Abstract Understanding of Core-Knowledge Concepts: Humans vs. LLMs

Alessandro B. Palmarini [1] **Melanie Mitchell** [1]

Abstract

The ability to form and use abstractions in a few-shot manner is a key aspect of human cognition; it is this capacity that enables us to understand and act appropriately in novel situations. In this paper we report on comparisons between humans and GPT-4V on visual tasks designed to systematically assess few-shot abstraction capabilities using core-knowledge concepts related to objectness, object motion, spatial configurations and relationships, and basic numerosity. We test the impact of presenting tasks to GPT-4V using visual, mixed text-visual, and text-only representations. Our findings highlight that GPT-4V, one of today's most advanced multimodal LLMs, still lacks the flexible intelligence possessed by humans to efficiently relate different situations through novel abstractions.

1. Introduction

Humans have a remarkable ability to form abstractions—that is, to find patterns that relate seemingly diverse situations to one another. Abstractions enable us to transfer insights across different contexts and flexibly handle novel situations. This capability is key to forming analogies, developing new concepts, and, more generally, understanding and acting in the world (Hofstadter, 2001).

Various researchers have claimed that sufficiently large pre-trained language models can develop emergent abilities for general abstract pattern recognition (Mirchandani et al., 2023) and analogy-making (Webb et al., 2023). However, the internal mechanisms giving rise to these abilities are not well understood, and other researchers have cast doubt on the claims that these systems actually form humanlike abstractions (Gendron et al., 2023), showing in many cases that while large language models (LLMs) can solve problems involving content similar to that in their training data, they are weak in generalizing outside such problems (McCoy et al., 2023; Razeghi et al., 2022; Wu et al., 2023; Lewis & Mitchell, 2024). Some have interpreted this as evidence that LLMs rely not on generalizable abstract reasoning but on learning complex patterns of associations in their training data and performing "approximate retrieval" of these patterns in new situations (Kambhampati, 2023).

In this paper we use the ConceptARC benchmark (Moskvichev et al., 2023) to compare humans' and GPT-4V's capabilities for discovering and using abstractions. ConceptARC is a set of visual analogy tasks inspired by Chollet's (2019) Abstraction and Reasoning Corpus (ARC). Each task requires discovering a shared abstract concept in a small set of examples to understand and apply that concept to new instances. We compare humans' performance on these tasks with that of GPT-4V. While humans are given these tasks using images only, we test GPT-4V on three types of representations: (1) images only; (2) images plus text; and (3) text only. We find that humans' performance on these tasks far surpasses that of GPT-4V using any of these representations. Our results reinforce the conclusion that today's most powerful LLMs still lack the flexible intelligence possessed by humans to efficiently relate different situations with novel abstractions.

2. The Abstraction and Reasoning Corpus

Chollet (2019) proposes that intelligence lies in the efficiency with which prior information and experience can be transformed into new skills. High skill in specific tasks (e.g., playing chess or recognizing objects in images) is distinct from high intelligence (the process of acquiring such skills). In humans, skill and intelligence are intertwined: general intelligence is always used to create new skills. However, a system with high skill may not have any capacity for generalization, as one can always achieve arbitrarily high levels of skill in specific tasks by leveraging more prior information or experience. Measuring flexible intelligence necessitates a meta-task that evaluates *skill-acquisition efficiency* across various tasks while controlling for priors and experience. Chollet's Abstraction and Reasoning Corpus (ARC; 2019) was designed to evaluate this kind of flexible intelligence across AI systems and humans. ARC consists of 1000 man-

[1] Santa Fe Institute, Santa Fe, NM, USA. Correspondence to: Alessandro B. Palmarini <abp@santafe.edu>, Melanie Mitchell <mm@santafe.edu>.

Proceedings of the 41st International Conference on Machine Learning, Vienna, Austria. PMLR 235, 2024. Copyright 2024 by the author(s).

ually created 'tasks'. Each task contains a small number (typically 2-4) of grid input-output pairs transformed via the same abstract rule, and a test input grid. The task for the solver is to induce the underlying rule and apply it to the test input to generate a transformed output grid. Figure 1 provides an example task from the ARC domain.

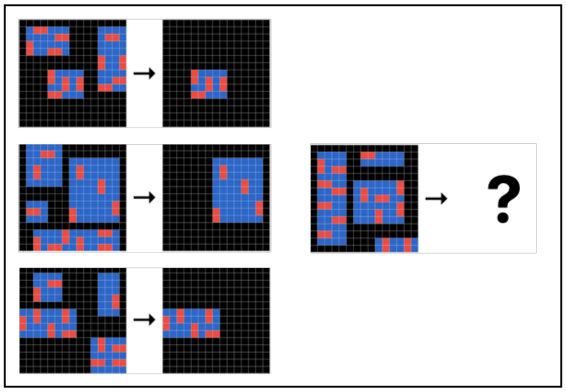


Figure 1. Example task from the ARC domain, showing three input-output pairs illustrating an abstract grid transformation, and a single test input. The solver must generate a new grid by applying the hidden abstract rule to the test input. In this task, the rule is to copy the contiguous colored block that contains the highest number of vertical red lines. (Best viewed in color.)

ARC controls the amount of experience that can be leveraged to solve tasks by providing only a few demonstrations. Additionally, ARC seeks to control the prior knowledge required by explicitly listing the necessary priors. Chollet designed the tasks to only require concepts related to a subset of Spelke's core knowledge systems (Spelke & Kinzler, 2007), such as objectness and object motion, spatial configurations and relationships, and numerosity. For instance, solving the task in Figure 1 involves parsing the grid into contiguous blocks (objectness), identifying smaller horizontal and vertical lines within those blocks (spatial geometry), and generating the block with the most vertical lines (numerosity).

Importantly, the tasks don't rely on language or other learned symbols (e.g. arrows) or concepts (e.g. "cup"). This provides a fair testing ground for comparing human intelligence and artificial intelligence by ensuring humans do not bring extra prior knowledge to the table. Additionally, in the case of evaluating LLMs, avoiding learned symbols helps avoid tasks being solved by pattern matching based on prior training data, which might underlie LLMs' apparent success on language-based abstraction tasks (Lewis & Mitchell, 2024).

Currently, the highest reported accuracy on ARC's hidden test set is 34% (Cole, 2024), achieved by a transformer-based model fine tuned on synthetic and augmented ARC data, though the details have not yet been released. Program synthesis approaches have achieved an accuracy of 31%.

Several groups have tested LLMs on subsets of ARC tasks (Gendron et al., 2023; Mirchandani et al., 2023; Xu et al., 2023; Wang et al., 2023), using different prompting formats, and generally found the best accuracy using straightforward text versions of tasks to be around 10–12%. Limited studies of human performance on subsets of ARC tasks have shown much higher accuracies (e.g., 84% in Johnson et al. 2021).

3. ConceptARC

Moskvichev et al. (2023) noted two problems with the original ARC corpus. First, they claimed, many of the tasks are quite difficult, even for humans, and this difficulty might be a barrier to progress in developing AI systems that can flexibly induce the relevant abstractions. Second, and most important, ARC does not offer *systematic* evaluation of understanding particular core concepts; even if a system can solve an individual ARC task, that does not necessarily mean that the system has a robust understanding of the underlying concepts. For example, if an ARC solver generates the correct output grid for Figure 1, one cannot conclude that the solver can generalize the concepts of "counting" and "greater than"; it might have used a different strategy, like selecting the colored region overlapping the center.

To address these issues, Moskvichev et al. created a new benchmark in the ARC domain, ConceptARC, whose tasks are intentionally designed to be easy for humans. Moreover, its 480 tasks are organized as systematic variations of particular core spatial and semantic concepts, such as *Top and Bottom, Inside and Outside, and Same and Different*. Each concept group contains 30 tasks, each of which instantiates the concept in a different way and with differing degrees of abstraction. Figure 2 provides two example tasks related to the concept of 'Sameness'. Moskvichev et al.'s claim was that high performance over these various instantiations of a given concept indicates a robust understanding of, and ability to reason abstractly about, the underlying concept.

In Moskvichev et al. (2023), 415 human participants were given ConceptARC tasks. Using a grid editor, participants generated the output (answer) grid for each task, enabling automatic correctness checks. The column labeled "Humans" in Table 1 shows the accuracies achieved by humans within each concept group, with an overall accuracy of 91%.

Moskvichev et al. also tested the program-synthesis approach that was the top performer in the original ARC challenge on Kaggle (Kaggle, 2020); this program had an overall accuracy of 52%.[1] Finally, they tested a non-multimodal version of GPT-4, for which the grids in each task were

[1] Notably, this is much higher than performance on the original ARC test set of 21%, indicating that ConceptARC is easier for current algorithms.

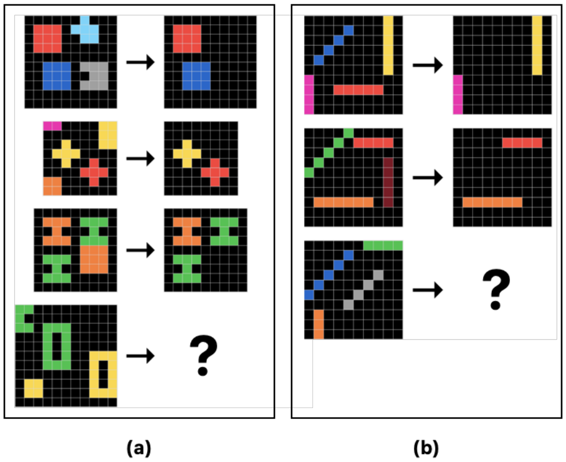


Figure 2. Two example tasks from the ConceptARC concept group *Same and Different*, each instantiating 'sameness' differently: **(a)** sameness between shapes; **(b)** sameness between line orientations.

encoded as arrays of numbers, corresponding to colored cells. For example, the array [0 0 1] [0 1 0] represents a 2×3 grid, where 0s represent black cells and 1s represent blue cells. Using this "text representation" to encode tasks, GPT-4 achieved an overall accuracy of 25%. Follow-up work from Mitchell et al. (2023) retested GPT-4 with a more expressive prompt that included an example solved task (one-shot learning), increasing the overall accuracy to 33%.

A potential criticism of the GPT-4 evaluations is the disparity between humans, who are presented with visual tasks, and LLMs, which receive text-only versions. Mitchell et al. (2023) addressed this by testing GPT-4V, a new multimodal version of GPT-4, using images of the tasks. However, their preliminary investigation involved only extremely simple tasks outside the main ConceptARC dataset.

We extend this investigation by evaluating GPT-4V on the entire ConceptARC dataset (with both image-only and mixed image-text representations), thereby assessing the current capability of multimodal LLMs to create novel abstractions from limited visual instances. Additionally, we use an updated prompt that explicitly lists the core concepts associated with ConceptARC's concept groups, giving GPT-4V explicit information on the possible abstract concepts that are needed for solving these tasks.

4. Experiments Evaluating GPT-4V

4.1. Image-Only Representations

We evaluated GPT-4V on ConceptARC using a prompt where, for each task, each grid was presented as a separate image. We build on work by Mitchell et al. (2023), who conducted preliminary investigations to test GPT-4V's performance using different prompt formats to present a task: an image containing all input-output pairs, a single image for each input-output pair, and individual images for each input and output grid. Only the last format yielded any correct solutions. Furthermore, when presented with an image, GPT-4V struggled to consistently translate the visual grid into a text representation, like the one described in Section 3. Therefore, to mitigate errors involved in mapping the intended output grid to a text representation, we requested only a natural-language description of the output grid.

For each task, we included a list of all 16 core concepts that the tasks are built around in the prompt, explicitly representing the prior knowledge assumed to be necessary for solving the tasks. Additionally, our prompt requests a natural-language description of the underlying rule (abstraction) as well as the output grid. The exact prompt is provided in Appendix A, Figure 3.

We used this prompting method to test GPT-4V (via the Azure OpenAI Service API[2]) on all 480 ConceptARC tasks (30 per each of the 16 concept groups)[3] with the temperature set to zero. Following Chollet's criteria, both humans and GPT-4V were given three attempts to produce the output grid for each task, and if one of the three guesses was correct, the task was considered solved. The accuracies (the fraction of correctly solved tasks within each concept group as well as the overall fraction of solved tasks) are presented in the "GPT-4V (image-only)" column of Table 1.

In some instances, GPT-4V correctly described the output grid by following an incorrect rule (one that did not work for one or more of the task demonstrations). These cases, where the system was "right for the wrong reason", were still counted as correct for measuring accuracy. However, they indicate that in some cases, even when producing correct answers, the system struggles with abstraction.

These accuracies include only responses with correctly described output grids, regardless of the accuracy of the transformation rule description. For each task, we manually classified GPT-4V's (best) description of both the abstract rule and test output grid as correct, partially correct, or incorrect. A partially correct answer contained nearly all details but missed or incorrectly specified a small yet crucial element. If we consider cases where GPT-4V correctly described the transformation rule but only partially or incorrectly described the output grid, its overall accuracy increases from 15% to 22%. A complete breakdown of output and rule classification for GPT-4V's performance is provided in Table 2. Sample tasks and their corresponding natural-language descriptions from GPT-4V, along with our classifications, are

[2] We used the version of GPT-4V (gpt4-vision-turbo) available in May 2024.

[3] The tasks can be downloaded from `https://github.com/victorvikram/ConceptARC`

Concept	Humans	GPT-4V (image-only)	GPT-4V (image & text)	GPT-4V (text-only)
Above and Below	0.90	0.13	0.23	0.23
Center	0.94	0.17	0.30	0.33
Clean Up	0.97	0.17	0.33	0.33
Complete Shape	0.85	0.03	0.27	0.33
Copy	0.94	0.00	0.20	0.40
Count	0.88	0.60	0.30	0.33
Extend To Boundary	0.93	0.10	0.17	0.13
Extract Objects	0.86	0.03	0.07	0.17
Filled and Not Filled	0.96	0.10	0.30	0.30
Horizontal and Vertical	0.91	0.23	0.30	0.27
Inside and Outside	0.91	0.17	0.17	0.20
Move To Boundary	0.91	0.23	0.23	0.17
Order	0.83	0.07	0.30	0.20
Same and Different	0.88	0.20	0.07	0.23
Top and Bottom 2D	0.95	0.13	0.40	0.63
Top and Bottom 3D	0.93	0.03	0.10	0.20
All concepts	**0.91**	**0.15**	**0.23**	**0.28**

Table 1. Accuracies of humans and GPT-4V (provided with image only, image and text, and text only representations of the tasks) on each concept group (30 tasks) and over all concepts (480 tasks) in ConceptARC. The results on humans are from Moskvichev et al. (2023).

provided in Appendix B.

		Rule		
		Correct	Partial	Incorrect
Output	Correct	0.05	0.04	0.06
	Partial	0.06	0.05	0.00
	Incorrect	0.01	0.07	0.66

Table 2. Breakdown of GPT-4V's (image-only) performance on ConceptARC. We chose the best of GPT-4V's three responses on each task. Each cell displays the fraction of those responses that contained a certain output grid classification and a certain abstract rule classification. For example, 5% of GPT-4V's responses correctly described both the abstract rule and the output grid.

4.2. Mixed Image-Text and Text-Only Representations

We conducted a second set of experiments to evaluate whether supplementing grid images with text representations would improve performance. Using the same prompting method as in the image-only experiments, we additionally provided text encodings of each grid (using arrays of numbers representing colors, as described in Section 3) alongside the image. The instructions in the prompt detailed how the colored cells in the images mapped to numbers in the text encoding. Additionally, since the model now receives a text encoding of the test input grid, we requested a text encoding of the output (answer) grid rather than a natural-language description. The exact prompt is provided in Appendix A, Figure 4, and the accuracies are presented in the column labeled "GPT-4V (image & text)" in Table 1.

Overall accuracy increases by 8% when GPT-4V has access to both text representations and images, indicating that additional text representations are beneficial. However, a third set of experiments, which look to isolate the influence of visual representations by using the same prompting method without images (exact prompt in Appendix A, Figure 5), suggests that the images are actually just degrading performance: when moving from image & text representations to text-only, overall accuracy increases to 28% (see column "GPT-4V (text-only)" in Table 1). This holds true for all concept groups except Count, where, surprisingly, image-only representations enable GPT-4V to perform significantly better, nearly doubling the accuracy of the text-only counterparts. This is particularly interesting given that LLMs are generally known to struggle with counting (Hendrycks et al., 2021).

5. Conclusion

Evaluating flexible intelligence requires focusing on skill-acquisition efficiency rather than skill, while controlling for priors and experience. ConceptARC serves as a benchmark adhering to these principles. The generally high accuracies achieved by humans across ConceptARC's concept groups demonstrates the ability to induce the underlying abstractions necessary to relate different variations of the given concepts. In this paper, we assessed the performance of GPT-4V on ConceptARC using image, text, and mixed image-text representations of tasks, while providing all the necessary prior concepts relevant for solving tasks. The relatively poor accuracies achieved in all these experiments demonstrates that, unlike humans, multimodal LLMs currently lack the ability to flexibly discover the necessary abstractions to relate and understand different instances of a visual concept. Surprisingly, despite the inherently visual nature of these tasks, providing LLMs with image inputs instead of purely textual inputs generally degrades their performance.

References

Chollet, F. On the measure of intelligence. *arXiv preprint arXiv:1911.01547*, 2019.

Cole, J. Test-Time Augmentation to solve ARC, 2024. URL https://lab42.global/arc-interview/.

Gendron, G., Bao, Q., Witbrock, M., and Dobbie, G. Large language models are not strong abstract reasoners. *arXiv preprint arXiv:2305.19555*, 2023.

Hendrycks, D., Burns, C., Kadavath, S., Arora, A., Basart, S., Tang, E., Song, D., and Steinhardt, J. Measuring mathematical problem solving with the math dataset. *arXiv preprint arXiv:2103.03874*, 2021.

Hofstadter, D. R. Epilogue: Analogy as the core of cognition. 2001.

Johnson, A., Vong, W. K., Lake, B. M., and Gureckis, T. M. Fast and flexible: Human program induction in abstract reasoning tasks. *arXiv preprint arXiv:2103.05823*, 2021.

Kaggle. Kaggle Abstraction and Reasoning Challenge. https://www.kaggle.com/c/abstraction-and-reasoning-challenge, 2020. Accessed 2023-11-09.

Kambhampati, S. Can LLMs really reason and plan? *Communications of the ACM*, 2023. September 12, 2023.

Lewis, M. and Mitchell, M. Using counterfactual tasks to evaluate the generality of analogical reasoning in large language models. *arXiv preprint arXiv:2402.08955*, 2024.

McCoy, R. T., Yao, S., Friedman, D., Hardy, M., and Griffiths, T. L. Embers of autoregression: Understanding large language models through the problem they are trained to solve. *arXiv preprint arXiv:2309.13638*, 2023.

Mirchandani, S., Xia, F., Florence, P., Ichter, B., Driess, D., Arenas, M. G., Rao, K., Sadigh, D., and Zeng, A. Large language models as general pattern machines. In *Seventh Conference on Robot Learning (CoRL 2023)*, 2023.

Mitchell, M., Palmarini, A. B., and Moskvichev, A. Comparing Humans, GPT-4, and GPT-4V on abstraction and reasoning tasks. *arXiv preprint arXiv:2311.09247*, 2023.

Moskvichev, A., Odouard, V. V., and Mitchell, M. The ConceptARC benchmark: Evaluating understanding and generalization in the ARC domain. *Transactions On Machine Learning Research*, 2023.

Razeghi, Y., Logan IV, R. L., Gardner, M., and Singh, S. Impact of pretraining term frequencies on few-shot numerical reasoning. In *Findings of the Association for Computational Linguistics: EMNLP 2022*, pp. 840–854, 2022.

Spelke, E. S. and Kinzler, K. D. Core knowledge. *Developmental Science*, 10(1):89–96, 2007.

Wang, R., Zelikman, E., Poesia, G., Pu, Y., Haber, N., and Goodman, N. D. Hypothesis search: Inductive reasoning with language models. *arXiv preprint arXiv:2309.05660*, 2023.

Webb, T., Holyoak, K. J., and Lu, H. Emergent analogical reasoning in large language models. *Nature Human Behaviour*, 7(9):1526–1541, 2023.

Wu, Z., Qiu, L., Ross, A., Akyürek, E., Chen, B., Wang, B., Kim, N., Andreas, J., and Kim, Y. Reasoning or reciting? exploring the capabilities and limitations of language models through counterfactual tasks. *arXiv preprint arXiv:2307.02477*, 2023.

Xu, Y., Li, W., Vaezipoor, P., Sanner, S., and Khalil, E. B. LLMs and the Abstraction and Reasoning Corpus: Successes, Failures, and the Importance of Object-based Representations. *arXiv preprint arXiv:2305.18354*, 2023.

A. Prompts Used for Evaluating GPT-4V

```
# PRIMARY INSTRUCTIONS
[System]
Here is a fun abstract reasoning task that you will be a genius at solving.
You will be given a series of demonstrations, labeled "Demonstration 1", "Demonstration 2", and so on. In each
demonstration, there will be two images, one labeled "Input" and the other labeled "Output". Each input and output is an
image of a 2D grid containing colored cells. There is a SINGLE rule that transforms each input grid to the corresponding
output grid. These demonstrations will be followed by a single image labeled "Test Input". Your task is describe the output
 grid that would be generated by applying the same transformation rule to the Test Input.
The rule may involve one or more of the following concepts:
* Identifying elements---that is, points, lines, or other objects---that are above and below other elements
* Identifying elements that are centered in some way
* Removing noise or otherwise cleaning up a grid
* Completing missing parts of objects or shapes
* Counting elements
* Extending elements to grid boundaries or other kinds of boundaries
* Extracting unique objects
* Identifying shapes that are filled-in versus shapes that are not filled-in
* Identifying orientation---horizontal or vertical---of elements
* Identifying elements contained in or outside of other elements
* Moving elements to grid boundaries or other kinds of boundaries
* Arranging elements in a grid in increasing or decreasing order
* Identifying which elements have same or different properties
* Identifying elements that are on top of or on the bottom of other elements

Narrow white lines are used to separate the grid cells. You should treat black cells in each grid as empty cells (
backgrounds).

# QUERY TASK
[User]
Demonstration 1:
Input:
[Image of input grid]
Output:
[Image of output grid]

Demonstration 2:
Input:
[Image of input grid]
Output:
[Image of output grid]

Test Input:
[Image of input grid]

# SECONDARY INSTRUCTIONS
[User]
Please provide a language description of the SINGLE rule that transforms each demonstration input grid into the
corresponding output grid. Please also provide a language description of the grid that would be generated if that rule was
applied to the Test Input grid.

[Assistant]

# IF THE SYSTEM RETURNS WRONG ANSWER (REPEAT UP TO TWO TIMES)
[User]
Your answer does not solve the puzzle. Try again.

[Assistant]
I apologize for my mistake. Here is a better answer:
```

Figure 3. Prompt used for evaluating GPT-4V with image-only representations of ConceptARC tasks. Adapted from Mitchell et al. (2023), the prompt now includes a list providing each of the prior concepts associated with each of ConceptARC's concept groups. Additionally, the underlying abstract rule is explicitly requested along with the test output grid description.

```
# PRIMARY INSTRUCTIONS
[System]
Here is a fun abstract reasoning task that you will be a genius at solving.
You will be given a series of demonstrations, labeled "Demonstration 1", "Demonstration 2", and so on. In each
demonstration, there will be two images, one labeled "Input" and the other labeled "Output". Each input and output is an
image of a 2D grid containing colored cells. In addition to the image of each grid, you will be given a text representation
 of the grid:  Each row of the grid will be represented as a set of numbers corresponding to colors. For example the text
[2 1 2] [1 2 1] represents a grid with two rows and three columns; the first row has (from left to right) cells colored red,
 blue, and red, and the second row has cells colored blue, red, and blue.

The numbers corresponding to each color are as follows:
Black: 0
Blue: 1
Red: 2
Green: 3
Yellow: 4
Gray: 5
Pink: 6
Orange: 7
Light blue: 8
Dark red: 9

There is a SINGLE rule that transforms each input grid to the corresponding output grid.  These demonstrations will be
followed by a single image labeled "Test Input".  Your task is to provide the text representation of the output grid that
would be generated by applying the same transformation rule to the Test Input.
The rule may involve one or more of the following concepts:
* Identifying elements---that is, points, lines, or other objects---that are above and below other elements
* Identifying elements that are centered in some way
* Removing noise or otherwise cleaning up a grid
* Completing missing parts of objects or shapes
* Counting elements
* Extending elements to grid boundaries or other kinds of boundaries
* Extracting unique objects
* Identifying shapes that are filled-in versus shapes that are not filled-in
* Identifying orientation---horizontal or vertical---of elements
* Identifying elements contained in or outside of other elements
* Moving elements to grid boundaries or other kinds of boundaries
* Arranging elements in a grid in increasing or decreasing order
* Identifying which elements have same or different properties
* Identifying elements that are on top of or on the bottom of other elements

Narrow white lines are used to separate the grid cells in images. You should treat black cells (cells with 0 in the text
representations) in each grid as empty cells (backgrounds).

# QUERY TASK
[User]
Demonstration 1:
Input:
[Image of input grid]
[0 0 0 0 0 0] [0 3 0 3 0 0] [0 3 3 0 0 0] [0 0 3 3 3 0] [0 3 3 0 3 0] [0 0 0 0 0 0]
Output:
[Image of output grid]
[0 0 0 0 0 0] [0 3 3 3 3 0] [0 3 3 3 3 0] [0 3 3 3 3 0] [0 3 3 3 3 0] [0 0 0 0 0 0]

Demonstration 2:
Input:
[Image of input grid]
[0 0 0 0 0 0] [0 4 0 0 4 0] [0 4 4 4 0 0] [0 4 0 4 4 0] [0 0 4 0 4 0] [0 0 0 0 0 0]
Output:
[Image of output grid]
[0 0 0 0 0 0] [0 4 4 4 4 0] [0 4 4 4 4 0] [0 4 4 4 4 0] [0 4 4 4 4 0] [0 0 0 0 0 0]

Test Input:
[Image of input grid]
[0 0 0 0 0 0] [0 6 0 0 6 6] [0 6 0 6 0 0] [0 6 0 6 0 0] [0 0 6 0 6 6] [0 0 6 6 6 0]

# SECONDARY INSTRUCTIONS
[User]
Please provide a language description of the SINGLE rule that transforms each demonstration input grid into the
corresponding output grid. Please also provide the text representation for the grid that would be generated if that rule
was applied to the Test Input grid.

[Assistant]

# IF THE SYSTEM RETURNS WRONG ANSWER (REPEAT UP TO TWO TIMES)
[User]
Your answer does not solve the puzzle. Try again.

[Assistant]
I apologize for my mistake. Here is a better answer:
```

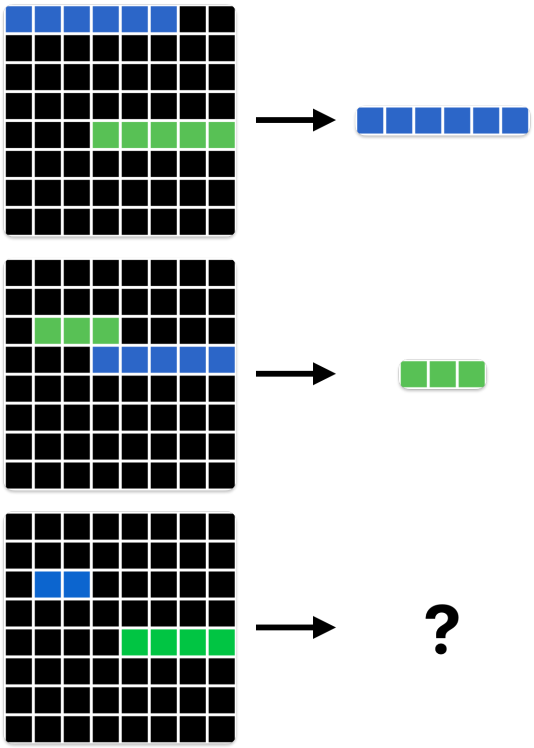
Figure 4. Prompt used for evaluating GPT-4V with mixed image & text representations of ConceptARC tasks. The prompt is a modified version of the image-only prompt in Figure 3, updated to include the array representation of each grid alongside the image and instructions detailing how the colored cells in the image representations map to numbers in the text representations.

```
# PRIMARY INSTRUCTIONS
[System]
Here is a fun abstract reasoning task that you will be a genius at solving.
You will be given a series of demonstrations, labeled "Demonstration 1", "Demonstration 2", and so on. In each
demonstration, there will be two grids, one labeled "Input" and the other labeled "Output". Each input and output is a grid
 of numbers representing a visual grid. Each row of the grid will be represented as a list of numbers. For example the text
 [2 1 2] [1 2 1] represents a grid with two rows and three columns.
There is a SINGLE rule that transforms each input grid to the corresponding output grid.  These demonstrations will be
followed by a single grid labeled "Test Input".  Your task is to provide the text representation of the output grid that
would be generated by applying the same transformation rule to the Test Input.
The rule may involve one or more of the following concepts:
* Identifying elements---that is, points, lines, or other objects---that are above and below other elements
* Identifying elements that are centered in some way
* Removing noise or otherwise cleaning up a grid
* Completing missing parts of objects or shapes
* Counting elements
* Extending elements to grid boundaries or other kinds of boundaries
* Extracting unique objects
* Identifying shapes that are filled-in versus shapes that are not filled-in
* Identifying orientation---horizontal or vertical---of elements
* Identifying elements contained in or outside of other elements
* Moving elements to grid boundaries or other kinds of boundaries
* Arranging elements in a grid in increasing or decreasing order
* Identifying which elements have same or different properties
* Identifying elements that are on top of or on the bottom of other elements

You should treat cells with 0 in each grid as empty cells (backgrounds).

# QUERY TASK
[User]
Demonstration 1:
Input:
[0 0 0 0 0 0] [0 3 0 3 0 0] [0 3 3 0 0 0] [0 0 3 3 3 0] [0 3 3 0 3 0] [0 0 0 0 0 0]
Output:
[0 0 0 0 0 0] [0 3 3 3 3 0] [0 3 3 3 3 0] [0 3 3 3 3 0] [0 3 3 3 3 0] [0 0 0 0 0 0]

Demonstration 2:
Input:
[0 0 0 0 0 0] [0 4 0 0 4 0] [0 4 4 4 0 0] [0 4 0 4 4 0] [0 0 4 0 4 0] [0 0 0 0 0 0]
Output:
[0 0 0 0 0 0] [0 4 4 4 4 0] [0 4 4 4 4 0] [0 4 4 4 4 0] [0 4 4 4 4 0] [0 0 0 0 0 0]

Test Input:
[0 0 0 0 0 0] [0 6 0 0 6 6] [0 6 0 6 0 0] [0 6 0 6 0 0] [0 0 6 0 6 6] [0 0 6 6 6 0]

# SECONDARY INSTRUCTIONS
[User]
Please provide a language description of the SINGLE rule that transforms each demonstration input grid into the
corresponding output grid. Please also provide the text representation for the grid that would be generated if that rule
was applied to the Test Input grid.

[Assistant]

# IF THE SYSTEM RETURNS WRONG ANSWER (REPEAT UP TO TWO TIMES)
[User]
Your answer does not solve the puzzle. Try again.

[Assistant]
I apologize for my mistake. Here is a better answer:
```

Figure 5. Prompt used for evaluating GPT-4V with text-only representations of ConceptARC tasks. The prompt is the same as the mixed image & text prompt given in Figure 4 with all images, and references to images, removed.

B. GPT-4V Sample Responses with Classifications

When evaluating GPT-4V on image-only task representations, we manually classified its descriptions of both the abstract rule and the test output grid as correct, partially correct, or incorrect. Figures 6, 7 and 8 showcase sample tasks exemplifying each classification for both the rule and the output grid descriptions. To maintain brevity, surrounding text from GPT-4V's responses has been omitted. While the displayed examples have matching classifications for both the rule and the output grid, a single response could receive different classifications for each. A complete breakdown of GPT-4V's performance in terms of output and rule classification is provided in Table 2 in Section 4.1.



Correct Rule

"From these demonstrations, the rule appears to be:

- Identify the color of the contiguous cells that form the topmost horizontal line in the grid.

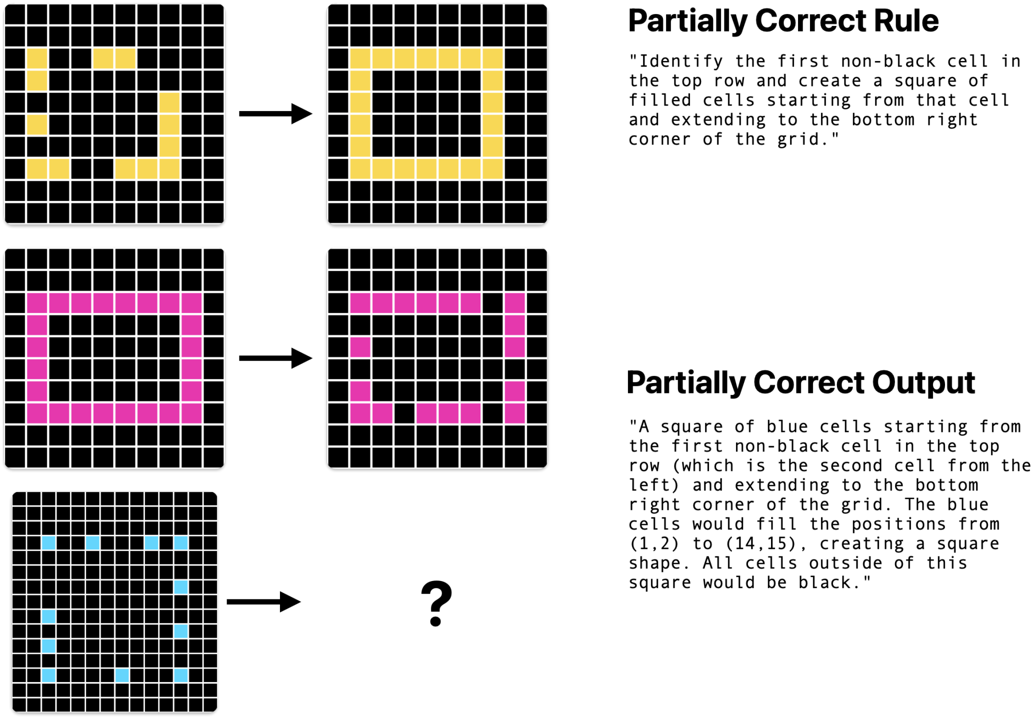
- Count the number of cells in that topmost horizontal line.

- Create an output grid with a single horizontal line of cells in that color, with the number of cells equal to the count from the topmost line in the input grid."

Correct Output

"So, the output grid generated by applying the rule to the Test Input grid would be a horizontal line of two blue cells."

Figure 6. Sample task and GPT-4V (image-only) response with both the abstract rule and output grid descriptions classified as correct.



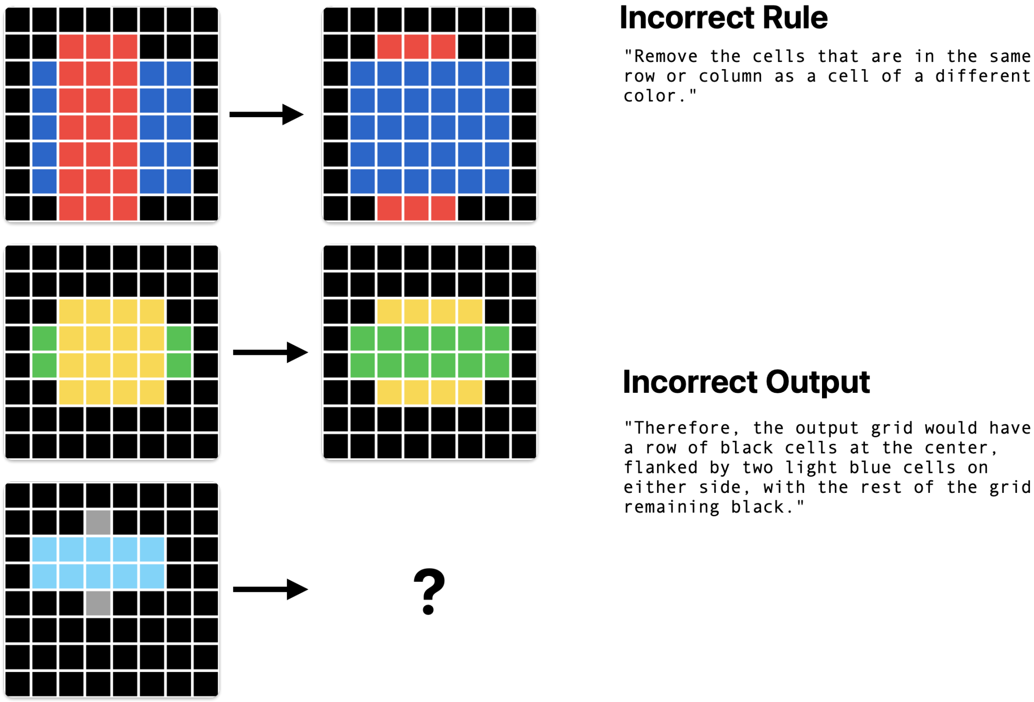
Partially Correct Rule

"Identify the first non-black cell in the top row and create a square of filled cells starting from that cell and extending to the bottom right corner of the grid."

Partially Correct Output

"A square of blue cells starting from the first non-black cell in the top row (which is the second cell from the left) and extending to the bottom right corner of the grid. The blue cells would fill the positions from (1,2) to (14,15), creating a square shape. All cells outside of this square would be black."

Figure 7. Sample task and GPT-4V (image-only) response with both the abstract rule and output grid descriptions classified as partially correct.



Incorrect Rule

"Remove the cells that are in the same row or column as a cell of a different color."

Incorrect Output

"Therefore, the output grid would have a row of black cells at the center, flanked by two light blue cells on either side, with the rest of the grid remaining black."

Figure 8. Sample task and GPT-4V (image-only) response with both the abstract rule and output grid descriptions classified as incorrect.