A Limitations

Since the experiments are compute-intensive, our experiments mainly focus on LLaMA2-7b, but there are many other LLMs trained with different number of parameters, data, or inductive biases. We also only consider one prompt template for each reasoning task, and acknowledge that experimenting with more prompts can provide a more comprehensive evaluation of pretrained LLMs. Last, we use parsers to parse the predictions of models in order to compare with the labels. One alternative approach is the use of other LLMs to compare the predictions with the labels. Some of the above concerns are common challenges for existing evaluation of LLMs. Future research could run evaluations on more LLMs and explore whether the tuning other layers (e.g., output layer, middle layers of transformer blocks) can lead to performance improvement, further proving that LLMs need some amount of task adaptations.

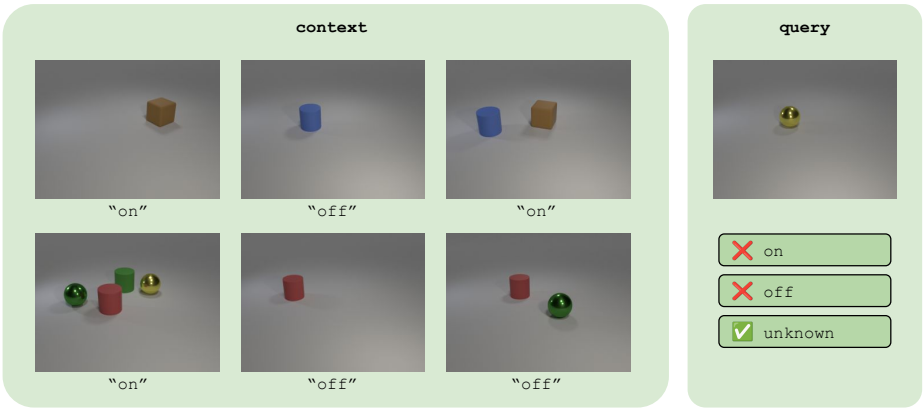
B Additional Figures

We show additional figures to illustrate the reasoning tasks we considered and variants of image inputs.

Pattern	Context	Query
BIG-Bench (BBF) Reverse of the first three elements and append a "4" at the end.	[1, 0, 9, 7, 4, 2, 5, 3, 6, 8] → [9, 0, 1, 4] [3, 8, 4, 6, 1, 5, 7, 0] → [4, 8, 3, 4] [5, 4, 7, 2, 9, 3, 8, 1] → [7, 4, 5, 4] [3, 9, 2, 0, 6, 8, 5, 1, 7] → [2, 9, 3, 4]	[9, 2, 1, 3, 4, 7, 6, 8, 5, 0] → *[1, 2, 9, 4]*
Evals-P If the first character of the input is in the list, then return "foo"; Otherwise, return "bar".	f, [o, z, a, n, g, e, j, f, i, c, l, u, b] → foo l, [v, u, f, b, m, y, j, h, n, c, d, a, p] → bar p, [c, e, s, h, q, o, a, t, k, d, n, l, z] → bar p, [c, h, m, z, d, v, k, l, j, e, x, p, n] → foo	u, [d, a, x, i, h, v, e, z, r, c, n, y, o] → *bar*
Evals-S Identify the correspondence between each digit and word.	13, 17, 1, 6 → Brown,White,Purple,Blue 1, 9, 6, 11 → Purple,Brown,Blue,White 13, 2, 17, 10 → Brown,Purple,White,Blue	5, 9, 2, 11 → *Blue,Brown,Purple,White*
Pointer-Value Retrieval (PVR) The first element indicates the index of the expected output in the remaining list (i.e., ignore the first element).	[5, 7, 4, 1, 8, 9, 8, 1, 9, 8, 4] → 8 [4, 0, 0, 7, 0, 1, 0, 5, 3, 0, 0] → 1 [0, 2, 8, 2, 5, 9, 4, 3, 8, 5, 4] → 2 [3, 3, 2, 6, 5, 7, 4, 6, 7, 4, 8] → 5	[3, 4, 9, 7, 1, 8, 7, 1, 0, 3, 5] → *1*
ACRE Determine whether the query object will activate the light.	A cyan cylinder in rubber is visible. The light is on. A gray cube in rubber is visible. The light is off. A cyan cylinder in rubber is visible. A gray cube in rubber is visible. The light is on. A blue cube in metal is visible. The light is off. A gray cylinder in rubber is visible. A gray cube in metal is visible. The light is off. A red sphere in metal is visible. A yellow cube in rubber is visible. The light is on.	A red sphere in metal is visible. The light is *undetermined*.
RAVEN Find and infer the last pattern from the given context.	1. On an image, a large lime square rotated at 180 degrees. 2. On an image, a medium lime square rotated at 180 degrees. 3. On an image, a huge lime square rotated at 180 degrees. 4. On an image, a huge yellow circle rotated at 0 degrees. 5. On an image, a large yellow circle rotated at 0 degrees. 6. On an image, a medium yellow circle rotated at 0 degrees. 7. On an image, a medium white hexagon rotated at-90 degrees. 8. On an image, a huge white hexagon rotated at-90 degrees.	The pattern that logically follows is: 9. *On an image, a large white hexagon rotated at-90 degrees.*

Figure B.1: Data examples of abstract reasoning tasks.

(a) ACRE



(b) MEWL

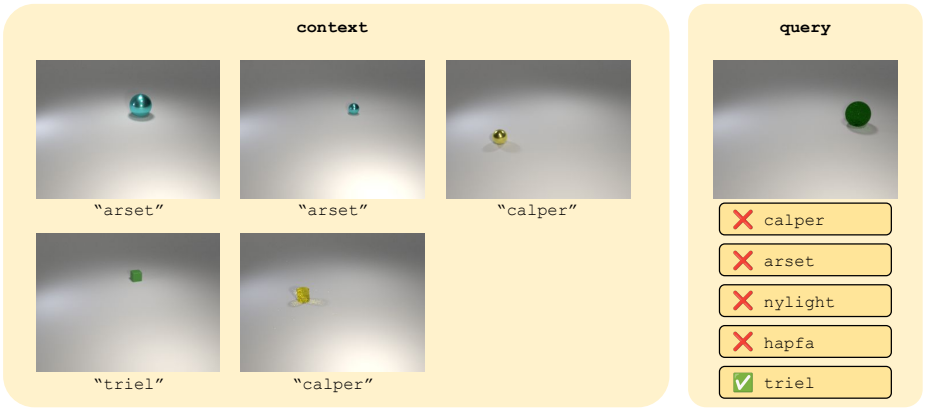


Figure B.2: Data examples of abstract visual reasoning tasks.

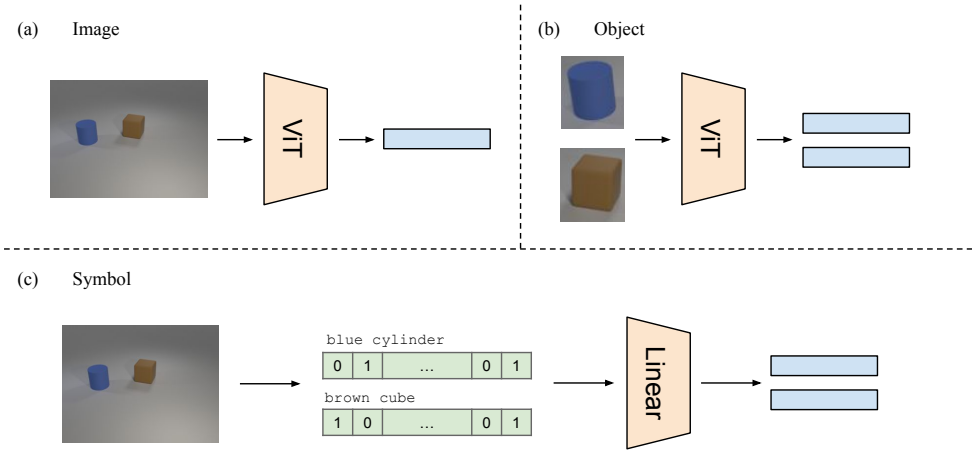


Figure B.3: Examples of variants of image inputs. (a) An image is directly fed into a ViT and obtain an image representation. (b) Each object crop is fed into a ViT and obtain an object representation. (c) Each object is parsed into a multi-hot vector, and a linear layer will output a corresponding object representation.

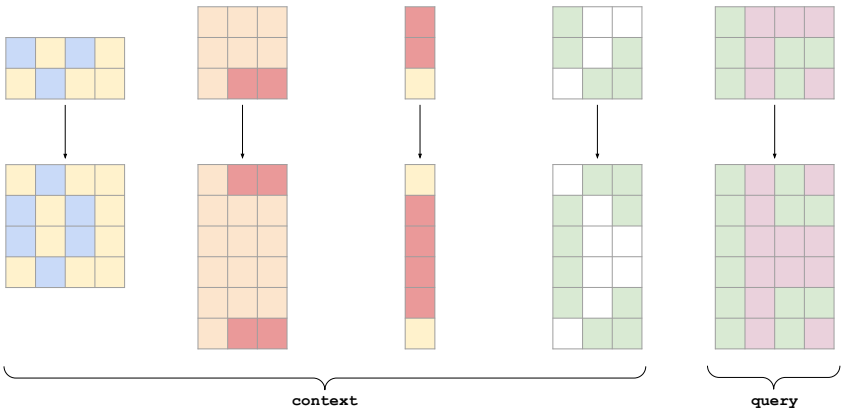


Figure B.4: Example of ARC dataset. There are 4 context examples and 1 query, where each example has an input grid (top) and an output grid (bottom). Each grid is represented as an integer array, where each integer refers to a color. In this example, the task is to generate the symmetry of the input grid and stack the symmetry on top of the original input.

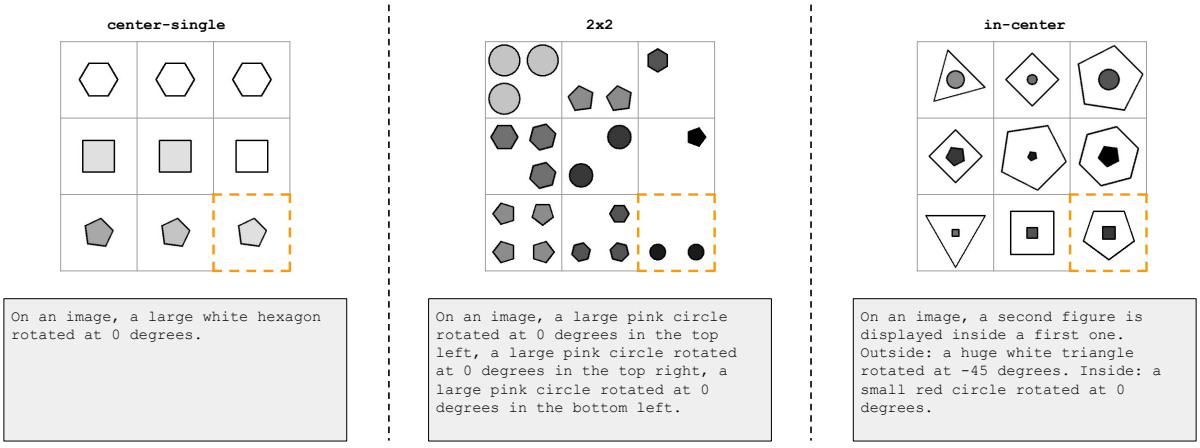


Figure B.5: Examples of RAVEN^T tasks used in generalizability and data efficiency analysis. Top shows the data example, and bottom shows the language description of the first frame in each example. The task is to fill in the ninth pattern (highlighted in orange) given the eight context frames. We focus on three tasks: `center-single`, `2x2` and `in-center`. `center-single` is the simplest task, since there is always only one object in each frame. `2x2` and `in-center` consider more than one objects in the frames and also involve different object alignments.