it genuinely understood the task.

These patterns suggest that different concept types require different interventions. Improving performance on 2D/3D concepts may require better spatial representations, while improving Transform concepts may require better execution verification.

Table 2: Quadrant distribution by concept category

Category	n	Corr+High	Corr+Low	Incorr+High	Incorr+Low
2D/3D	60	15%	32%	13%	40%
Spatial	90	39%	8%	33%	20%
Comparison	90	46%	16%	18%	21%
Transform	120	53%	2%	40%	5%
Extraction	120	51%	0%	31%	18%

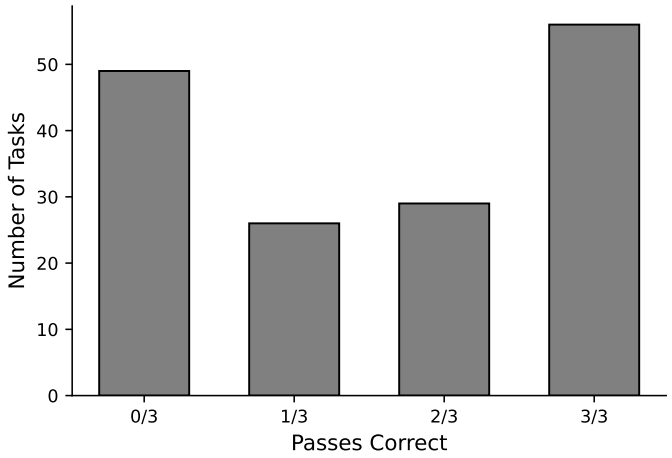


Figure 8: Correctness consistency across 3 passes. 34% of tasks are inconsistent (1/3 or 2/3 correct).

3.6 Consistency across attempts

Each task was run 3 times. Figure 8 shows how consistent the results are. Only 66% of tasks are fully consistent (either 0/3 or 3/3 correct). The remaining 34% produce different results across repeated attempts: the same input yields contradictory outputs depending on sampling.

This inconsistency has practical implications. A single evaluation may not reflect the model's true capability on a task. It also suggests that reasoning is not deterministic even for tasks the model "knows" how to solve.

3.7 Reasoning trace analysis

Figure 9 shows reasoning trace length by correctness. Incorrect answers produce 70% longer reasoning traces than correct answers (58k vs. 34k characters). This difference is highly significant ($t = -12.6, p < 10^{-31}$).

Extended deliberation correlates with failure, not success. When the model struggles, it produces verbose but unproductive reasoning. This has implications for inference-time compute scaling: more tokens do not necessarily lead to better reasoning.

4 Discussion

4.1 Implications for evaluation

Accuracy alone is misleading. A 52.5% accuracy score hides the fact that 9% of correct answers come from flawed reasoning, and 29% of incorrect answers come despite sound understanding. Benchmarks should consider reasoning process, not just output.

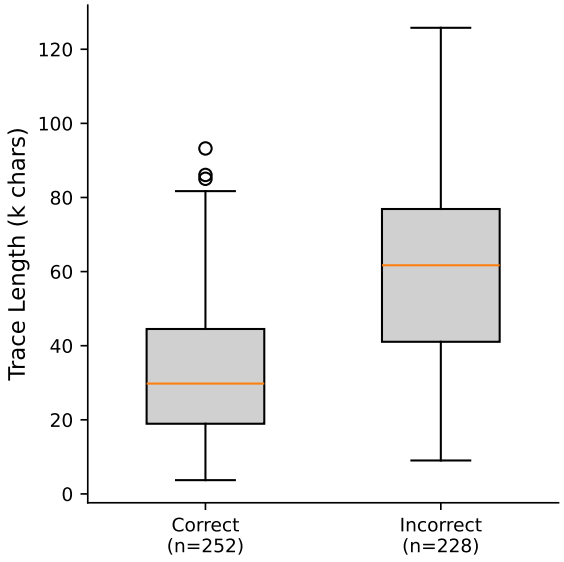


Figure 9: Incorrect answers produce 70% longer reasoning traces (58k vs. 34k chars, $p < 10^{-31}$).

4.2 Different failures need different interventions

Our analysis suggests a taxonomy of failure modes. Cases with correct answers but low understanding (pattern matching) may benefit from requiring explanations or conceptual grounding. Cases with incorrect answers but high understanding (execution errors) may benefit from better verification and self-correction mechanisms. Systematic gaps like the 2D/3D weakness may require different representations or architectural changes rather than more training data.

4.3 Limitations

Our study has several limitations. We use a single judge model; multiple judges could provide inter-rater reliability. The rubric is inherently subjective: what counts as "understanding" is debatable. The score ceiling effect (clustering at 8) suggests finer discrimination may be possible with refined rubrics. Finally, results are specific to DeepSeek-V3.2 and may differ for other models.

5 Related Work

ConceptARC. Moskvichev et al. (2023) introduced ConceptARC as a benchmark for abstract visual reasoning, showing humans substantially outperform AI systems including GPT-4. Our work extends this by evaluating reasoning quality, not just accuracy.

ARC. The Abstraction and Reasoning Corpus (Chollet, 2019) tests general fluid intelligence through novel reasoning tasks. ConceptARC organizes similar tasks around explicit concepts.

Process-based evaluation. Prior work has emphasized evaluating reasoning traces rather than just outputs (Lightman et al., 2023; Uesato et al., 2022). Our two-phase approach operationalizes this for abstract reasoning.

Self-consistency. Wang et al. (2023) showed sampling multiple outputs improves reliability. Our consistency analysis (Figure 8) reveals 34% of tasks produce contradictory results across attempts.

6 Conclusion

We introduced a two-phase evaluation framework that separates understanding from correctness in LLM reasoning. Applied to ConceptARC, we find a 38% mismatch rate between understanding

and correctness, systematic weakness in 2D/3D spatial reasoning (Cohen's $d = 1.53$), 34% inconsistency across repeated attempts, longer reasoning correlating with failure rather than success, and different concept types failing in different ways that require different interventions.

Accuracy is not understanding. Evaluation should consider reasoning process, not just output.

ACKNOWLEDGMENTS

This research was supported by the Accelerate Programme for Scientific Discovery at the University of Cambridge. The programme is funded by Schmidt Sciences.

REFERENCES

François Chollet. On the measure of intelligence, 2019. URL https://arxiv.org/abs/1911.01547.

DeepSeek-AI. Deepseek-v3.2: Pushing the frontier of open large language models, 2025. URL https://arxiv.org/abs/2512.02556.

Hunter Lightman, Vineet Kosaraju, Yura Burda, Harri Edwards, Bowen Baker, Teddy Lee, Jan Leike, John Schulman, Ilya Sutskever, and Karl Cobbe. Let's verify step by step, 2023. URL https://arxiv.org/abs/2305.20050.

Arseny Moskvichev, Victor Vikram Odouard, and Melanie Mitchell. The conceptarc benchmark: Evaluating understanding and generalization in the arc domain, 2023. URL https://arxiv.org/abs/2305.07141.

Jonathan Uesato, Nate Kushman, Ramana Kumar, Francis Song, Noah Siegel, Lisa Wang, Antonia Creswell, Geoffrey Irving, and Irina Higgins. Solving math word problems with process- and outcome-based feedback, 2022. URL https://arxiv.org/abs/2211.14275.

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, 2023. URL https://arxiv.org/abs/2203.11171.

A CONCEPT DEFINITIONS

The 16 ConceptARC concepts used in this study are defined as follows:

- **AboveBelow**: Identify and manipulate objects based on their vertical position relative to a horizontal dividing line.
- **Center**: Find or mark the center point of shapes or the grid.
- **CleanUp**: Remove noise, artifacts, or incomplete patterns from the grid while preserving the main structure or complete shapes.
- **CompleteShape**: Fill in missing parts of partially drawn shapes to make them complete and symmetric.
- **Copy**: Duplicate objects or patterns from one location to another within the grid, potentially with transformations.
- **Count**: Count objects, cells, or patterns and represent the count visually, often by filling cells or creating output grids sized by the count.
- **ExtendToBoundary**: Extend lines, shapes, or patterns until they reach the edge of the grid or another boundary.
- **ExtractObjects**: Identify and isolate distinct objects from the input grid, potentially outputting them separately or in a specific arrangement.
- **FilledNotFilled**: Distinguish between filled (solid) and unfilled (hollow) shapes.

- **HorizontalVertical**: Distinguish between horizontal (landscape) and vertical (portrait) lines, patterns, or orientations.
- **InsideOutside**: Distinguish between regions inside enclosed shapes versus regions outside, applying different rules to each.
- **MoveToBoundary**: Move objects from their current position to a boundary edge of the grid or another boundary.
- **Order**: Arrange objects in a specific order based on size, position, color, or other properties.
- **SameDifferent**: Identify objects that are the same or different from each other and apply transformations based on this classification.
- **TopBottom2D**: See the grid as a flat object. Reason about how top and bottom parts and shapes interact or hide each other from one side.
- **TopBottom3D**: See the grid as objects with depth. Decide what is in front, behind or hidden when shapes interact from a 3D perspective.

B PROMPTS

B.1 SOLVER PROMPTS

System prompt:

> You are DeepSeek-Reasoner, an advanced AI specialized in abstract reasoning tasks. You excel at pattern recognition, spatial reasoning, and inferring transformation rules from examples.
>
> When solving ConceptARC tasks:
>
> 1. Analyze each training example carefully
> 2. Identify patterns, regularities, and the transformation rule
> 3. Explain your reasoning step by step
> 4. Apply the rule to produce the output
>
> Always provide your final answer as valid JSON with an `"outputs"` key.

User prompt template:

> You are solving a ConceptARC task. ConceptARC tasks involve 2D grids of colored cells (represented as integers 0–9, where 0 is typically the background).
>
> You are given training input/output grid pairs that demonstrate a transformation rule. Your task is to:
>
> 1. Carefully analyze the training examples to infer the transformation rule
> 2. Apply the same rule to the test input(s) to produce the output(s)
>
> Think step-by-step in detail. Consider:
>
> - What changes between input and output?
> - Are there patterns based on color, position, shape, or other properties?
> - What is the underlying concept or rule?

```
TRAINING EXAMPLES:
[Examples provided here]
TEST INPUT:
[Test input provided here]
```

> After your reasoning, you MUST output your final answer as a JSON object with an `"outputs"` key containing a list of output grids.

B.2 JUDGE PROMPTS

System prompt:

You are a precise evaluator of conceptual understanding in AI reasoning traces. Your task is to assess how well a model understood and applied a specific concept, independent of whether it got the final answer correct.

Be objective and base your evaluation solely on the reasoning trace provided. Look for:

- Explicit mentions of the concept or related ideas
- Correct application of the concept in reasoning
- Evidence of understanding vs. lucky guessing

Always respond with valid JSON containing `"score"` and `"justification"` keys.

User prompt template:

You are evaluating whether a model correctly understood a ConceptARC concept based on its reasoning trace.

```
CONCEPT NAME: [Concept name]
CONCEPT DEFINITION: [Description from Appendix A]
```

Evaluation Method. Score four independent dimensions using strict criteria. Be discriminating.

Dimension 1: Rule-Concept Alignment (1–4 points)

- **1**: Rule contradicts or is completely unrelated to the concept
- **2**: Rule has some connection but misses the core mechanism
- **3**: Rule captures the concept but with imprecision or missing details
- **4**: Rule precisely matches the concept definition with all key elements (rare)

Dimension 2: Concept Articulation (0–2 points)

- **0**: Never references the concept; purely mechanical pattern description
- **1**: Uses related terms or describes the concept indirectly
- **2**: Explicitly names the concept or uses unambiguous equivalent language

Dimension 3: Application Correctness (0–2 points)

- **0**: Rule application is incorrect or absent
- **1**: Partially correct; works for some examples but not all, or has execution errors
- **2**: Fully correct systematic application to all examples

Dimension 4: Explanatory Depth (0–2 points)

- **0**: No explanation of reasoning; just states what happens
- **1**: Explains the pattern/rule but not why it relates to the concept
- **2**: Connects the rule to the concept's purpose; explains the "why" not just "what" (rare)

Scoring. Final Score = D1 + D2 + D3 + D4 (range: 1–10). Expected distribution: 1–4 = wrong concept or fundamental misunderstanding; 5–7 = partial understanding or correct but mechanical; 8–10 = strong understanding with good articulation (10 is rare).

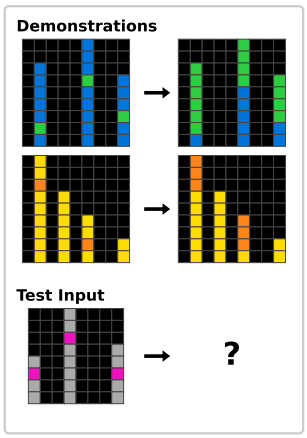
```
REASONING TRACE: [Trace provided here]
```

Respond with only a JSON object with keys `"score"` (1–10) and `"justification"` (2–3 sentences explaining the score).

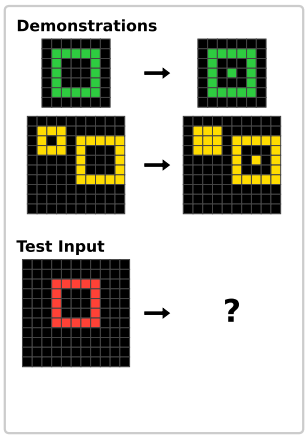
C TASK ILLUSTRATIONS

The following figures show one example task from each of the 16 ConceptARC concept groups. Each task shows the training demonstrations (input-output pairs) and the test input that the model must solve.

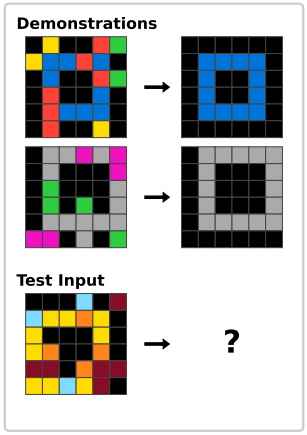
Above Below



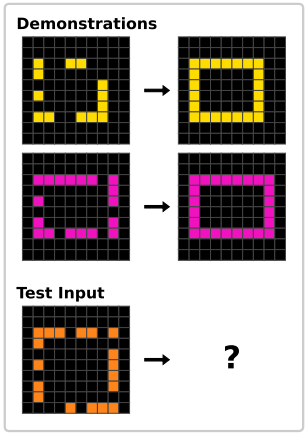
Center



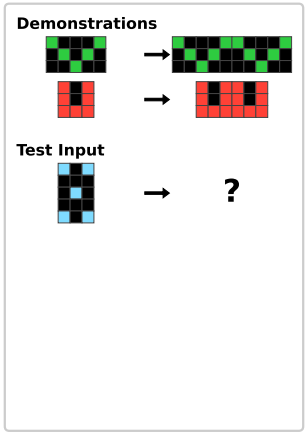
Clean Up



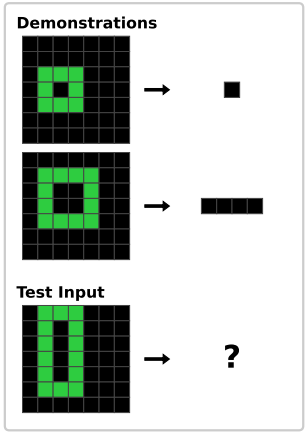
Complete Shape



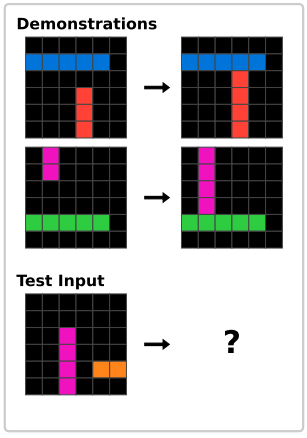
Copy



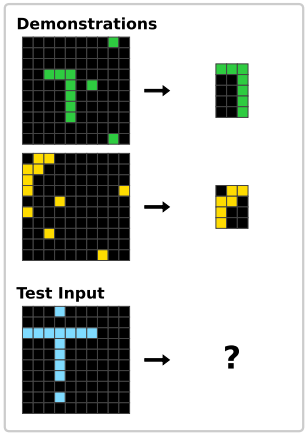
Count



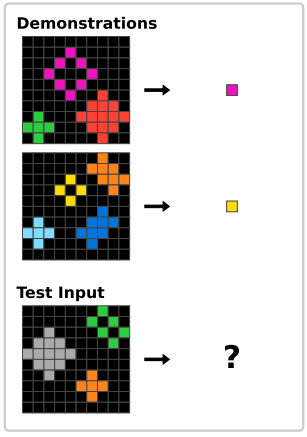
Extend To Boundary



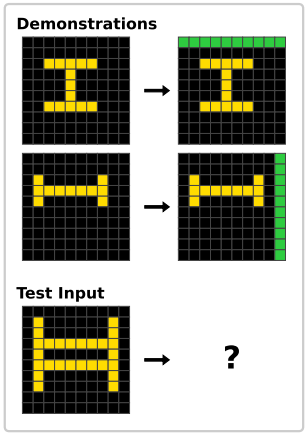
Extract Objects



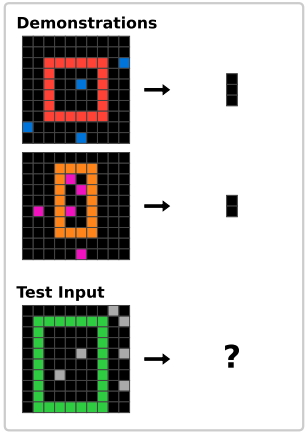
Filled Not Filled



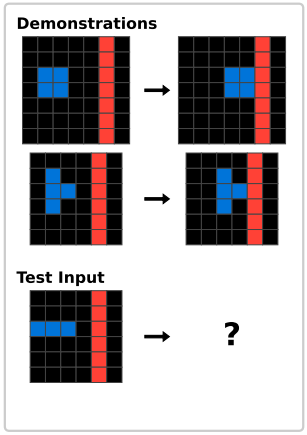
Horizontal Vertical



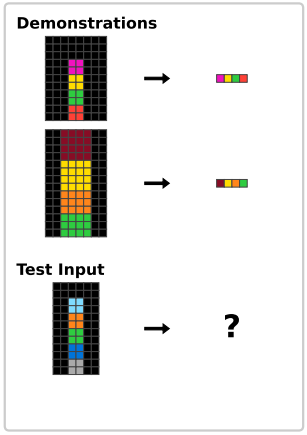
Inside Outside



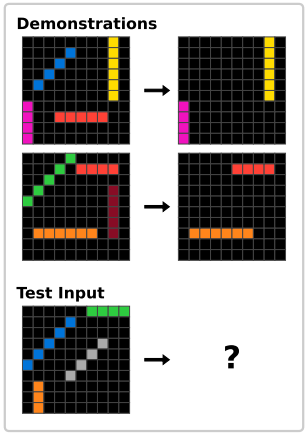
Move To Boundary



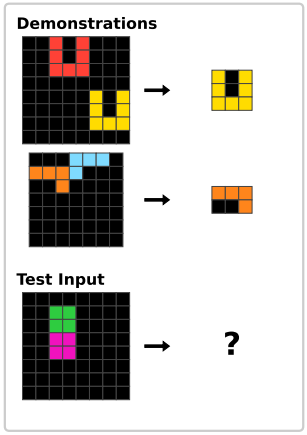
Order



Same Different



Top Bottom 2D



Top Bottom 3D