What is an "Abstract Reasoner"? Revisiting Experiments and Arguments about Large Language Models

Tian Yun
Brown University
tian_yun@brown.edu

Chen Sun
Brown University
chensun@brown.edu

Ellie Pavlick
Brown University
ellie_pavlick@brown.edu

Abstract

Recent work has argued that large language models (LLMs) are not "abstract reasoners", citing their poor zero-shot performance on a variety of challenging tasks as evidence. We revisit these experiments in order to add nuance to the claim. First, we show that while LLMs indeed perform poorly in a zero-shot setting, even tuning a small subset of parameters for input encoding can enable near-perfect performance. However, we also show that this finetuning does not necessarily transfer across datasets. We take this collection of empirical results as an invitation to (re-)open the discussion of what it means to be an "abstract reasoner", and why it matters whether LLMs fit the bill.[1]

1 Introduction

The question of whether large language models (LLMs) are "abstract reasoners" has been the frequent subject of recent work, both directly (Hu et al., 2023; Webb et al., 2023; Gendron et al., 2024; Musker et al., 2024) and indirectly (Chollet, 2019; Mitchell et al., 2023; Moskvichev et al., 2023). The answer to this question feels weighty. LLMs currently dominate modern approaches to AI, and abstract reasoning is arguably the linchpin of general and flexible intelligence (Gentner et al., 2001; Han et al., 2024; Mitchell, 2021). If LLMs are not abstract reasoners, it follows that fundamental changes are needed in how AI is developed.

The challenge with this question is that there is little consensus on what it means to be an "abstract reasoner", and what evidence would convincingly demonstrate that an LLM, or any model, is or is not one. Most recently, it has been argued that LLMs are not abstract reasoners on the basis of their poor performance when tested out-of-the-box on adapted visual, analogical, and quantitative reasoning tasks (Figure B.1 for examples) that require models to infer and generalize patterns from a limited number of observations (Gendron et al., 2024; Mitchell et al., 2023; Stevenson et al., 2024). In this work, we revisit this experimental setup. We replicate the results of earlier studies, but add additional experiments which demonstrate the need for more nuance before drawing strong conclusions.

Specifically, we follow the experimental design from Gendron et al. (2024), and replicate their finding that off-the-shelf pretrained LLMs perform badly across a range of challenging reasoning tasks (§4). However, we find that optimizing just the embedding layer for the task (leaving the transformer blocks frozen) all but eliminates the problem, allowing the model to perform comparably to finetuning the entire model, and sometimes even solve the task perfectly (§5). This result extends beyond simple embeddings and, in fact, a frozen pre-trained LLM can perform well on visual reasoning tasks as long as the visual encoder is fine-tuned on in-domain task data (§6).

Together, these results paint a more subtle picture of LLMs: much of their representations and inferential capabilities appear to be transferable across very diverse tasks, but non-trivial effort is required on the input side for each new task in order to harness these capabilities. In light of this, we (re-)open the larger discussion which is simultaneously empirical and philosophical (§7): What does it mean to be an abstract reasoner, and why do we care whether LLMs fit the bill?

2 Related Work

2.1 Analogical Reasoning

Prior work has studied the question of abstract reasoning of LLMs via analogical reasoning, such as matrix reasoning (Webb et al., 2023), letter-string analogies (Mitchell, 2021; Hofstadter et al., 1995) and pointer-value retrieval (Zhang et al., 2021b). These analogical reasoning benchmarks require a

[1]Code and resources are available at: https://github.com/tttyuntian/abstract_reasoner_llm

Proceedings of the 29th Conference on Computational Natural Language Learning, pages 156–168

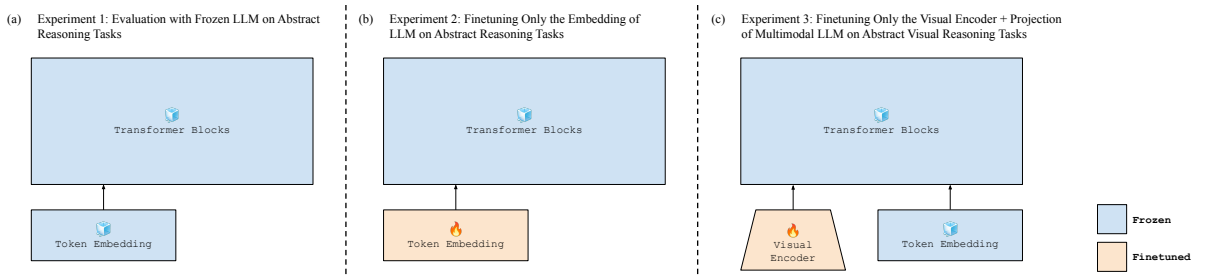


Figure 1: Illustration of our experimental settings. In Setting (a), we freeze the whole LLM and run evaluations. This is treated as language baseline when image captions are inputs on abstract visual reasoning tasks. In Settings (b) and (c), we freeze the pretrained transformer blocks and finetune only the input layers (i.e., token embedding layer and visual encoder). In Setting (c), we freeze the token embedding layer to study the impact of tuning the visual encoder in a controlled setting. Note that the inputs are pure language in Settings (a) and (b), while the inputs are language prompts with image representations in Setting (c).

model to infer the patterns from a limited number of observations and apply the discovered patterns to the new queries.

Despite the impressive performance of LLMs, there is yet no consensus on whether LLMs are strong analogical reasoners. Some studies show evidence suggesting that LLMs can even surpass the human baseline on analogical reasoning tasks (Hu et al., 2023; Webb et al., 2023), while the others show that LLMs achieve very limited performance on a set of analogical reasoning benchmarks (Gendron et al., 2024) or they are not robust to counterfactual examples or irrelevant information (Lewis and Mitchell, 2024; Musker et al., 2024). We use similar tasks and models as the prior work, but incorporate additional tasks and a wider range of finetuning experiments in order to situate the results within a larger discussion about abstract reasoning.

2.2 Visual Analogical Reasoning

Analogical reasoning can go beyond symbols and words and involve visual input, such as in ARC (Chollet, 2019), ACRE (Zhang et al., 2021a), RAVEN (Zhang et al., 2019; Hofstadter et al., 1995) and MEWL (Jiang et al., 2023). Recent approaches on visual analogical reasoning can be categorized into neuro-symbolic methods (Mao et al., 2019; Hudson and Manning, 2019), or neural networks with implicit representations (Ding et al., 2021; Sun et al., 2024; Bhattacharyya et al., 2023). Both approaches roughly follow the same outline of the perception stage and the reasoning stage. The perception stage usually relies on task-specific visual encoders, such as symbolic object encoders (Zhang et al., 2021a), object detectors (Ding et al., 2021), or on task-specific training strategies for these visual encoders (Sun et al., 2024; Bhattacharyya et al., 2023). The reasoning stage introduces inductive biases by developing task-specific reasoning modules (Hu et al., 2021b; Benny et al., 2021). In this work, we investigate if the transformer blocks of a pretrained LLM can be used as a reasoner for different visual analogical reasoning tasks.

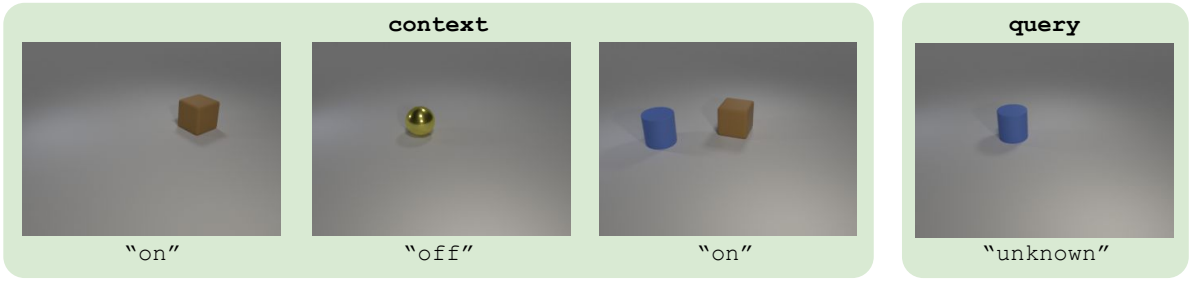
2.3 Multimodal Large Language Models

Prior work shows that transformer blocks pretrained on natural language can be transferred to non-language sequence modeling problems by optimizing new input and output layers (Lu et al., 2022). With the rise of LLMs, recent work freezes pretrained vision models and pretrained LLMs, and only learns a mapping to project visual representations to language latent space in order to perform on multimodal tasks (Merullo et al., 2023; Liu et al., 2023; Li et al., 2023; Liu et al., 2024). Tong et al. (2025) investigates the impact of vision-only models in multimodal LLMs and reaches impressive performance on downstream tasks. Our work is similar to these models, but connects it to a larger, more philosophical debate about the meaning of "abstract reasoning".

3 Datasets

3.1 Reasoning Tasks from Gendron et al.

We follow the evaluation benchmark used by Gendron et al. (2024) to quantitatively measure the so-called "abstract reasoning" capabilities of language models. This benchmark contains seven tasks, each of which evaluates the ability of a model to infer patterns from a limited number of examples. These seven tasks can be divided into two categories: open question answering (OPQA) and multiple-



(a) Textual reasoning task

You are a helpful assistant that determines whether the light will be activated by the objects. Some objects can activate the light. The other objects cannot activate the light. There are three possible light states: on, off, and unknown.
Input: there is a brown cube. Light: on.
Input: there is a yellow sphere. Light: off.
Input: there is a brown cube and a blue cylinder. Light: on.
Input: there is a blue cylinder. Light: unknown.

(b) Visual reasoning task

You are a helpful assistant that determines whether the light will be activated by the objects. Some objects can activate the light. The other objects cannot activate the light. There are three possible light states: on, off, and unknown.
Input: `[context_image_1_representation]` Light: on.
Input: `[context_image_2_representation]` Light: off.
Input: `[context_image_3_representation]` Light: on.
Input: `[query_image_1_representation]` Light: unknown.

Figure 2: Illustration of the use of language models for text-based and image-based versions of ACRE. Each data example will be formulated into a prompt for an LLM to make a prediction for the query. In textual reasoning task, each context frame is represented by a frame caption. In visual reasoning tasks, each context frame is represented by an encoded frame representation.

choice question answering (MCQA). OPQA tasks require a model to generate the correct answer, while MPQA tasks require a model to select the correct answer from the given set of answer candidates. OPQA tasks include Abstract Reasoning Challenge (ARC) (Chollet, 2019), BIG-Bench dataset (BBF) (Rule, 2020; Srivastava et al., 2022), Evals-P (Achiam et al., 2023), and Pointer-Value Retrieval (PVR) (Zhang et al., 2021b). MCQA tasks include ACRE^T (Zhang et al., 2021a), RAVEN^T [2](Zhang et al., 2019), and Evals-P (Achiam et al., 2023). For ACRE^T and RAVEN^T, we also consider ACRE^T-Symb and RAVEN^T-Symb, where the panel descriptions are converted into symbols (e.g., using integers to represent different objects).

3.2 Additional (Visual) Reasoning Tasks

In addition to the models and tasks considered by Gendron et al. (2024), we additionally consider how well LLM representations transfer fo the multimodal language model framework (MLLM). To support these experiments, we consider two visual reasoning tasks: ACRE (Zhang et al., 2021a) and MEWL (Jiang et al., 2023). In ACRE, given the 5 context frames and 1 query frame, a model needs to predict the activation status of Blicket detector in the query frame, which can be *on*, *off*, or *unknown*. In MEWL, given 6 context frames and 1 query frame, a model needs to understand the meaning of the novel words and select the correct utterance out of 5 options for the query frame.

4 Frozen Pretrained LLMs

We first seek to replicate Gendron et al. (2024)'s finding that frozen pretrained LLMs achieve low performance across a large suite of reasoning tasks. We reproduce these evaluations on LLaMA2 with 7 billion parameters (Touvron et al., 2023). Table 1 shows the results on OPQA and MCQA tasks. We observe that even though there are small gaps

[2]ACRE^T and RAVEN^T are text-based version of the original tasks.

	OPQA						MCQA				
	ARC	BBF	Evals-S	PVR	RAVEN^T	RAVEN^T-Symb	ACRE^T-Text	ACRE^T-Symb	Evals-P	RAVEN^T	RAVEN^T-Symb
Random	-	-	-	-	-	-	33.3	33.3	50.0	12.5	12.5
LLaMA2-7b-chat (NZ)	0.5	10.8	0.0	0.0	0.0	0.1	1.4	0.3	50.0	2.6	14.9
LLaMA2-7b-chat (Ours)	1.0	26.4	0.0	21.8	0.0	1.0	26.4	38.1	52.0	12.9	11.4

Table 1: Performance of frozen pretrained LLMs on open question answering (OPQA) and multiple-choice question answering (MCQA) benchmarks. We show the results `LLaMA2-7b-chat(NZ)` reported in Gendron et al. (2024) and our reproduced results `(Ours)` following the evaluation from Gendron et al. (2024).

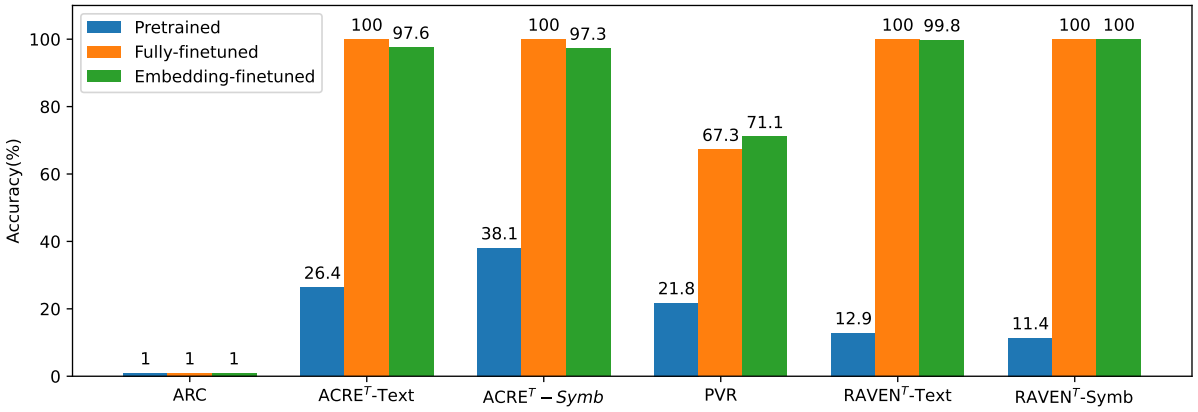


Figure 3: Performance of finetuned LLMs on OPQA (ARC and PVR) and MCQA (ACRE^T and RAVEN^T) benchmarks. LLMs with finetuned embedding layer perform significantly better than their pretrained counterparts, and perform on par with or even surpass the fully finetuned LLMs with LORA. Note that ACRE^T and RAVEN^T are text-based version of original datasets, which may make the tasks easier to solve.

between the original results and the reproduced results, the performance of the pretrained LLMs are still low. Even when the answer candidates are provided in MCQA tasks, the models mostly perform as poor as random baselines (e.g., 33.3% on ACRE and 12.5% on RAVEN). We observe significant gaps between original results and ours on BBF and PVR, and attribute them to the choice of parser used to process the model's predictions.

Overall, our results are, if anything, stronger than what has been previously reported in this evaluation setting. But even so, it is hard to argue that these numbers represent "strong" performance. We thus agree with Gendron et al. (2024) that these results indicate poor transfer ability. What requires additional investigation, however, is whether this poor transfer is interpretable as a lack of abstract reasoning ability.

5 Finetuned Embedding Layers

Given that pretrained LLMs perform poorly off-the-shelf, it is natural to ask whether they can be adapted to these task, and if so, just how much adaptation is necessary. We explore two ways to finetune the LLMs: (1) finetuning all layers with low-rank adaptation (LoRA) (Hu et al., 2021a); (2) finetunning only the embedding layer of the LLMs. LoRA finetuning has become a standard way of adapting a model to a task and represents an upper bound on how well the model could be made to perform the task under the most permissive conditions. In contrast, finetuning just the embedding layer represents a conceptually different type of transfer with respect to the question of this paper. Namely, finetuning just the embeddings is analogous to changing just the input to the system–e.g., ensuring the input is in the format the system expects–but leaving the system itself unchanged (see additional discussion and qualifications about this analogy in §7).

We finetune the embedding layer for 50 epochs using AdamW optimizer (Loshchilov and Hutter, 2019) with early stopping based on the validation set. Following Gendron et al. (2024), we conduct experiments on 2 OPQA tasks (ARC, PVR) and 2 MCQA tasks (ACRE^T, RAVEN^T-mcqa[3].).

Figure 3 shows the results of finetuned LLaMA2. We observe that LoRA-finetuned models perform significantly better than their pretrained counter-

[3]We are aware of the defects of RAVEN, and we use the original RAVEN since it was previously used by Gendron et al. (2024)

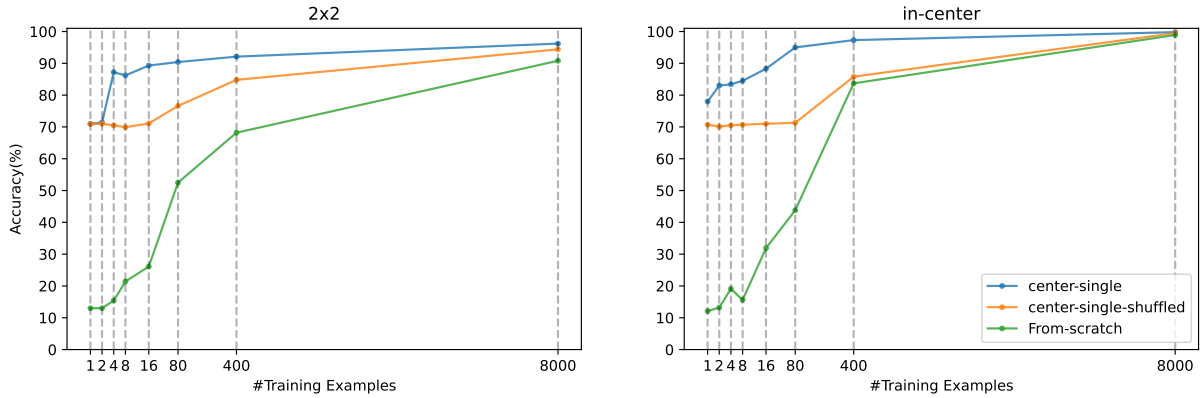


Figure 4: Data efficiency analyses on LLaMA2-7b with token embedding layer finetuned on `center-single` or `center-single-shuffled` and further finetuned on `2x2` and `in-center` tasks in RAVEN with limited amount of data. Y-axis (`#Training Examples`) represents the absolute number of examples used for finetuning. `From-scratch` means the token embedding of a pretrained LLaMA2-7b is directly finetuned on `2x2` and `in-center` tasks. Given that there are 8000 training examples in total, we observe that after finetuning on `center-single`, the model becomes significantly more data efficient. By comparing `center-single` and `center-single-shuffled` lines, we observe that data efficiency of the model mainly comes from the occurrences of task-relevant tokens, rather than the reasoning logic of the tasks.

parts, and can even solve ACRE^T and RAVEN^T perfectly. Moreover, LLaMA2 with a finetuned embedding layer can perform on par with the LoRA-finetuned LLaMA2[4].

Generalizability and Data Efficiency. We conduct experiments on RAVEN^T to further look into two properties of the finetuned input layers: generalizability and data efficiency. An ideal abstract reasoner is expected to generalize to novel tasks with limited amount of observations.

We take LLaMA2-7b's token embedding layer finetuned on one task variant (namely, the `center-single` task) and further finetune this layer with varying amount of training examples for 500 steps on two different task variants (`2x2` and `in-center`), both of which require reasoning over more attributes (e.g., different object alignments). Figure B.5 shows examples of these tasks. We consider three settings: (1) "`center-single`", where the token embedding has been finetuned on the original `center-single` task; (2) "`center-single-shuffled`", where the token embedding has been finetuned on `center-single` task with randomly shuffled labels. This setting preserves the visual features, but breaks the logical "reasoning" structure of the task, and thus serves as a test of how much of the positive transfer is due to low-level visual cues vs. higher-level more abstract features; (3) "`from-scratch`", where the token embedding of a pretrained LLaMA2-7b is directly finetuned on `2x2` and `in-center` tasks. We use this to study the impact of finetuning on `center-single` task.

Figure 4 shows the results. LLaMA2-7b with token embeddings finetuned just on 80 examples can perform competitively aganist LLaMA2-7b directly finetuned on full dataset (8k examples) of the tasks. The fairly small gap between the `center-single` and `center-single-shuffled` lines suggests that the positive transfer is primarily explained by the lower-level visual features rather than the reasoning logic of the tasks.

6 Visual Encoder Trained from Scratch

Prior work has shown that transformer blocks pretrained on natural language can be tranferred to non-language sequence modeling tasks, such as image recognition and protein fold prediction (Lu et al., 2022). Given the surprising effectiveness of finetuning just the embedding layer of LLaMA2 on text-only abstract reasoning tasks, we hypothesize that the frozen transformer blocks of a pretrained LLM will perform well on abstract visual reasoning tasks if the visual encoder is tuned for the task. That is, we follow the multimodal LLM framework (MLLM) which consists of a visual backbone, a language backbone, and a linear projection layer which maps visual representations to language latent space. We keep the transformer blocks and

[4]We attribute the low performance of ARC to its complexity and the length of each data sequence (excluding the expected answer), where 75% of data has >2000 tokens.

	Method	I.I.D.	Compositional	Systematic
Language Baseline	LLaMA2-7b	26.4	26.1	29.9
	GPT-4	66.4	66.4	64.0
	GPT-4-Turbo	69.7	69.9	67.4
Existing Approaches	NS-OPT	66.3	69.0	67.4
	ALOE	-	91.8	93.9
	IV-CL	93.0	93.2	92.6
	LRR	-	**98.2**	**99.2**
	LLaVA-NeXT-Mistral-7B	38.4	36.9	36.9
	GPT-4o	62.6	61.5	61.7
Ours	LLaMA2-7b-Object	**95.5**	97.5	86.5

(a) ACRE

	Method	shape	color	material	object	composite	relation	bootstrap	number	pragmatic	Avg.
Language baselines	LLaMA2-7b	49.7	61.2	52.5	73.8	35.2	19.2	29.5	21.8	22.2	40.6
	BERT*	94.8	98.8	97.5	19.5	**97.8**	**22.2**	62.2	21.8	**99.8**	68.3
	GPT-3.5	**96.8**	82.3	87.0	**98.2**	88.3	20.0	45.8	22.7	26.7	63.1
Existing Approaches	ALOE	34.2	33.2	31.0	19.5	30.5	21.5	27.5	23.3	20.8	26.8
	Flamingo-1.1B	49.3	35.3	48.5	19.2	38.2	18.8	57.3	84.2	18.0	41.0
Ours	LLaMA2-7b-Object	59.3	**100.0**	**98.8**	96.8	50.4	17.3	**87.0**	**99.5**	19.2	**69.8**

(b) MEWL

Table 2: Results of LLaMA2-7b with train-from-scratch visual encoders on sub-tasks in ACRE and MEWL. **Bolded results** are the best results, and underlined ones are the second best. All language baselines are frozen, except BERT which is finetuned on MEWL tasks. The results show that frozen LLaMA2 with learned visual encoder perform significantly better than its language counterpart and even outperform the existing approaches.

the token embedding layer of language backbone frozen, and only train the visual encoder and the projection layer. If this MLLM with a trained visual encoder can perform better than its language backbone with oracle visual perception, then it provides further evidence for the above interpretation of the frozen LLM as a highly transferable system.

6.1 Variants of Image Inputs

In order to run these experiments, we consider three variants of image inputs. Figure B.3 shows the examples of each variant.

Symbol. A frame is represented by a set of multi-hot object representations, where each object representation is the concatenation of its one-hot vectors for object attributes (i.e., color, material, and shape) and a vector of object location information. This mimics the experiments in §5 by assuming oracle visual perception, and allows us to directly contrast language and visual inputs.

Object. A frame is represented by object representations, where each object is an object crop from the frame. This variant assumes ground truth object detection exists in order to control the factors of reasoning performance.

Image. A frame is represented by its RGB image. This variant simplifies the inputs the most, but requires the visual encoder to encode object properties and spatial relationships between objects directly from the frames.

6.2 Language Baseline

For our language baseline, we provide a frozen LLM directly with language descriptions of the abstract visual reasoning problem. Frame captions can be considered as oracle visual perception, where each frame is represented by its caption (e.g., "There is a blue cylinder and a brown cube.").

6.3 Implementation Details

On ACRE, we use the training set with 6K samples, where each sample contains 6 context frames and 4 query frames. Thus, the training set has 24K sequences. On MEWL, we use the training sets of the 9 sub-tasks, each of which involves 600 samples. Thus, the training set has 5400 sequences.

For the language backbone, we use LLaMA2 with 7 billion parameters (Touvron et al., 2023). For the visual backbone, to encode image inputs, we use a 2-layer ViT (Dosovitskiy, 2020) with 4 attention heads and 768-hidden dimensional space; to encode symbolic representations of images, we use a symbolic encoder which encodes object at-

	ACRE				MEWL									
	I.I.D.	Comp.	Sys.	Avg.	shape	color	material	object	composite	relation	bootstrap	number	pragmatic	Avg.
LLaMA2-7b-Image	75.8	77.7	71.7	75.1	35.0	99.8	57.7	26.2	32.7	19.8	31.8	45.2	21.3	41.1
LLaMA2-7b-Object	95.5	97.5	86.5	93.2	59.3	100.0	98.8	96.8	50.4	17.3	87.0	99.5	19.2	69.8
LLaMA2-7b-Symbol (Linear)	91.0	94.9	86.8	90.9	100.0	99.8	100.0	98.0	42.5	18.0	35.0	78.2	18.3	65.5
LLaMA2-7b-Symbol (MLP)	98.3	99.5	84.6	94.1	100.0	100.0	100.0	98.8	71.3	16.2	91.3	99.7	22.3	77.7

Table 3: Analysis on the presence of object-centric information. -Symbol rows can be considered as upper bound, since the inputs are symbolic representations of images. The performance gap between -Image and -Object reflects the importance of object-centric inductive bias in abstract visual reasoning tasks.

tributes with embedding layers and encodes objects' location information[5] with a linear layer.

During finetuning, we freeze the language backbone and finetune the visual encoder and the linear projection. We use the AdamW optimizer with a learning rate of 3×10^{-5}. We finetune the visual backbone for 20 epochs on ACRE, and 40 epochs on MEWL. The batch size is set to 64.

6.4 Results

Table 2 shows the results of LLaMA2-7b with learned visual encoders on ACRE and MEWL. On ACRE, we observe that LLaMA2 with train-from-scratch visual encoders can perform significantly better than their language-only counterpart. These models can even outperform majority of the multimodal state-of-the-art, including IV-CL (Sun et al., 2024) and LRR (Bhattacharyya et al., 2023), which are pretrained with video data. On MEWL, we observe the same pattern that LLaMA2 with learned visual encoders can outperform prior state-of-the-art and also the language baselines which assume perfect visual perception.

In Table 3, we further investigate different ways to represent an image. The large performance gap between LLaMA2-7b-Image and -Object (e.g., average of 41.1% versus 69.8% on MEWL), indicating that object-centric information is important for the pretrained transformer blocks to better solve abstract visual reasoning tasks. In all, these results demonstrate that with a frozen language backbone, learning just the visual encoder from scratch can already improve the model's performance on abstract visual reasoning tasks significantly. However, task-specific design choices, such as object-centric representations, would be needed.

7 Discussion

The question of whether LLMs are "abstract reasoners" has consequences for how we understand and thus how we develop increasingly advanced artificial intelligence. The challenge is that there is no consensus for what it means to be an "abstract reasoner". In their recent work, Gendron et al. (2024) operationalize abstract reasoning as the ability to transfer zero-shot to a range of complex reasoning tasks. They find that LLMs perform poorly on this evaluation, and thus conclude that they are not abstract reasoners.

In this work, we reproduce Gendron et al. (2024)'s findings, but push back against their interpretation. In particular, we provide new experiments which show that tuning just the embedding layer is remarkably effective. Indeed, across a variety of textual and multimodal tasks, frozen pretrained LLMs can achieve high levels of performance as long as the input representations are adapted sufficiently for each task [6].

It seems too stringent a criteria to require that that abstract reasoners perform arbitrary tasks on arbitrary inputs without adaptation. By way of counterargument, consider the good old fashioned AI (GOFAI) systems of the 1990s, which typically included symbolic systems internally, e.g., databases implemented in SQL or rules for logical inference implemented in PROLOG. By most intuitive definitions, these databases and rules would be considered "abstract" and the tasks the systems performed over them would be "reasoning". But we would not expect these systems to operate well over a database implemented in MongoDB, or to apply rules defined by Python. Rather, the need to operate on representations of a particular format is a consequence of, not an exception to, the system's abstraction.

[5] Each object location is represented as $[x_1, y_1, x_2, y_2, w, h, w \times h]$

[6] While we argue that input-level finetuning can enable pretrained models to perform well on a range of tasks, we acknowledge that this does not necessarily imply the models have acquired generalized abstract reasoning in a cognitive sense. Rather, it may reflect the alignment of input representations with the pretrained model's existing capabilities. A more robust theoretical framework would be needed to precisely distinguish between mere representational alignment and true abstraction across domains and tasks.

Of course, we don't claim that the internal processing of an LLM is exactly analogous to that of a GOFAI system. Of course, in an LLM, tuning the input embedding layer might do more than simply "rerepresent", but rather might encode some task-specific processing as well. But interpreted loosely, the analogy is useful for highlighting how the question of adaptability and transferability relates to the question of abstraction and reasoning.

Indeed, this relationship has been considered in depth by philosophers of AI, long before LLMs. For example, Dennett (1997) appeals to transferability in his attempt to describe the difference between human cognition[7] and simpler computational systems:

> Consider the lowly thermostat...we might agree to grant it the capacity for about half a dozen different beliefs...it can believe the room is too cold or too hot, that the boiler is on or off...and so forth...suppose we *de-interpret* its beliefs and desires, it can believe the A is too F or G...and so forth....by attaching the thermostatic control mechanism to different input and output devices, it could be made to regulate the amount of water in a tank, or the speed of a train for instance...But as systems become perceptually richer and behaviorally more versatile, it becomes harder and harder to make substitutions in the actual links of the system to the world without changing the organization of the system itself. ...There comes to be a two-way constraint of growing specificity between the device and the environment. Fix the device in any one state and it demands a very specific environment in which to operate properly (you can no longer switch it easily from regulating temperature to regulating speed or anything else); but at the same time, if you do not *fix* the state it is in, but just plunk it down in a changed environment, its sensory attachments will be sensitive and discriminative enough to respond appropriately to the change...

Although Dennett is not discussing the notion of "abstract reasoners" *per se*, he observes that intelligent systems do not transfer well unless they are allowed to adapt[8]. Indeed, Dennett argues that this is a defining property, one that differentiates human-like intelligence from simpler (albeit perhaps more abstract) systems such as thermostats.

Dennett's argument is relevant here not because LLMs are human-like or even human-level in their reasoning abilities (they are far from it!). Rather, Dennett articulates a position that is implicit in contemporary discussions about LLMs and "abstract reasoning". That is, that we care about how well a system adapts to new environments because adapting well to new environments is a hallmark of intelligence. Indeed, this is often cited explicitly as the motivation for studies of this nature (e.g., "the question of whether or not LLMs can perform human-like reasoning remains open..." (Gendron et al., 2024)). But if evaluating human-likeness or human-levelness is the motivation for studying abstract reasoning, then arguments such as Dennett's provide a compelling case against using zero-shot transfer ability as a relevant metric.

Of course, there is another, more practical, argument for why we might care about whether LLMs are abstract reasoners, which is simply that we want LLMs to transfer well zero-shot to many tasks in order to facilitate easier, cheaper, and more efficient development of systems. Indeed, the thermostat's highly abstract design is a feature, not a bug. This type of hardware abstraction is what allows similar components and control mechanisms to be readily repurposed to support many types of use cases. A "human like" thermostat might be very undesirable.

Thus, before seeking to answer the question of whether LLMs are "abstract reasoners", we must first determine, as a community, why we care. Do we care because we want to understand how human-like they are, or do we care because we want to facilitate more efficient technological progress? Almost certainly, we care about both, but we should not expect the same experiments to bear on both lines of inquiry. Finding clarity around these questions–what is an abstract reasoner and why do we care about building one?–is the essential next step if we are to make progress toward either, or both, goals.

[7] Dennet's essay is not about reasoning, but rather about *intentional* systems, or systems that have true "beliefs" about the world and act according to them.

[8] While our experiments adapt the input layer (e.g., token embedding) of a model, adaptation does not have to be limited to the input layers. Indeed, adaptation throughout the model would be consistent with Dennett's argument. A full exploration of this is beyond the scope of this paper, but is an interesting direction for future work.

8 Conclusion

In this paper, we have (re-)opened the discussion of what it means to be an "abstract reasoner", and why it matters whether LLMs are "abstract reasoners". We have offered empirical results showing that off-the-shelf pretrained LLMs indeed perform poorly on reasoning benchmarks in a zero-shot setting. However, on a variety of textual and multimodal reasoning tasks, frozen pretrained LLMs can reach high levels of performance when the input embeddings are tuned. With this collection of empirical results, we argue that there is a need to determine why we care about whether LLMs are "abstract reasoners" before answering this question.

9 Acknowledgement

We would like to thank all reviewers and the area chair for their valuable feedback. We would like to thank Samuel Musker, Calvin Luo, and other members of the SuperLab at Brown University for their discussions and insights. The project depicted is sponsored in part by a Young Faculty Award from the Defense Advanced Research Projects Agency, Grant #D24AP00261. The content of the information does not necessarily reflect the position, or the policy of the government and no official endorsement of this work should be inferred.

References

Josh Achiam, Steven Adler, Sandhini Agarwal, Lama Ahmad, Ilge Akkaya, Florencia Leoni Aleman, Diogo Almeida, Janko Altenschmidt, Sam Altman, Shyamal Anadkat, et al. 2023. Gpt-4 technical report. *arXiv preprint arXiv:2303.08774*.

Yaniv Benny, Niv Pekar, and Lior Wolf. 2021. Scale-localized abstract reasoning. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, pages 12557–12565.

Apratim Bhattacharyya, Sunny Panchal, Reza Pourreza, Mingu Lee, Pulkit Madan, and Roland Memisevic. 2023. Look, remember and reason: Grounded reasoning in videos with language models. In *The Twelfth International Conference on Learning Representations*.

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

Daniel C. Dennett. 1997. True believers:the intentional strategy and why it works. In *Mind Design II: Philosophy, Psychology, and Artificial Intelligence*. The MIT Press.

David Ding, Felix Hill, Adam Santoro, Malcolm Reynolds, and Matt Botvinick. 2021. Attention over learned object embeddings enables complex visual reasoning. *Advances in neural information processing systems*, 34:9112–9124.

Alexey Dosovitskiy. 2020. An image is worth 16x16 words: Transformers for image recognition at scale. *arXiv preprint arXiv:2010.11929*.

Gaël Gendron, Qiming Bao, Michael Witbrock, and Gillian Dobbie. 2024. Large language models are not strong abstract reasoners. In *IJCAR*.

Dedre Gentner, Keith J Holyoak, and Boicho N Kokinov. 2001. Introduction: The place of analogy in cognition. *The analogical mind: Perspectives from cognitive science*, pages 1–19.

Simon Jerome Han, Keith J Ransom, Andrew Perfors, and Charles Kemp. 2024. Inductive reasoning in humans and large language models. *Cognitive Systems Research*, 83:101155.

Douglas R Hofstadter, Melanie Mitchell, et al. 1995. The copycat project: A model of mental fluidity and analogy-making. *Advances in connectionist and neural computation theory*, 2:205–267.

Edward J Hu, Yelong Shen, Phillip Wallis, Zeyuan Allen-Zhu, Yuanzhi Li, Shean Wang, Lu Wang, and Weizhu Chen. 2021a. Lora: Low-rank adaptation of large language models. *arXiv preprint arXiv:2106.09685*.

Sheng Hu, Yuqing Ma, Xianglong Liu, Yanlu Wei, and Shihao Bai. 2021b. Stratified rule-aware network for abstract visual reasoning. In *Proceedings of the AAAI Conference on Artificial Intelligence*, volume 35, pages 1567–1574.

Xiaoyang Hu, Shane Storks, Richard L Lewis, and Joyce Chai. 2023. In-context analogical reasoning with pre-trained language models. In *The 61st Annual Meeting Of The Association For Computational Linguistics*.

Drew Hudson and Christopher D Manning. 2019. Learning by abstraction: The neural state machine. *Advances in Neural Information Processing Systems*, 32.

Guangyuan Jiang, Manjie Xu, Shiji Xin, Wei Liang, Yujia Peng, Chi Zhang, and Yixin Zhu. 2023. Mewl: Few-shot multimodal word learning with referential uncertainty. In *International Conference on Machine Learning*, pages 15144–15169. PMLR.

Martha Lewis and Melanie Mitchell. 2024. Using counterfactual tasks to evaluate the generality of analogical reasoning in large language models. In *Proceedings of the Annual Meeting of the Cognitive Science Society*, volume 46.

Junnan Li, Dongxu Li, Silvio Savarese, and Steven Hoi. 2023. Blip-2: Bootstrapping language-image pre-training with frozen image encoders and large language models. In *International conference on machine learning*, pages 19730–19742. PMLR.

Haotian Liu, Chunyuan Li, Yuheng Li, and Yong Jae Lee. 2024. Improved baselines with visual instruction tuning. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, pages 26296–26306.

Haotian Liu, Chunyuan Li, Qingyang Wu, and Yong Jae Lee. 2023. Visual instruction tuning. *Advances in neural information processing systems*, 36:34892–34916.

Ilya Loshchilov and Frank Hutter. 2019. Decoupled weight decay regularization. In *International Conference on Learning Representations*.

Kevin Lu, Aditya Grover, Pieter Abbeel, and Igor Mordatch. 2022. Frozen pretrained transformers as universal computation engines. In *Proceedings of the AAAI conference on artificial intelligence*, volume 36, pages 7628–7636.

Jiayuan Mao, Chuang Gan, Pushmeet Kohli, Joshua B Tenenbaum, and Jiajun Wu. 2019. The neuro-symbolic concept learner: Interpreting scenes, words, and sentences from natural supervision. *arXiv preprint arXiv:1904.12584*.

Jack Merullo, Louis Castricato, Carsten Eickhoff, and Ellie Pavlick. 2023. Linearly mapping from image to text space. In *The Eleventh International Conference on Learning Representations*.

Melanie Mitchell. 2021. Abstraction and analogy-making in artificial intelligence. *Annals of the New York Academy of Sciences*, 1505(1):79–101.

Melanie Mitchell, Alessandro B Palmarini, and Arseny Moskvichev. 2023. Comparing humans, gpt-4, and gpt-4v on abstraction and reasoning tasks. *arXiv preprint arXiv:2311.09247*.

Arsenii Kirillovich Moskvichev, Victor Vikram Odouard, and Melanie Mitchell. 2023. The conceptarc benchmark: Evaluating understanding and generalization in the arc domain. *Transactions on machine learning research*.

Sam Musker, Alex Duchnowski, Raphaël Millière, and Ellie Pavlick. 2024. Semantic structure-mapping in llm and human analogical reasoning. *arXiv preprint arXiv:2406.13803*.

Joshua Stewart Rule. 2020. *The child as hacker: building more human-like models of learning*. Ph.D. thesis, Massachusetts Institute of Technology.

Aarohi Srivastava, Abhinav Rastogi, Abhishek Rao, Abu Awal Md Shoeb, Abubakar Abid, Adam Fisch, Adam R Brown, Adam Santoro, Aditya Gupta, Adrià Garriga-Alonso, et al. 2022. Beyond the imitation game: Quantifying and extrapolating the capabilities of language models. *arXiv preprint arXiv:2206.04615*.

Claire E Stevenson, Alexandra Pafford, Han LJ van der Maas, and Melanie Mitchell. 2024. Can large language models generalize analogy solving like people can? *arXiv preprint arXiv:2411.02348*.

Chen Sun, Calvin Luo, Xingyi Zhou, Anurag Arnab, and Cordelia Schmid. 2024. Does visual pretraining help end-to-end reasoning? *Advances in Neural Information Processing Systems*, 36.

Peter Tong, Ellis Brown, Penghao Wu, Sanghyun Woo, Adithya Jairam Vedagiri IYER, Sai Charitha Akula, Shusheng Yang, Jihan Yang, Manoj Middepogu, Ziteng Wang, et al. 2025. Cambrian-1: A fully open, vision-centric exploration of multimodal llms. *Advances in Neural Information Processing Systems*, 37:87310–87356.

Hugo Touvron, Louis Martin, Kevin Stone, Peter Albert, Amjad Almahairi, Yasmine Babaei, Nikolay Bashlykov, Soumya Batra, Prajjwal Bhargava, Shruti Bhosale, et al. 2023. Llama 2: Open foundation and fine-tuned chat models. *arXiv preprint arXiv:2307.09288*.

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

Chi Zhang, Feng Gao, Baoxiong Jia, Yixin Zhu, and Song-Chun Zhu. 2019. Raven: A dataset for relational and analogical visual reasoning. In *Proceedings of the IEEE/CVF conference on computer vision and pattern recognition*, pages 5317–5327.

Chi Zhang, Baoxiong Jia, Mark Edmonds, Song-Chun Zhu, and Yixin Zhu. 2021a. Acre: Abstract causal reasoning beyond covariation. In *Proceedings of the ieee/cvf conference on computer vision and pattern recognition*, pages 10643–10653.

Chiyuan Zhang, Maithra Raghu, Jon Kleinberg, and Samy Bengio. 2021b. Pointer value retrieval: A new benchmark for understanding the limits of neural network generalization. *arXiv preprint arXiv:2107.12580*.

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.

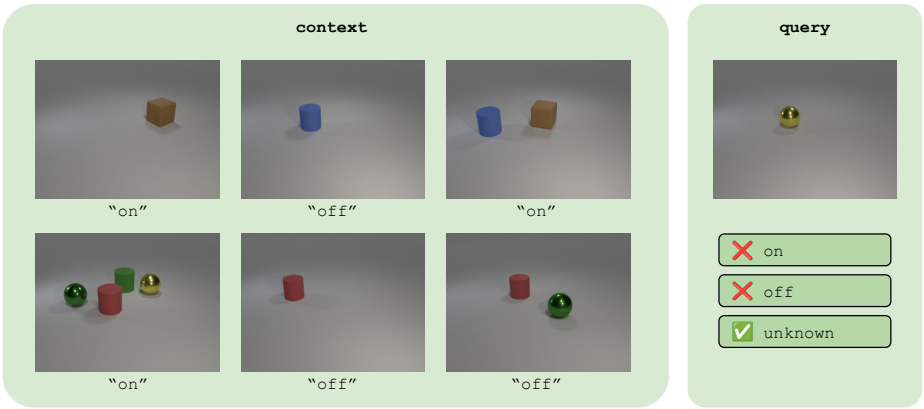
B Additional Figures

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

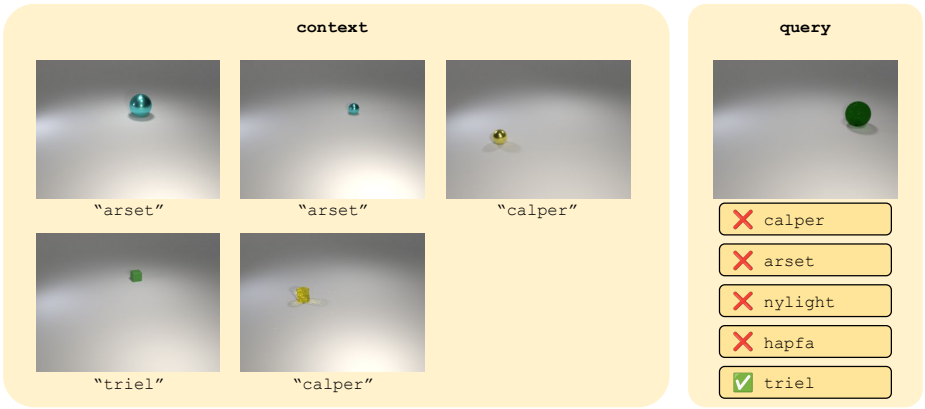


Figure B.2: Data examples of abstract visual reasoning tasks.

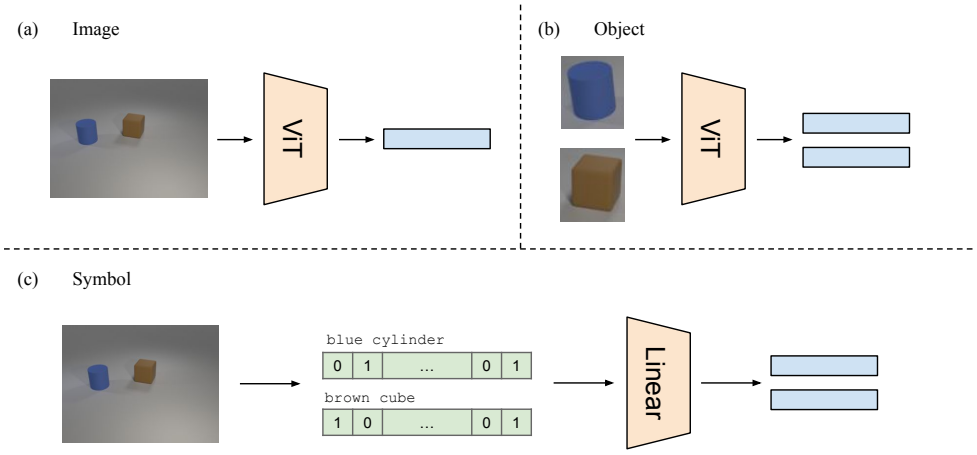


Figure B.3: Examples of variants of image inputs. (a) An image is directly fed into a ViT and obtain an image representation. (b) Each object crop is fed into a ViT and obtain an object representation. (c) Each object is parsed into a multi-hot vector, and a linear layer will output a corresponding object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

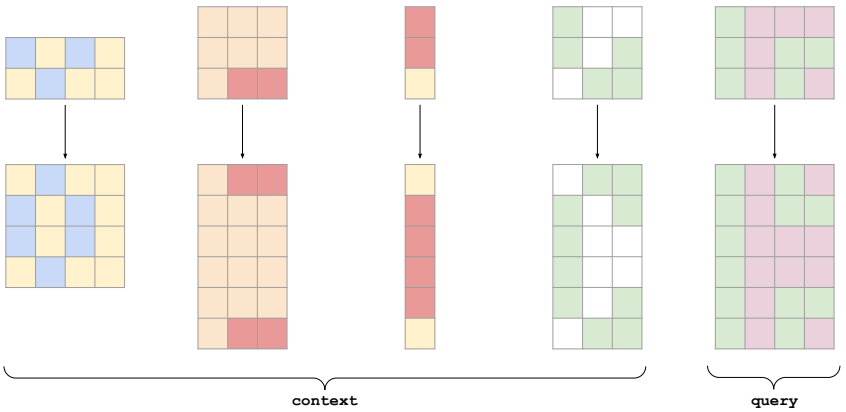


Figure B.4: Example of ARC dataset. There are 4 context examples and 1 query, where each example has an input grid (top) and an output grid (bottom). Each grid is represented as an integer array, where each integer refers to a color. In this example, the task is to generate the symmetry of the input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

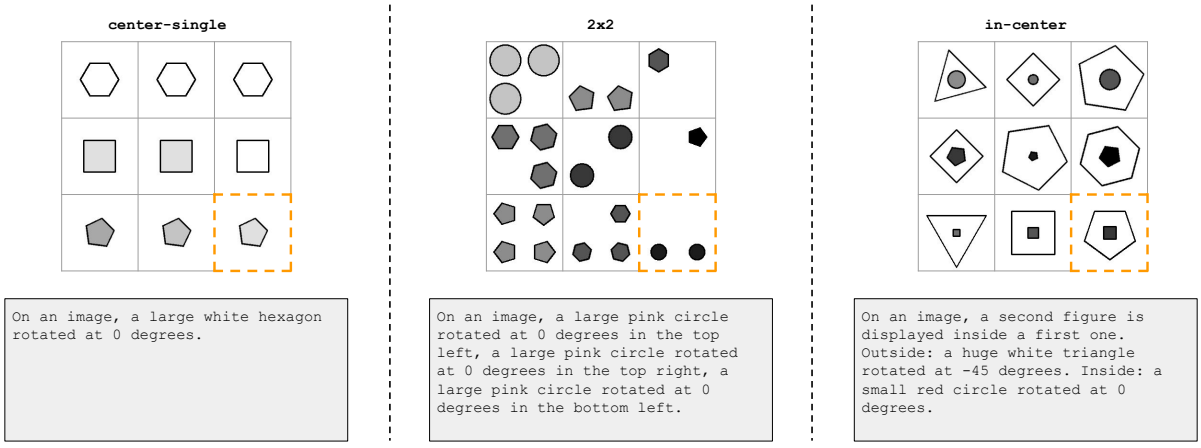


Figure B.5: Examples of RAVEN^T tasks used in generalizability and data efficiency analysis. Top shows the data example, and bottom shows the language description of the first frame in each example. The task is to fill in the ninth pattern (highlighted in orange) given the eight context frames. We focus 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.