Regulation Using Large Language Models to Generate Synthetic Data for Evaluating Analogical Ability

Donghyeon Shin[1] , **Seungpil Lee**[1] , **Klea Lena Kovačec**[1] and **Sundong Kim**[1]

[1]Gwangju Institute of Science and Technology

{shindong97411, iamseungpil, klealk8, sdkim0211}@gmail.com

Abstract

As the reasoning abilities of artificial intelligence gain more attention, generating reliable benchmarks to evaluate reasoning capabilities is becoming increasingly significant. Abstract and Reasoning Corpus (ARC) is one of the introduced reasoning benchmarks, providing challenging problems that artificial intelligence has yet to solve. While ARC has been recognized for assessing reasoning abilities, its evaluation format has presented challenges for analysis, leading to the necessity for revised benchmarks. Considering this, our research aimed to modify the benchmark into a multiple-choice language format to make it more suitable for evaluating large language models (LLMs), termed MC-LARC. We evaluated the analogical reasoning abilities of ChatGPT4V with MC-LARC, confirming that 1) a multiple-choice format can support the language model's reasoning capabilities and 2) facilitate evidence analysis. However, we noticed LLMs relying on shortcuts when tackling MC-LARC. By analyzing this, we identified areas to consider in multiple-choice synthesis and specified criteria for what constitutes good choices based on these findings.

1 Introduction

Research on artificial intelligence with reasoning capabilities is attracting attention, leading to the proposal of benchmarks to measure such abilities. The Abstraction and Reasoning Corpus (ARC) is one such benchmark designed to evaluate reasoning abilities. Each ARC task consists of 2–5 examples where both input and output are provided, along with one problem where only the input is given. The goal is to infer the rule from the examples and deduce the answer to the problem. The input and output grids in ARC can range from a minimum 1×1 grid to a maximum 30×30 grid, with each grid filled with up to 10 different colors. Unlike existing reasoning benchmarks, ARC is specialized in evaluating reasoning abilities alone by reducing the amount of prior knowledge and data required to solve the problems. In this way, ARC is designed to effectively assess inferential abilities; however, its evaluative format, which includes generation, poses challenges in gauging the level of inferential ability achieved. Even if the problem-solving logical process was correct, any deviation in the generated grid leads to the entire response being deemed incorrect. Therefore, evaluating the accuracy of the problem-solving logical process becomes difficult. This challenge is also evident in derived datasets like Mini-ARC [Kim *et al.*, 2022] and 1D-ARC [Xu *et al.*, 2023]. These datasets underwent transformations such as fixed grid sizes or reducing 2D arrays to 1D arrays respectively, but they still share the same limitation in that the evaluative format involves generation.

To address this limitation, this paper proposes a modified benchmark called MC-LARC that transforms the evaluation format from generation to selection. It converts the dataset into a multiple-choice language format by using Large Language Models (LLMs) to generate four alternative options based on the correct answer to ARC tasks. We conducted experiments to investigate the impact of the transformation into multiple-choice form and found the following two points: 1) we confirmed an increase in the accuracy of LLMs on ARC problems from about 10% to 75%. This indicates that the options in MC-LARC have served a supportive role in the inference of LLMs, which are more aligned with language generation and comprehension than image processing. 2) Evaluating the extent of the inferential abilities of LLMs becomes more clearly feasible. However, it was observed that LLMs used shortcuts while solving MC-LARC, finding the correct answer by considering the form or internal context of the choices to eliminate inappropriate options, rather than utilizing reasoning abilities. Based on this analysis, it was confirmed that when synthesizing data into a multiple-choice format using LLMs, sufficient and accurate context information should be provided to avoid unnecessary additional information. Additionally, this analysis established criteria for what constitutes good multiple-choice options.

2 Related Works

2.1 Benchmark for Analogy Abstraction Tasks

Abstraction and Reasoning Corpus (ARC) The Abstraction and Reasoning Corpus (ARC) benchmark [Chollet, 2019] was created for the purpose of measuring intelligence in computer systems. This benchmark requires inference based on complex prior knowledge such as arithmetic abilities, geometric understanding, and topological understanding. The goal is to derive common rules from examples and apply them to infer the appropriate output image for a given test input image. Each task provides 2–5 pairs of example input and output images. The original ARC benchmark consists of 400 training set, 400 evaluation set, and 200 test set. Moreover, the ARC benchmark is represented as 2D matrices.

Language-complete ARC (LARC) The LARC [Acquaviva *et al.*, 2022] dataset consists of 400 ARC training data, each accompanied by 1) a description of the input image and 2) a natural language description of the rules between the input and output images. Both the input description and the output description must be language-complete. Language-complete refers to having sufficient relevant information even when neither input nor output images are provided. In other words, humans should be able to understand the task sufficiently based solely on the description of LARC without the presence of images. A language-complete LARC is manifested in the Refined LARC below Figure 1.

2.2 Synthetic Data Creation Using LLM

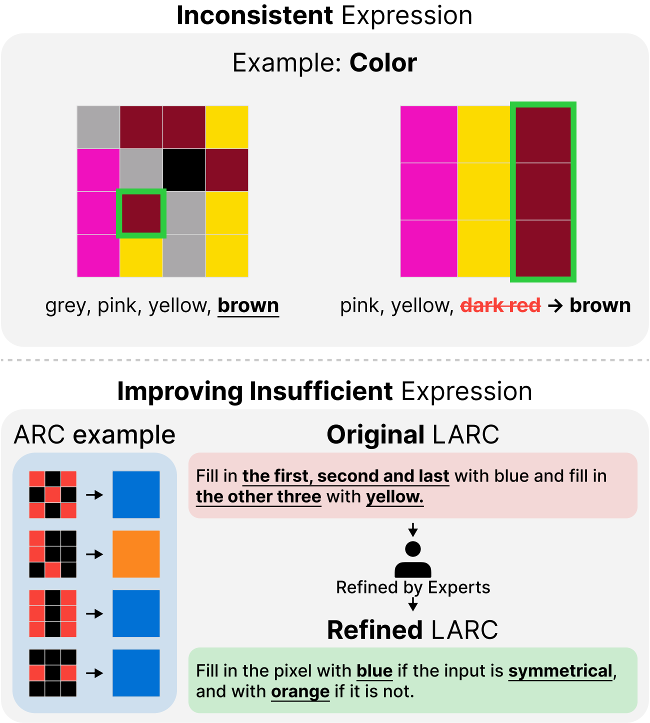
Current trends in synthetic data creation with LLMs focus on enhancing the faithfulness and utility of generated data to better align with real-world data distributions. [Veselovsky *et al.*, 2023] employed grounding, filtering, and taxonomy-based generation techniques to improve the faithfulness of synthetic data for sarcasm detection, which is relevant to our work on MC-LARC's benchmark for assessing LLM reasoning abilities. [Schimanski *et al.*, 2024] created synthetic data for evidence-based QA by generating questions and sources, and applying quality filters to distinguish relevant from irrelevant questions. Challenges in generating faithful data have led to proposed filtering techniques to address distribution deviations and improve quality. [Yang *et al.*, 2024] highlights the importance of avoiding shortcuts in analyzing correct answers, urging us to conduct further experiments to detect flaws and improve methodologies.

3 Methodology

We created MC-LARC through the following two steps: 1) manually refining the existing LARC, and 2) utilizing ChatGPT4 to generate wrong options based on LARC.

Refining process The original LARC exhibited significant quality issues, as evidenced by Figure 1. These issues manifested primarily in 1) inconsistencies across expressions for the same concept and 2) a lack of information in the provided explanations. For instance, the upper part of Figure 1 illustrates different representations for the same concepts, leading to user confusion. Additionally, the explanations accompanying the tasks often omitted crucial information necessary for their successful completion. These issues emerged as a consequence of the dataset's compilation by numerous non-experts using Amazon Mechanical Turk.

In addition to the issues highlighted in Figure 1, there were further cases of inconsistency throughout the dataset. These inconsistencies involved not only color but also shape representations and grid manipulation operations. The presence of these multiple issues complicates the process of generating new datasets based on LARC, emphasizing the challenges of relying on flawed data sources.


Inconsistent Expression
Example: **Color**
grey, pink, yellow, **brown**
pink, yellow, ~~dark red~~ → **brown**
Improving Insufficient Expression
ARC example
Original LARC
Refined by Experts
Refined LARC

Fill in **the first, second and last** with blue and fill in **the other three** with **yellow.**

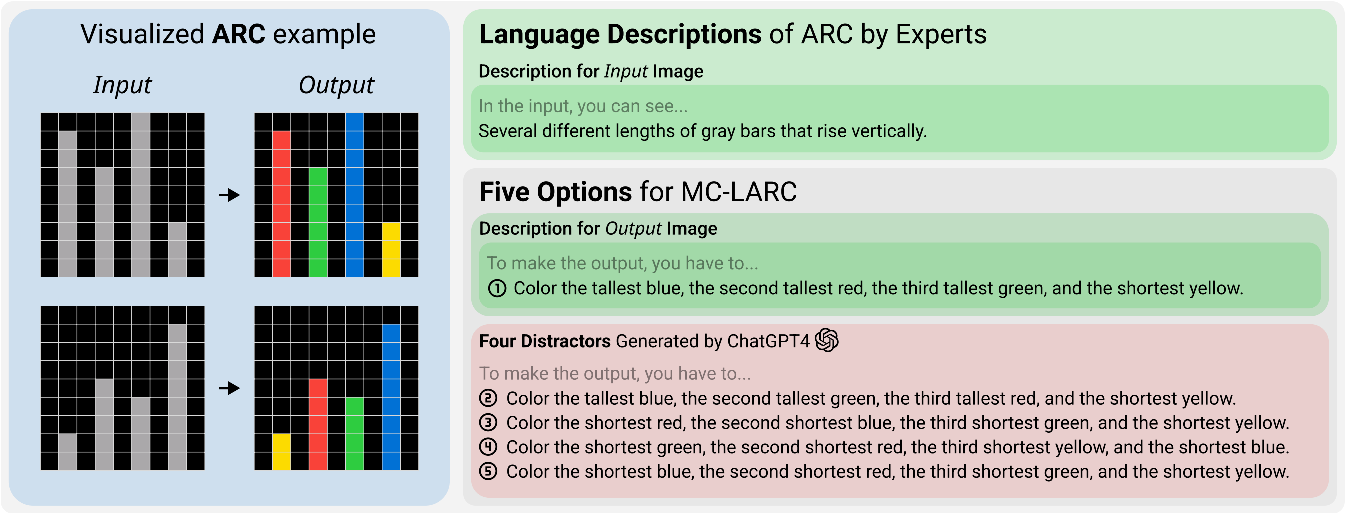
Fill in the pixel with **blue** if the input is **symmetrical**, and with **orange** if it is not.

Figure 1: Two main issues of LARC. (Upper part) There are instances where different expressions are used for the same concept within LARC. For example, some LARC expressions describe brown as "dark red". (Lower part) This task involves identifying the symmetry of the input grid to predict the output image result. However, some original LARC expressions provide insufficient information necessary for ARC problem-solving. These have been revised to contain sufficient and accurate information by experts.

To address these issues, we conducted a refining process to enhance quality. This process prioritized ensuring consistency in expressions and rectifying erroneous representations. Figure 1 provides an overview of this refining process.

Generating wrong options with ChatGPT4 Based on the given output description of LARC, we generated four distractors through ChatGPT4, as illustrated in Figure 2. However, allowing unrestricted generation of distractors led to issues such as creating out-of-context choices unrelated to the task. To address this problem, we improved by adding constraints during the prompt level. The constraints added to the prompt are as follows:

- **In context vocabulary**: To generate plausible distractors, it was necessary to limit the expressions within the context that aligns with the ARC domain. To achieve this, two contextual constraints were imposed. One involved adding descriptions about the ARC environment, while the other entailed mentioning specific words that should not be used.
- **Length of options**: When generating distractors for lengthy options, there were cases where LLM produced


Visualized **ARC** example
Input
Output
Language Descriptions of ARC by Experts
Description for *Input* Image

In the input, you can see...
Several different lengths of gray bars that rise vertically.

Five Options for MC-LARC
Description for *Output* Image

To make the output, you have to...
① Color the tallest blue, the second tallest red, the third tallest green, and the shortest yellow.

Four Distractors Generated by ChatGPT4

To make the output, you have to...
② Color the tallest blue, the second tallest green, the third tallest red, and the shortest yellow.
③ Color the shortest red, the second shortest blue, the third shortest green, and the shortest yellow.
④ Color the shortest green, the second shortest red, the third shortest yellow, and the shortest blue.
⑤ Color the shortest blue, the second shortest red, the third shortest green, and the shortest yellow.

Figure 2: The composition of MC-LARC. It consists of a visualized ARC example and five multiple-choice options. The five multiple-choice options consist of the correct solution and four distractors. (Blue part) It visualizes ARC represented as a 2D matrix. (Green part) It is LARC refined manually by experts. (Red part) Using ChatGPT4, four distractors were generated from the output description (Red boundary) of the refined LARC. To solve MC-LARC, the solver must identify common rules from the Visualized ARC example and choose the option that best describes those rules.

relatively short options, leading to easily solvable problems. Therefore, we restricted the LLM to generate incorrect options of similar lengths to the correct options.

- **Format**: When creating distractors, we ensured that the opening phrases of the sentences exactly matched the correct answer option's *'To make the output, you have to...'*. If the opening phrases of the incorrect options vary, it could lead to selecting the correct answer based on the format rather than the meaning of the sentence.

As shown in Figure 6, before constraints were added, the model generated options that were either completely irrelevant to the ARC problem context or altered parts that were not core concepts. These were classified as either `bad` or `moderate`. However, after the constraints were applied, the model did not produce any `bad` options, and the options were classified only as `best` or `moderate`. Despite this improvement, the model still faces the challenge of not being able to produce `best` options for all tasks.

4 Experiments

To verify that the augmented multiple-choice options generated by the LLM did not inadvertently reveal more information than intended, we conducted a control test where the LLM was presented with only the options, devoid of any accompanying images. If the options were crafted appropriately and free from informational bias, the LLM's expected accuracy rate would approximate 20%. Additionally, this image-free experiment required the LLM to justify its choice for each option.

4.1 Influence of Multiple Choices

For the MC-LARC, we asked the ChatGPT4V model 5 times per problem, and as shown in Table 1, the accuracy of correctly answered tasks out of the total 400 tasks was 75%.

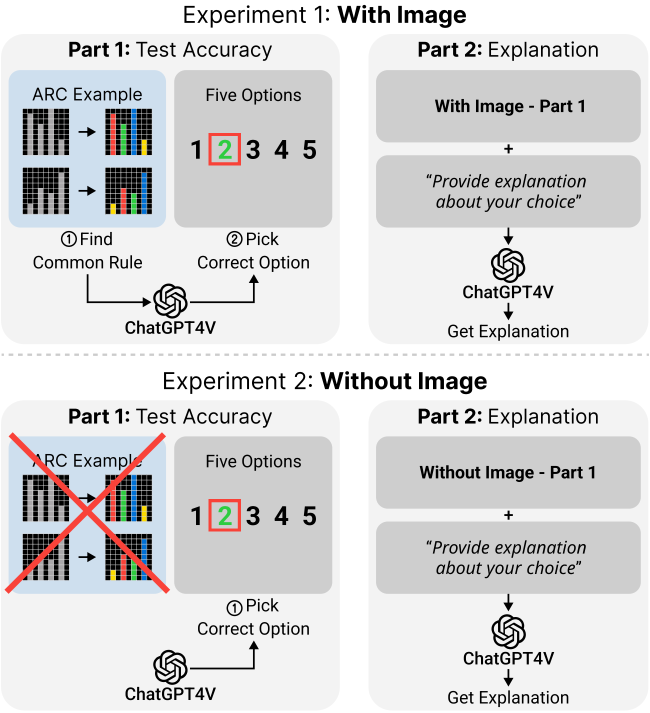
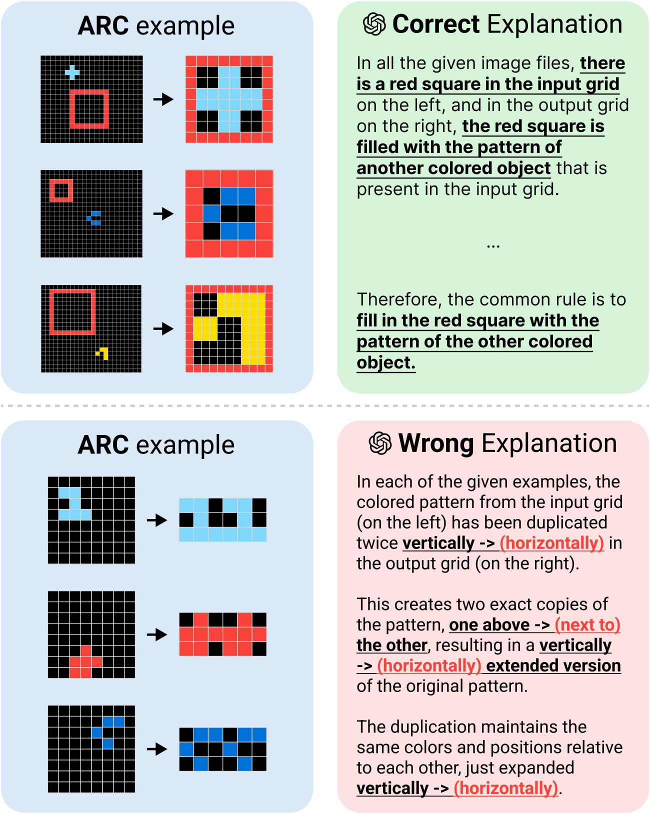


Figure 3: Overview of the conducted experiments. The upper part illustrates the first experiment, which includes visualized ARC example images, while the lower part depicts the second experiment, which does not include these images. Each experiment is divided into two parts. In Part 1, ChatGPT4 is tasked with solving the MC-LARC to measure accuracy. In Part 2, it is requested to provide explanations for its choices, in addition to completing the tasks from Part 1.

Table 1: A table summarizing the results of experiments where ChatGPT4V solved MC-LARC five times. It shows statistics on the accuracy and Krippendorff's Alpha score. The statistics show the mean, standard deviation, and 95% confidence interval for the accuracy. Krippendorff's Alpha score evaluates whether ChatGPT4V's responses are reliable across the five repeated experiments.

Category	Mean (%)	Std.	95% CI (%)	Alpha
With images	75.81	1.11	74.93 - 76.70	0.8329
Without images	64.61	1.75	63.08 - 66.14	0.7995



In all the given image files, **there is a red square in the input grid** on the left, and in the output grid on the right, **the red square is filled with the pattern of another colored object** that is present in the input grid.

...

Therefore, the common rule is to **fill in the red square with the pattern of the other colored object.**

In each of the given examples, the colored pattern from the input grid (on the left) has been duplicated twice **vertically ->** (horizontally) in the output grid (on the right).

This creates two exact copies of the pattern, **one above ->** (next to) **the other**, resulting in a **vertically -> (horizontally) extended version** of the original pattern.

The duplication maintains the same colors and positions relative to each other, just expanded **vertically ->** (horizontally).

Figure 4: A result of requesting an explanation of the experiments with provided images. (Upper part) It shows an example where the answer to MC-LARC is correctly chosen. (Lower part) It demonstrates the incorrect answers due to failure to infer the correct solution.

Considering that the accuracy of LLMs on ARC tasks is around 10% [Qiu *et al.*, 2024], this is certainly a high score. Additionally, Krippendorff's Alpha score of approximately 0.83 confirmed that the LLM was consistently reasoning the answers.

To further evaluate the reasoning process of the LLM, we additionally asked for the reason behind selecting each option. As a result, there were cases where both the answer and the reasoning process were correct or both were incorrect, but there were almost no cases where the answer was correct but the explanation was wrong, or where the answer was wrong but the explanation was correct. This indicates a decrease in the errors of generating correct answers through incorrect reasoning processes or giving inconsistent answers, which tend to occur when LLMs directly solve ARC tasks [Lee *et al.*, 2024]. Therefore, even when multiple-choice options, including incorrect options along with the answer description, were provided, we could confirm that the LLM's reasoning ability was partially improved.

4.2 Problems on Augmentation

However, there were indications that the LLM found a shortcut when solving MC-LARC. MC-LARC should be solved by inferring the rule from the given images and choosing the correct option, but the LLM achieved an accuracy of 65% even when the task was provided without images. The Krippendorff's Alpha score was also 0.79, not much lower than the experiment with images provided. This can be understood as evidence that the LLM found a consistent logic for getting the correct answers.

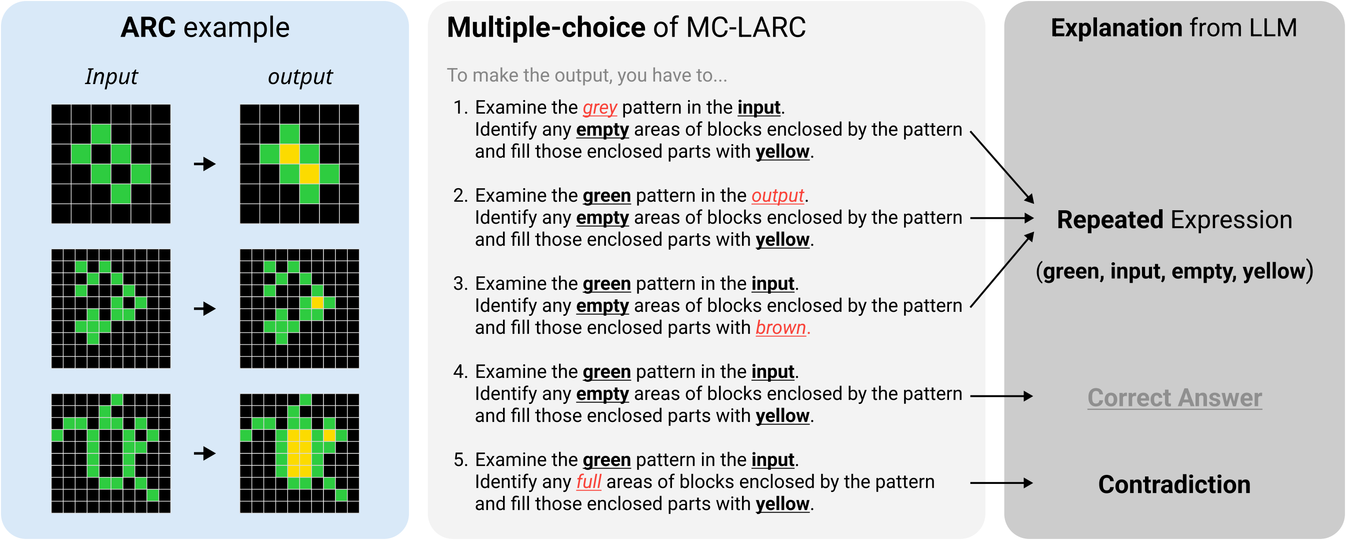
To analyze how the LLM solved MC-LARC without the problem images, we additionally asked the LLM to explain the reasoning behind its answers. As shown in Figure 5, we found that the LLM inferred the correct option by 1) choosing the option with the most repeated expressions and 2) eliminating options that were self-contradictory.

We point out two problems in the generation process: First, generating four different incorrect options from one correct option became problematic, as the correct option naturally included more repeated words than the incorrect options. Second, not providing image and context information for option generation led to contradictory or incompatible expressions in some options. Therefore, from this experiment, we can conclude that to fairly evaluate reasoning ability, the process of generating choices should be improved to avoid providing additional information that could serve as a shortcut.

4.3 Good Option and Bad Option

From the two experiments above, we confirmed that converting to a multiple-choice format has advantages as an inference problem in two aspects: 1) providing additional information to solve the reasoning problem, and 2) allowing for a more transparent evaluation of the reasoning process. However, we also found cases where unintended shortcuts were discovered, and to address this issue, the process of augmenting choices needs to be improved. But before improving the choice generation process, this question must be answered first: What distinguishes a good choice from a bad choice?

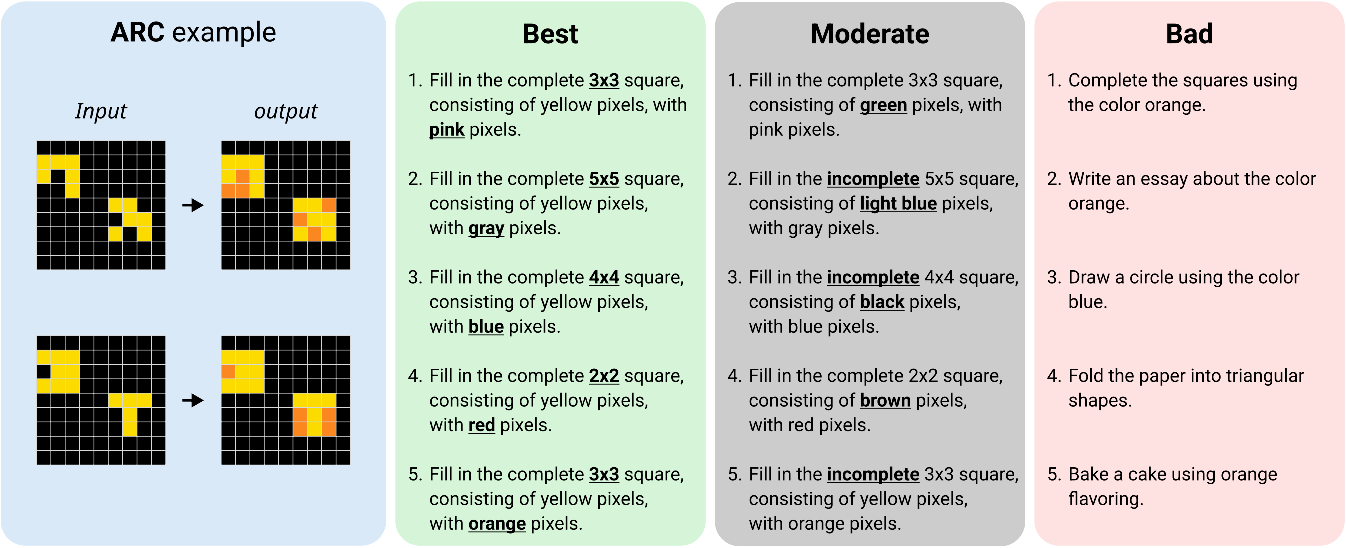
As we examined the augmented choice examples generated by the LLM, we were able to categorize the choices into three levels of quality, as shown in Figure 6. The best choices modified the core part of the problem that fits the context. In ARC, the core is the part where a change occurs between images, so in the given examples, completing a square by filling in orange pixels is the core. Thus, choices questioning the change to orange can be considered the best type of choice. Next, choices that were possible to predict from the input image but did not capture the core of the problem were of moderate quality. Examples include using colors not present in the input image or specifying grid sizes that were not present. Finally, choices that included cases that cannot occur in the ARC domain at all were the worst. Commands like "Write an



To make the output, you have to...

1. Examine the *grey* pattern in the **input**. Identify any **empty** areas of blocks enclosed by the pattern and fill those enclosed parts with **yellow**.
2. Examine the **green** pattern in the *output*. Identify any **empty** areas of blocks enclosed by the pattern and fill those enclosed parts with **yellow**.
3. Examine the **green** pattern in the **input**. Identify any **empty** areas of blocks enclosed by the pattern and fill those enclosed parts with *brown*.
4. Examine the **green** pattern in the **input**. Identify any **empty** areas of blocks enclosed by the pattern and fill those enclosed parts with **yellow**.
5. Examine the **green** pattern in the **input**. Identify any *full* areas of blocks enclosed by the pattern and fill those enclosed parts with **yellow**.

Figure 5: Example of an experiment without an image. When given five options, the LLM solves the problem by analyzing them in the following manner. By examining the options, the LLM identifies repeated expressions and excludes the options that use different vocabulary from the others. Additionally, it excludes options that cannot be represented in the ARC grid by identifying semantic contradiction within the sentences themselves.



Best

1. Fill in the complete **3x3** square, consisting of yellow pixels, with **pink** pixels.
2. Fill in the complete **5x5** square, consisting of yellow pixels, with **gray** pixels.
3. Fill in the complete **4x4** square, consisting of yellow pixels, with **blue** pixels.
4. Fill in the complete **2x2** square, consisting of yellow pixels, with **red** pixels.
5. Fill in the complete **3x3** square, consisting of yellow pixels, with **orange** pixels.

Moderate

1. Fill in the complete 3x3 square, consisting of **green** pixels, with pink pixels.
2. Fill in the **incomplete** 5x5 square, consisting of **light blue** pixels, with gray pixels.
3. Fill in the **incomplete** 4x4 square, consisting of **black** pixels, with blue pixels.
4. Fill in the complete 2x2 square, consisting of **brown** pixels, with red pixels.
5. Fill in the **incomplete** 3x3 square, consisting of yellow pixels, with orange pixels.

Bad

1. Complete the squares using the color orange.
2. Write an essay about the color orange.
3. Draw a circle using the color blue.
4. Fold the paper into triangular shapes.
5. Bake a cake using orange flavoring.

Figure 6: Three examples of multi-choice options augmented differently by the LLM. The given problem is to fill in an object with holes with the color orange to make a 3×3 square, where the size of the square and the color are the core aspects of the problem. The good example demonstrates an understanding of the core of the problem and provides consistent variations, while the poorer examples increasingly include choices that are unrelated to the problem and inconsistent.

essay" are irrelevant to ARC and do not require any reasoning process to solve the problem, making them poor choices. Therefore, good text descriptions and choices should 1) include the core of the problem in the choices, and 2) be consistent within the context of the problem. Identifying the criteria in form and content needed to generate good choices during the augmentation process is the contribution of this study.

5 Conclusion

In conclusion, to overcome the limitations of the existing ARC in measuring inferential reasoning ability, we created a new multiple-choice dataset called MC-LARC. As a result, the multiple-choice format allowed for a clearer analysis of logical flow during problem-solving and provided supplementary support for the solver's reasoning abilities. However, in an additional control experiment without images, we found that the LLM solved problems by finding shortcuts instead of using reasoning abilities. This highlights the regulation needed when using LLMs to synthesize multiple-choice questions. Based on these findings, we propose specific conditions for designing multiple-choice questions that effectively evaluate the required reasoning abilities without enabling shortcuts.

These findings have several important implications. Firstly, they offer valuable insights into the appropriate methods for evaluating inferential reasoning, demonstrating the potential of using multiple-choice questions for this purpose. Secondly, by identifying the constraints to consider when using LLMs to synthesize multiple-choice questions, this research paves the way for the development of more sophisticated and automated high-quality question generators.

Acknowledgements

This work was supported by the IITP (RS-2023-00216011, No. 2019-0-01842), funded by the Ministry of Science and ICT, Korea. Additionally, the Azure OpenAI API used for this work was supported by the Accelerating Foundation Models Research Grant from Microsoft.

References

[Acquaviva *et al.*, 2022] Samuel Acquaviva, Yewen Pu, Marta Kryven, Theodoros Sechopoulos, Catherine Wong, Gabrielle Ecanow, Maxwell Nye, Michael Tessler, and Joshua B. Tenenbaum. Communicating Natural Programs to Humans and Machines. In *NeurIPS*, 2022.

[Chollet, 2019] François Chollet. On the Measure of Intelligence. *arXiv:1911.01547*, 2019.

[Kim *et al.*, 2022] Subin Kim, Prin Phunyaphibarn, Donghyun Ahn, and Sundong Kim. Playgrounds for Abstraction and Reasoning. In *NeurIPS Workshop on nCSI*, 2022.

[Lee *et al.*, 2024] Seungpil Lee, Woochang Sim, Donghyeon Shin, Sanha Hwang, Wongyu Seo, Jiwon Park, Seokki Lee, Sejin Kim, and Sundong Kim. Reasoning Abilities of Large Language Models: In-Depth Analysis on the Abstraction and Reasoning Corpus. *arXiv:2403.11793*, 2024.

[Qiu *et al.*, 2024] Linlu Qiu, Liwei Jiang, Ximing Lu, Melanie Sclar, Valentina Pyatkin, Chandra Bhagavatula, Bailin Wang, Yoon Kim, Yejin Choi, Nouha Dziri, and Xiang Ren. Phenomenal Yet Puzzling: Testing Inductive Reasoning Capabilities of Language Models with Hypothesis Refinement. In *ICLR*, 2024.

[Schimanski *et al.*, 2024] Tobias Schimanski, Jingwei Ni, Mathias Kraus, Elliott Ash, and Markus Leippold. Towards Faithful and Robust LLM Specialists for Evidence-Based Question-Answering. *arXiv:2402.08277*, 2024.

[Veselovsky *et al.*, 2023] Veniamin Veselovsky, Manoel Horta Ribeiro, Akhil Arora, Martin Josifoski, Ashton Anderson, and Robert West. Generating Faithful Synthetic Data with Large Language Models: A Case Study in Computational Social Science. *arXiv:2305.15041*, 2023.

[Xu *et al.*, 2023] Yudong Xu, Wenhao Li, Pashootan Vaezipoor, Scott Sanner, and Elias B Khalil. LLMs and the Abstraction and Reasoning Corpus: Successes, Failures, and the Importance of Object-based Representations. *Transactions on Machine Learning Research*, 2023.

[Yang *et al.*, 2024] Kaiqi Yang, Hang Li, Hongzhi Wen, Tai-Quan Peng, Jiliang Tang, and Hui Liu. Are Large Language Models (LLMs) Good Social Predictors? *arXiv:2402.12620*, 2024.