A 1D-ARC full dataset

Figure 9: **1D ARC: visualizations of selected sample tasks.** For each task type, one pair of input-output images is shown.

Table 5: **Full results of direct-grid and object-based approach on the 1D-ARC dataset.** Each row displays results for one generated task type using different methods. The values correspond to the number of tasks out of 50; higher is better and the top-performer is bolded for each task type.

Task	Direct-grid		Object-based	
	GPT-3.5	GPT-4	GPT-3.5	GPT-4
Move 1	10	33	39	**50**
Move 2	3	13	22	**50**
Move 3	7	12	14	**49**
Move Dynamic	6	11	7	**37**
Move 2 Towards	3	17	17	**50**
Fill	6	33	44	**49**
Padded Fill	3	13	37	**44**
Hollow	2	28	40	**48**
Flip	11	35	20	**50**
Mirror	4	10	6	**13**
Denoise	11	18	48	**48**
Denoise Multicolor	13	30	36	**50**
Pattern Copy	11	18	31	**45**
Pattern Copy Multicolor	16	19	21	**47**
Recolor by Odd Even	13	**16**	15	13
Recolor by Size	2	14	21	**40**
Recolor by Size Comparison	6	10	17	**28**
Scaling	14	44	34	**46**

B 1D-ARC Design Process

1. **Identify ARC task that requires core prior knowledge.**

   See Figure 10(Left).

2. **Determine the solution and the inherent knowledge priors.**

   Solution: Recolor objects based on their sizes.

   Knowledge priors:

   - Objectness (Object cohesion)
   - Numbers and Counting priors (Understanding of object size through counting)

3. **Develop the 1D-ARC task using the identified solution and knowledge priors.**

   See Figure 10(Right).

4. **Reiterate steps 1-3 to create a comprehensive 1D-ARC dataset.**

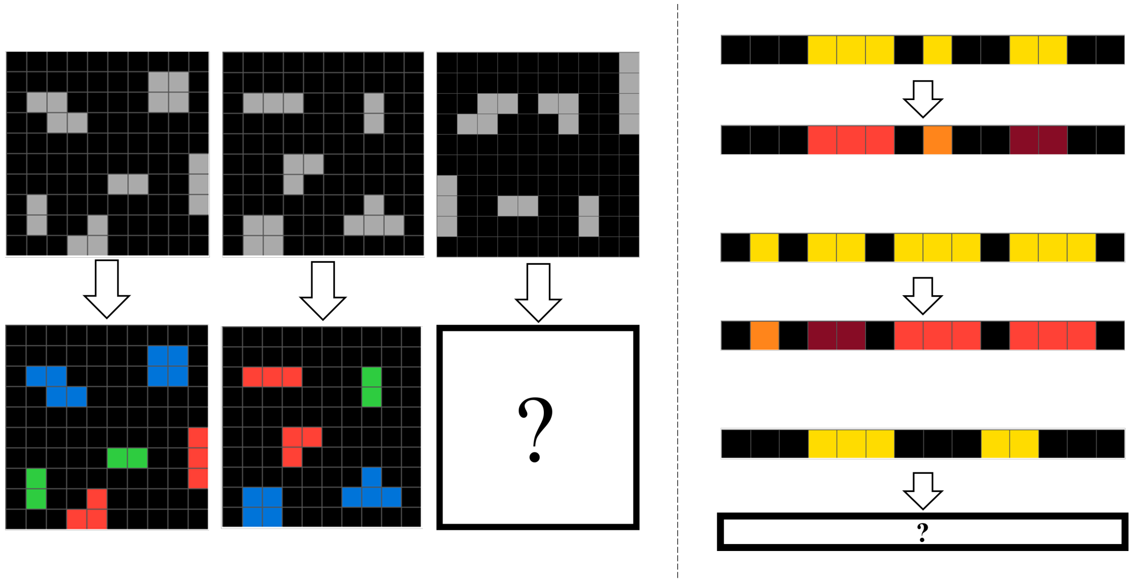

Figure 10: **Example of an ARC task (Left) and a 1D-ARC task adapted from it (Right)**

C Vertical and horizontal dataset

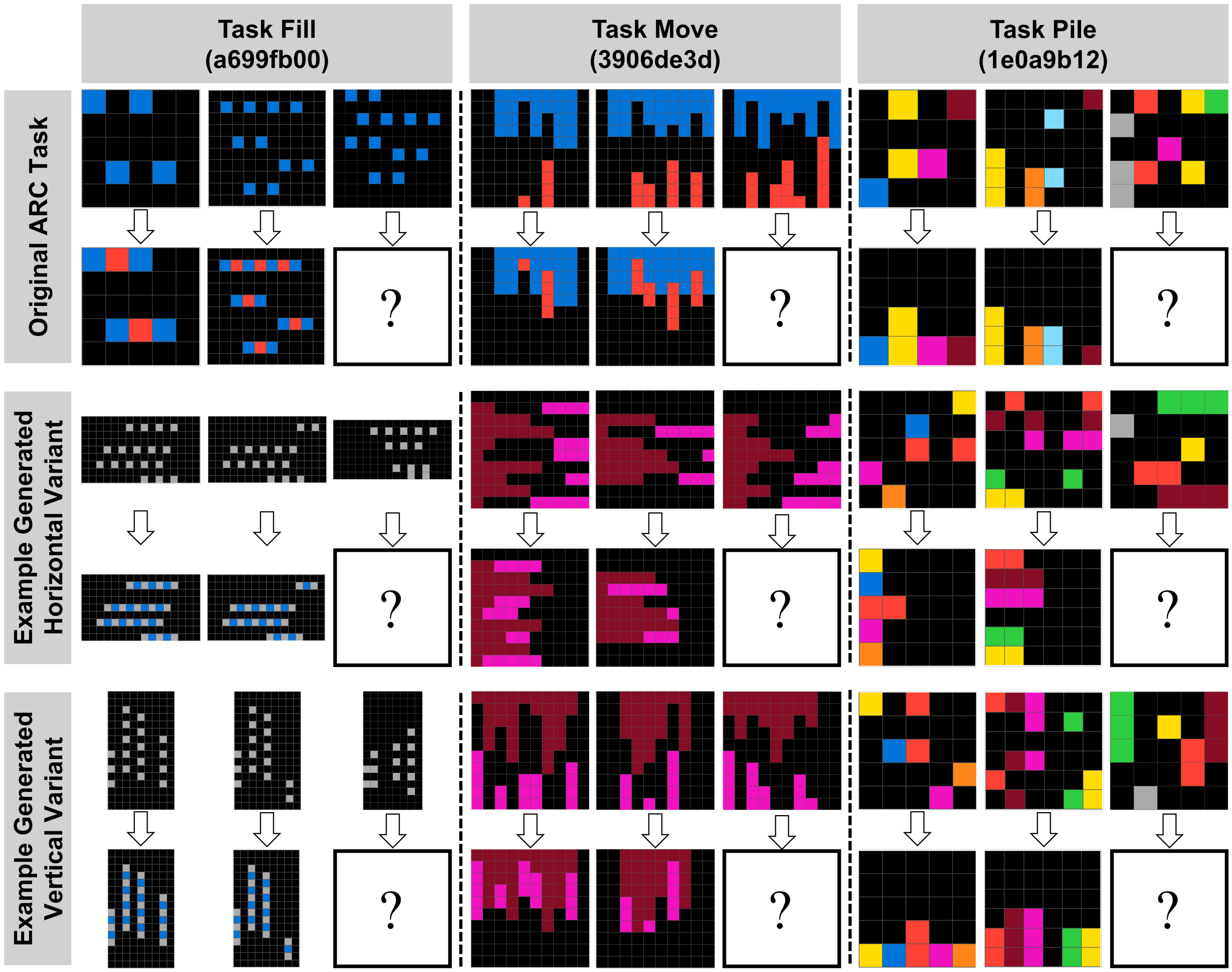

Figure 11: **Visualization for generated vertical and horizontal variants for three ARC tasks.** Each column displays the original ARC task on the top, one generated horizontal variant in the middle, and one generated vertical variant at the bottom.

D GPT reasonings for solved tasks

Table 6: **GPT Reasoning for correctly solved ARC tasks using best performing direct-grid approach.** The correctness of the reasoning provided is determined manually. If a reasoning was not provided, it is reported as N/A.

Task	Reasoning	Correctness
25ff71a9	Transformation applied: 1. Move color objects (blue or red) 1 row down.	✓
aabf363d	N/A	✗

d5d6de2d	Transformation applied: 1. Replace color red object with color green object 2. Remove the outer border of the color green object	✓
dc433765	Transformation applied: 1. Move color green object 1 pixel down and color yellow object 1 pixel up.	✗
b1948b0a	N/A	✗
74dd1130	N/A	✗
c0f76784	Transformation applied: 1. Fill the empty spaces inside the gray objects with the following colors: - Cyan for the top object - Orange for the middle object - Purple for the bottom object	✗
6150a2bd	N/A	✗
c8f0f002	N/A	✗
67385a82	N/A	✗
67a3c6ac	N/A	✗
a5313dff	Transformation applied: 1. Replace the black cells inside the red shape with blue cells.	✓
a699fb00	Transformation applied: 1. Add a red object to the right of every blue object, except when there is already a red object to the right of the blue object.	✗

Table 7: **GPT Reasoning for correctly solved ARC tasks using best performing object-based approach.** The correctness of the reasoning provided is determined manually. If a reasoning was not provided, it is reported as N/A.

Task	Reasoning	Correctness
a79310a0	Transformation applied: 1. Move color 8 object 1 pixel to the right 2. Recolor color 8 object to color 2	✓
25ff71a9	Transformation applied: 1. Move the object 1 pixel to the right.	✓
aabf363d	Transformation applied: 1. Recolor the larger object with the color of the smaller object.	✓
d5d6de2d	Transformation applied: 1. Remove color-2 objects 2. Recolor color-0 objects to color-3	✓
6c434453	Transformation applied: 1. For each color-1 object with size 8, remove the corners to create two color-2 objects with size 5.	✓

dc1df850	Transformation applied: 1. Surround each color-2 object with color-1 objects in a 3x3 square, excluding the center where the color-2 object is located.	✓
aedd82e4	Transformation applied: 1. Recolor all color-2 objects that are not touching the largest color-2 object to color-1.	✓
dc433765	Transformation applied: 1. Move color 3 object 1 pixel towards color 4 object.	✓
6e82a1ae	Transformation applied: 1. Recolor color 5 objects to color 1 if size is 4 2. Recolor color 5 objects to color 2 if size is 3 3. Recolor color 5 objects to color 3 if size is 2	✓
b1948b0a	Transformation applied: 1.Recolor color 6 objects to color 2	✓
4258a5f9	Transformation applied: 1. For each color-5 object, create a color-1 object with 8 pixels surrounding the color-5 object.	✓
ded97339	Transformation applied: 1. Connect color-8 objects with the same color in a straight line.	✓
7f4411dc	Transformation applied: 1.Remove all color objects with size 1.	✓
e9614598	Transformation applied: 1.Find the midpoint between the two color-1 objects. 2.Create a color-3 object of size 5 in a cross shape centered at the midpoint.	✓
ea32f347	Transformation applied: 1. Recolor the largest color-5 object to color-1 2. Recolor the smallest color-5 object to color-2 3. Recolor the remaining color-5 object to color-4	✓
6150a2bd	Transformation applied: 1.Rotate the colors of the objects 90 degrees clockwise.	✗
913fb3ed	Transformation applied: 1. For each object, create a new object with 8 coordinates surrounding the original object. 2. Assign colors to the new objects based on the original object's color: color 2 -> color 1, color 3 -> color 6, color 8 -> color 4.	✓
c8f0f002	Transformation applied: 1.Recolor color 7 objects to color 5	✓
67385a82	Transformation applied: 1.Recolor color 3 objects with size $>$ 1 to color 8.	✓
ce22a75a	Transformation applied: 1. Surround each color-5 object with color-1 objects in a 3x3 square, keeping the color-5 object in the center.	✗

67a3c6ac	Transformation applied: 1.Rotate the colors clockwise within the objects.	✗
a5313dff	Transformation applied: 1.Recolor color 0 objects to color 1	✓
a699fb00	Transformation applied: 1. Recolor color 0 objects to color 2	✓

E Logistic regression analysis

Feature	**Coefficient**	
	Direct-grid	Object-based
Number of colored pixels in test input image	-0.151312	-0.365261
Average number of colored pixels in training input images	0.215891	0.326572
Number of unique colors in test input image	-0.282226	0.346230
Average number of unique colors in training input images	0.192485	-1.186780
Number of pixels changed in test instance	0.110529	0.142800
Average number of pixels changed in training instances	-0.152656	-0.090327
Test input image size	-0.004665	0.001771
Training input images average size	-0.013070	-0.005959
Number of training instances	0.297392	0.158643
Performance Metric	**Score**	
	Direct-grid	Object-based
Precision (unsolved)	0.78	0.83
Precision (solved)	0.44	0.73
Recall (unsolved)	0.86	0.74
Recall (solved)	0.31	0.83

Table 8: **Results of logistic regression analysis.** Top: Comparison of feature coefficients for the best performing direct-grid and object-based approaches, demonstrating the impact of each feature on an ARC task's solvability. Bottom: Precision and recall scores of logistic regression model for solved and unsolved tasks.

F Visualization of GPT solutions on example ARC tasks

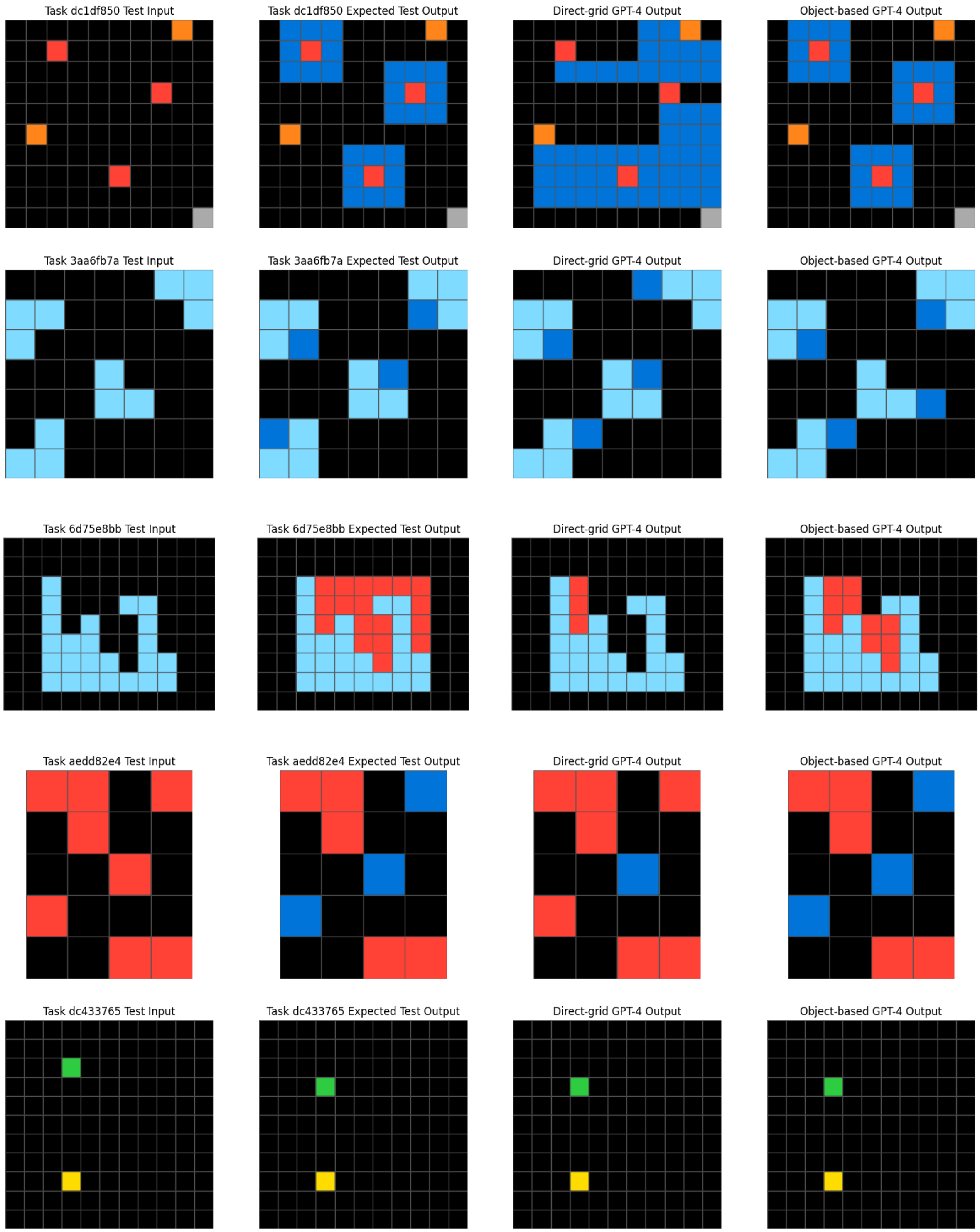

Figure 12: **Visualization of example ARC Tasks.** Each row showcases an ARC task. The first column displays the test input for that task, while the second column shows the expected test output. The third and fourth columns present the predicted outputs using the best-performing direct-grid approach and object-based approach, respectively.

G Experiments with GPT-4V

This section briefly introduces the preliminary experiments conducted for GPT-4V. GPT-4V API is not yet available at the time of our research, therefore, the experiments were conducted by manually prompting ChatGPT-4 with image attachments. First, the prompting method is described and visualized, followed by a display of the results. Examples of GPT's output will also be shown.

G.1 Prompting

The prompting style follows the few-shot approach introduced in Section 2.2. It consists of a "instructions" and "task" sections. The "instructions" section is similar to before where it outlines the nature of an ARC task and the expected behavior of the LLM. The "task" section provides information about the ARC task of interest, however, instead of using textual representations, an image containing the few-shot examples as well as the test input is provided. Figure 13 shows an example the prompts.

G.2 Results

GPT-4V cannot currently produce images as outputs, so it produces texts that describe the output grid. To evaluate the results, we manually examined GPT-V4's outputs and reconstructed the output images based on the descriptions it produced. We found that GPT-4V was only able to correctly describe the output image for 2 tasks out of the 50 tasks we defined earlier, even more so, only a further 3 out of 50 tasks had the correct logic before producing the wrong output. 2 examples of ARC tasks as well as GPT-4V outputs for them can be seen in Figure 14

GPT-4V Input

Instructions

Let's play a puzzle that focus on reasoning and logic. In this puzzle, you will be provided with an image file containing a few demonstrations of how an "Input Grid" gets transformed into a corresponding "Output Grid". In the end of image, you will get a brand new "Test Input Grid", then you must answer the corresponding "Test Output Grid" as an output image and describe the transformations used step by step starting from the "Test Input Grid".

Task

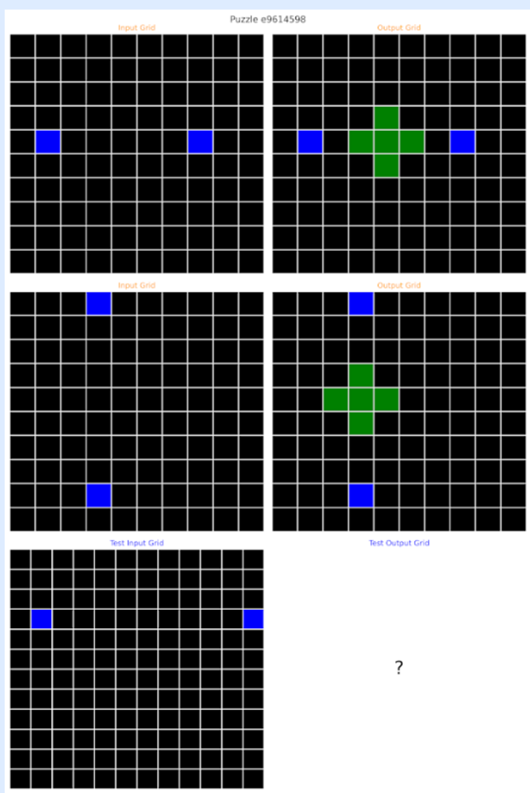

Figure 13: **Visualization for prompting GPT-4V with an ARC task.**

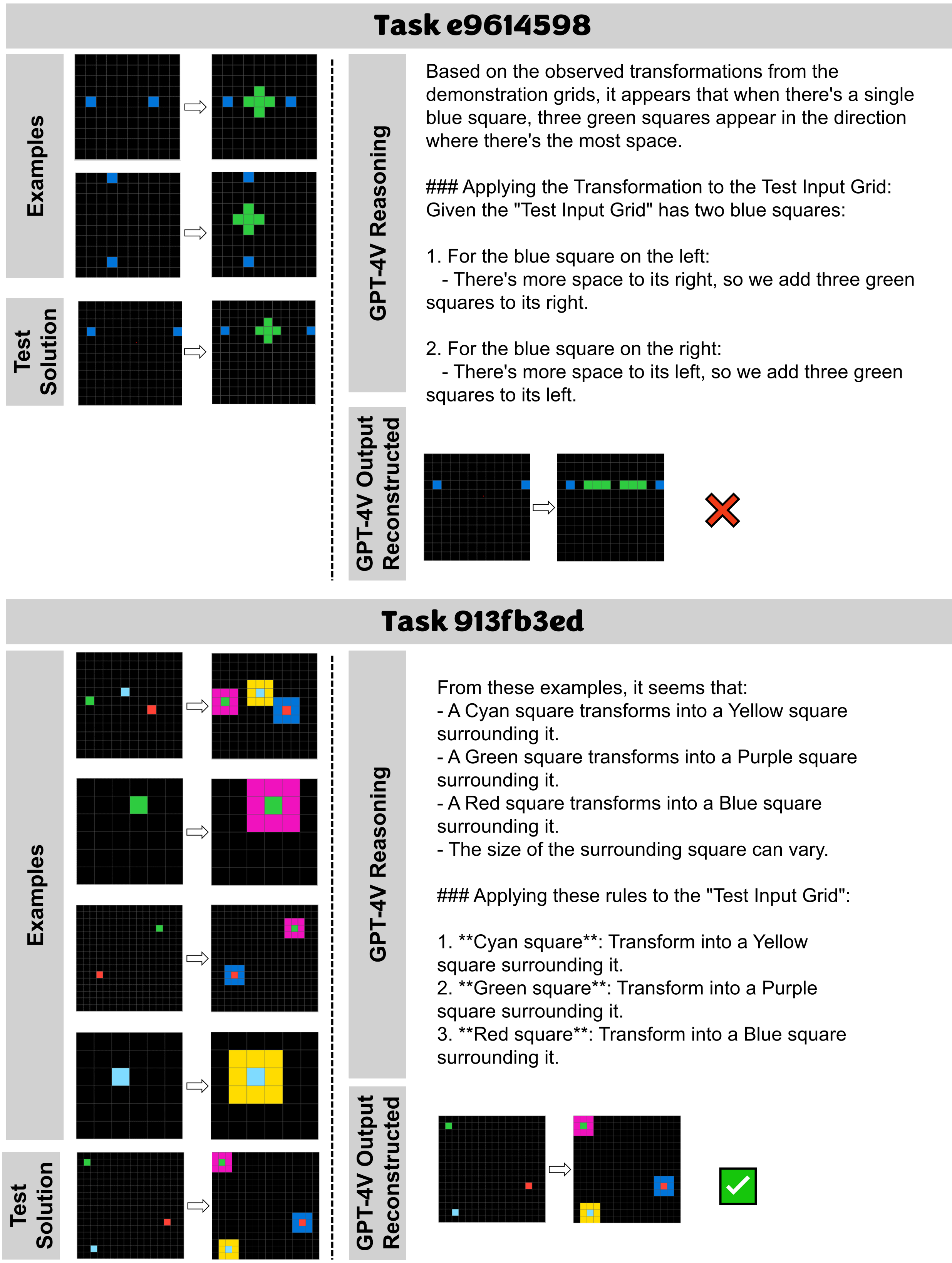

Figure 14: **Visualization of GPT-4V outputs for 2 ARC tasks.**