AugARC: Augmented Abstraction and Reasoning Benchmark for Large Language Models

Kiril Bikov[1], **Mikel Bober-Irizar**[1], **Soumya Banerjee**[1,*]

[1]University of Cambridge
Cambridge
United Kingdom
[*]Corresponding author: sb2333@cam.ac.uk

Abstract

The Abstraction and Reasoning Corpus (ARC) benchmarks broad generalization, and poses a significant challenge to existing machine learning models. In this work, we introduce augmented ARC datasets and a new benchmark (AugARC) for large-language models (LLMs), which measures abstraction and reasoning. We evaluate the accuracy of base LLMs on AugARC and show a consistent improvement in performance compared to the normal ARC benchmark. Using augmented ARC data, we fine-tune LLMs and observe a significant gain in ARC accuracy after training. Due to the limited size of the ARC training dataset (400 tasks), previous studies have not attempted to train LLMs on ARC. Our augmentation of ARC allows us to overcome this limitation. Using a reflection approach, we combine LLMs and a previous domain specific language (DSL) solver. Our work introduces an augmented version of ARC - AugARC, and motivates further research into enhancing data quality for better reasoning in AI systems.

Introduction

Despite significant progress in machine learning, today's AI systems still lack human-level abstract reasoning Korteling et al. (2021); Boden et al. (2017); Shneiderman (2020). To address the gap between human intelligence and AI models, François Chollet created the Abstraction and Reasoning Corpus (ARC) Chollet (2019). ARC consists of 1000 visual tasks, that capture essential aspects of abstraction and analogy. The ARC tasks are split into 400 for training, 400 for evaluation and hidden 200 tasks for testing. A Program Synthesis approach from 2020 solved 40% of the complete evaluation set Icecuber (2023), and a voting ensemble from 2024 solved 40.25% of the tasks in the evaluation set Bober-Irizar and Banerjee (2024).

We aim to fully explore the abilities of base large-language models (LLMs) on ARC and how those can be combined in multi-model systems. We introduce a new augmented ARC (AugARC) benchmark tailored towards LLMs, which shows consistently improved performance across all tested LLMs. We show the benefit of fine-tuning LLMs on augmented ARC data. Finally, we built a reflection system based on multiple solvers Bober-Irizar and Banerjee (2024).

AugARC: Augmented ARC for LLMs

The ARC training data can be utilized for fine-tuning LLMs and improving their performance on the evaluation and test sets. One potential issue with this approach is the size of the training set - it contains only 400 samples. Since LLMs have billions of parameters, they usually cannot be effectively trained on smaller datasets and instead require more samples. Therefore, due to its small size, the ARC training dataset limits the ability to fine-tune LLMs for improved broad generalization and reasoning.

Augmented Training Data

To overcome the limited number of ARC training tasks, we propose an augmentation procedure that can significantly extend the training dataset. Our approach expands the ARC training set by applying the following transformations:

- **Rotation:** clockwise rotation of each ARC grid for a given task by 90° or 270°.
- **Flipping:** flips each ARC grid of a task horizontally (along the y-axis) and vertically (along the x-axis).
- **Permutations:** rearranges the sequence of demonstration input-output pairs before the test input grid. We set a threshold for the maximum number of permutations per task to produce datasets of various sizes.

Depending on the transformations applied and the maximum number of permutations applied, the augmented ARC training datasets vary from 2000 up to over 18 million tasks. The AugARC data is available from the following repository: https://github.com/kiril-bikov/AugARC

3-Shot AugARC Benchmark

A key reason for the relatively scarce ARC research on LLMs is the lack of a textual version of the benchmark. The only benchmark suitable for LLMs that resembles Chollet's visual ARC Chollet (2019) is the AI2 Reasoning Challenge Clark et al. (2018); Pătras et al. (2022). AI2 is a multi-choice question answering benchmark that focuses on assessing reasoning. Although AI2 is a more popular and well-established reasoning benchmark for LLMs compared

Dataset Size	Max Permutations
2 000 tasks	-
4 000 tasks	2
5 715 tasks	3
7 430 tasks	4
9 145 tasks	5
18 668 610 tasks	All

Table 1: Size of the augmented ARC training datasets according to the maximum number of permutations. All datasets include 90° and 270° rotations, horizontal and vertical flipping. The augmented datasets range from 2000 to 18 million tasks.

to Chollet's ARC Chollet (2019), the latter is more effective at evaluating broad generalization abilities due to its handcrafted abstract logic.

Identifying that the lack of a textual ARC benchmark is a significant barrier for evaluating LLMs, we create the AugARC benchmark. The AugARC benchmark provides an easy and unified way to evaluate LLMs on 3-shot accuracy on reasoning tasks. In AugARC, each ARC task starts with a textual description explaining the format of the problem. Each ARC grid is represented as a 2D matrix of numbers.

AugARC Input to LLMs The first prediction is based on a normal ARC task, whereas the second and the third ones are 90° and 270° clockwise rotated versions of the same task. The AugARC benchmark is tailored towards LLMs' architecture, as those models process inputs in an auto-regressive, sequential manner. By rotating the ARC tasks, LLMs are presented with a different sequence of numbers (2D matrices) which contain the same abstract logic.

Reproducing ARC Solutions from AugARC Outputs Although the second and third shot in AugARC are based on rotated ARC tasks, the output of the LLMs can easily be transformed back to a solution to the original ARC problem. Once an output is generated by the LLM, it is simply rotated back in an anticlockwise direction. In this way, AugARC only changes the input representation of the ARC problems, but the outputs by the models are then rotated to valid ARC solutions. This process ensures that the results with the proposed AugARC approach are directly comparable with previous ARC attempts.

Method

Fine-tuning LLMs on augmented ARC tasks

Although LLMs have shown impressive capabilities, they can sometimes hallucinate. One potential way to reduce such hallucinations and improve performance on abstract logical tasks is to fine-tune LLMs. Due to the limited size of the ARC training dataset (400 tasks), previous studies have not attempted to train LLMs on ARC. Our augmentation of ARC allows us to overcome this limitation and have sufficient ARC data to fine-tune LLMs.

For efficient training of LLMs, we use Quantized Low-Rank Adaptation (QLoRA) with 4-bit NormalFloat (NF4) quantization (Dettmers et al. 2024). Low-Rank Adaptation constrains the update of a pre-trained weight matrix $W_0 \in \mathbb{R}^{d\times k}$ with a low-rank decomposition $W_0 + \Delta W = W_0 + BA$, where $B \in \mathbb{R}^{d\times r}$, $A \in \mathbb{R}^{r\times k}$, and the rank $r \ll \min(d, k)$ (Hu et al. 2021). During training, W_0 is frozen and does not receive gradient updates, while A and B contain trainable parameters. Both W_0 and $\Delta W = BA$ are multiplied with the same input, and their respective output vectors are summed coordinate-wise (Hu et al. 2021).

Using QLoRA, we fine-tune LLMs on an augmented ARC training dataset consisting of 2000 tasks 1. Due to a significant increase in computational complexity, we avoid fine-tuning the models on some of the bigger augmented ARC training sets from Table 1. For the same reason, we only train LLMs with parameters ranging from 7 to 13 billion.

Reflection System for ARC

A previous promising approach which solves 40.25% of the ARC evaluation tasks combines solutions from different ARC solvers Bober-Irizar and Banerjee (2024). The voting ensemble lacks any "intelligent" analysis of the potential solutions and instead uses a weighting algorithm Bober-Irizar and Banerjee (2024). Therefore, we propose a Reflection System for solving ARC.

The Reflection System relies on models that could have various architectures - LLMs and Program Synthesis solvers. It executes in two main stages, as visualised in Figure 1. In the first stage, each model makes a prediction on the given ARC task. The models work independently and cannot access the outputs of other models. Once the model produces ARC predictions, those are passed in the second stage to the reflection model Lee et al. (2024); Renze and Guven (2024). Conditioned on the given ARC task, the reflection model chooses the prediction from the models that is most likely to be correct.

Experiments

We perform all experiments on the ARC evaluation set which consists of 400 tasks. By design, the ARC evaluation set is significantly more challenging than the training set Chollet (2019). The creator of ARC, François Chollet, emphasised that the performance of intelligent systems should be measured by the fraction of solved tasks on the evaluation set Chollet (2019). Therefore, we perform our experiments on the evaluation set and use 3 shots per task, as set out in the ARC design Chollet (2019).

To present fully reproducible results, all experiments are executed on the complete evaluation set. Some previous solvers have been evaluated on a subset of the ARC evaluation data, making it difficult to understand the true performance of the solver Xu, Khalil, and Sanner (2023); Lei, Lipovetzky, and Ehinger (2024). Our testing approach ensures that future studies could easily use our results for direct comparison with new ARC solvers.

Performance on base ARC and AugARC

We start our experiments with LLMs on the base ARC benchmark, shown in Table 2. The ARC accuracy across 7-

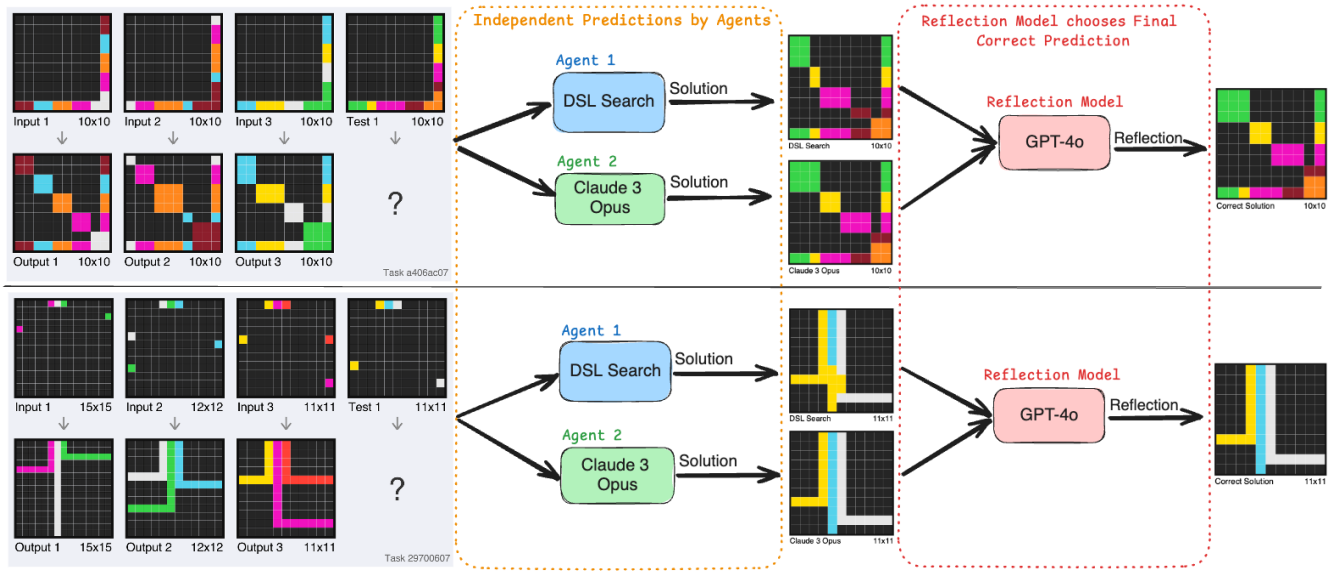

Figure 1: Reflection Systems - execution on two ARC evaluation tasks. Initially, multiple models(LLMs and DSL Search) make independent predictions on the task. Then, the task and the prediction are presented to the reflection model, which chooses the correct final prediction. In the example, model 1 is based on program synthesis (DSL Search) and model 2 is an LLM (Claude 3 Opus). The reflection model is an LLM (GPT-4o). Both task flows are actual demonstration of how our Reflection System configurations perform on ARC evaluation tasks. In both cases, the Reflection System produces correct final solution.

13 billion models ranges from 5 to 9 solved tasks. Bigger LLMs solve slightly more ARC tasks, from 7 to 20, with Gemini Pro achieving the highest accuracy (20).

Model	ARC	AugARC	Increase
Llama-2 7B	5/400	7/400	29%
Mistral 7B	9/400	15/400	67%
Llama-2 13B	5/400	8/400	100%
Llama-2 70B	7/400	14/400	100%
Mixtral 8x7B	9/400	18/400	**125%**
Gemini Pro	20/400	**33/400**	65%

Table 2: Performance of LLMs on ARC and AugARC (on the evaluation set). There is a consistent increase of the accuracy of LLMs when using the AugARC inputs compared to using the base ARC ones (29-125%).

Using the same LLMs, we evaluate the performance on AugARC. For all LLMs, there is a clear accuracy improvement on AugARC compared to the base ARC. The increase varies from 29% for Llama-2 7B up to 125% for Mixtral 8x7B, with the majority of models achieving at least 60%.

The significant improvement in all LLMs on AugARC compared to ARC suggests that changing the grid structure of the tasks for the second and third shot leads to enhanced accuracy. LLMs process the ARC tasks sequentially, and thus are directly influenced by the exact order of the grids. Based on the results, we conclude that the proposed AugARC benchmark is well suited for testing LLMs.

Since AugARC results are directly comparable to ARC, we proceed to use AugARC for the remainder of our experiments.

ARC accuracy across LLMs

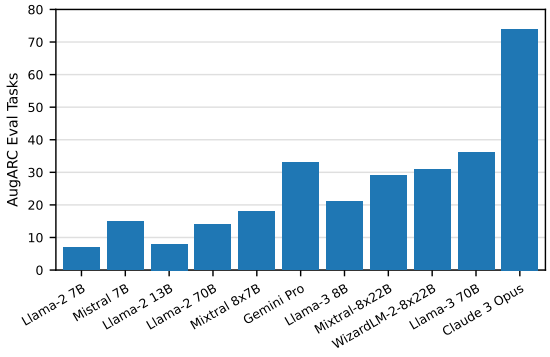

Figure 2: ARC evaluation tasks solved by LLMs. Claude 3 Opus solves the most ARC tasks (74).

The ARC accuracy of LLMs ranges between 7 to 74 solved tasks, as visualised in Figure 2. The best performance by a smaller 7B model is achieved by Llama-3 8B (21). Some bigger open-source LLMs can solve more than 30 ARC tasks, with Llama-3 70B achieving 36. The highest number of solved ARC tasks, 74, is by Claude 3 Opus.

The ARC results demonstrated some variability in performance across LLMs. Bigger models appear to be more accurate on ARC compared to smaller ones. Most LLMs achieve an accuracy in the range of 10-35 tasks, with the only excep-

tion being Claude 3 Opus with 74 out of 400 ARC tasks.

Performance of Fine-tuned LLMs on ARC

To observe whether we can reduce the performance gap between smaller and bigger LLMs on ARC, we fine-tune the 7 and 13B models. All training flows are executed on a single Nvidia A100 80GB GPU.

The results in Table 3 show that the fine-tuned LLMs solve between 18 and 34 ARC tasks. Training benefited all the models substantially - the small fine-tuned Llama-2 7B and 13B models achieved a performance on par with the base versions of significantly bigger models such as Llama-2 70B. After fine-tuning, Mistral 7B outperforms the standard Mixtral 8x7B by 5 correct tasks. The highest result of 34 correct solutions after fine-tuning by Llama-3 8B is impressive, as it outperforms Gemini Pro.

Model	Base	Fine-tuned	Increase
Llama-2 7B	7/400	21/400	**200%**
Mistral 7B	15/400	23/400	53%
Llama-2 13B	8/400	18/400	125%
Llama-3 8B	21/400	**34/400**	62%

Table 3: ARC evaluation results of base and fine-tuned LLMs. The increase column shows the improvement in accuracy from a base LLM compared to its fine-tuned version. All LLMs consistently show improved ARC performance after fine-tuning, ranging from 62% to 200%.

The results in Table 3 demonstrate a significant increase in ARC performance across all fine-tuned LLMs compared to their base versions. The improvement in accuracy after training varies between 53% in Mistral 7B up to 200% in Llama-2 7B. While Llama-2 7B and 13B both achieve more than 100% improvement - 125% and 200% respectively, Mistral 7B and Llama-3 8B improved in the range of 50% to 65%.

Based on our results, we conclude that training small LLMs on an AugARC dataset consistently improves their performance. Notably, fine-tuning smaller LLMs (7-13B parameters) is so effective that it can lead to better ARC performance than significantly bigger base LLMs.

Performance of the Reflection System

We experiment with Reflection System configurations based on two or three models and with different reflection models. We always include the program synthesis solver (DSL Search (Icecuber 2023)) as a solver in all of our reflection system experiments. We also always include the LLM with highest ARC accuracy as a model (Claude 3 Opus). We experiment with base and fine-tuned LLMs for the reflection models (and a potential third model) to find the Reflection System configurations which achieve the highest ARC accuracy.

Table 4 shows that the ARC performance by different reflection system configurations varies between 133 and 166 solved evaluation tasks. In a 2-model setting, with DSL Search and Claude 3 Opus, Llama-3 70B struggles as a reflection model, solving only 133 tasks. GPT-4-turbo and GPT-4o perform significantly better as reflection models, solving 165 and 166 ARC tasks. When adding a fine-tuned Llama-3 8B as a third model, the reflection system solves 163 ARC tasks.

model 1	model 2	model 3	Reflection Model	ARC Correct
DSL Search	Claude 3 Opus	-	Llama-3 70B	133/400
DSL Search	Claude 3 Opus	-	GPT-4-turbo	165/400
DSL Search	Claude 3 Opus	-	GPT-4o	**166/400**
DSL Search	Claude 3 Opus	Fine-Tuned Llama-3 8B	Claude 3.5 Sonnet	163/400

Table 4: Correctly solved ARC evaluation tasks in a 3-shot setting by different Reflection System configurations. The best 2-model performance is with DSL Search and Claude 3 Opus as models and GPT-4o as a reflection model (166). The highest 3-model accuracy adds a fine-tuned Llama-3 8B model (163).

Our best 2-model and 3-model reflection system configurations both outperform the best single LLM, Claude 3 Opus (74), and the best program synthesis approach, which has been tested on the complete ARC evaluation set - the DSL Search (160). Based on the results, we argue that our reflection system is an effective approach for combining LLMs and Program Synthesis solvers into systems for enhanced ARC performance.

Limitations

Since we did not have access to the data used for pre-training the LLMs, we cannot exclude the possibility that some models might have been pre-trained either on ARC tasks or on other very similar abstract problems. It can be argued that the significant improvement after fine-tuning demonstrates that most of the tested LLMs have not been pre-trained on ARC. Nevertheless, the substantially higher ARC results by Claude 3 Opus compared to all other LLMs raise some concerns that this model might have been pre-trained on ARC.

Conclusion

We propose an augmentation procedure for ARC that rotates the tasks 90- and 270-degree clockwise. With the augmented ARC data, we fine-tune LLMs and produce improved results on the reasoning tasks. We also introduce a new AugARC benchmark, which leads to better results for LLMs compared to the normal ARC. Finally, we create a new Reflection System for solving ARC. In future work, AugARC can be extended using more complex data augmentation techniques such as geometric transformations instead of rotations and flipping. Additionally, future studies can attempt to fine-tune LLMs on larger augmented datasets.

References

Bober-Irizar, M.; and Banerjee, S. 2024. Neural networks for abstraction and reasoning: Towards broad generalization in machines. *arXiv preprint arXiv:2402.03507*.

Boden, M.; Bryson, J.; Caldwell, D.; Dautenhahn, K.; Edwards, L.; Kember, S.; Newman, P.; Parry, V.; Pegman, G.; Rodden, T.; et al. 2017. Principles of robotics: regulating robots in the real world. *Connection Science*, 29(2): 124–129.

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

Clark, P.; Cowhey, I.; Etzioni, O.; Khot, T.; Sabharwal, A.; Schoenick, C.; and Tafjord, O. 2018. Think you have solved question answering? try arc, the ai2 reasoning challenge. *arXiv preprint arXiv:1803.05457*.

Dettmers, T.; Pagnoni, A.; Holtzman, A.; and Zettlemoyer, L. 2024. Qlora: Efficient finetuning of quantized llms. *Advances in Neural Information Processing Systems*, 36.

Hu, E. J.; Shen, Y.; Wallis, P.; Allen-Zhu, Z.; Li, Y.; Wang, S.; Wang, L.; and Chen, W. 2021. Lora: Low-rank adaptation of large language models. *arXiv preprint arXiv:2106.09685*.

Icecuber. 2023. ARC: 1st Place Solution. Available at: https://www.kaggle.com/code/icecuber/arc-1st-place-solution/execution. Accessed: 01 August 2024.

Korteling, J. H.; van de Boer-Visschedijk, G. C.; Blankendaal, R. A.; Boonekamp, R. C.; and Eikelboom, A. R. 2021. Human-versus artificial intelligence. *Frontiers in artificial intelligence*, 4: 622364.

Lee, K.; Hwang, D.; Park, S.; Jang, Y.; and Lee, M. 2024. Reinforcement Learning from Reflective Feedback (RLRF): Aligning and Improving LLMs via Fine-Grained Self-Reflection. *arXiv preprint arXiv:2403.14238*.

Lei, C.; Lipovetzky, N.; and Ehinger, K. A. 2024. Generalized planning for the abstraction and reasoning corpus. In *Proceedings of the AAAI Conference on Artificial Intelligence*, volume 38, 20168–20175.

Pătras, C.-B.; Pîrtoacă, G.-S.; Rebedea, T.; Ruşeţi, Ş.; et al. 2022. More with Less: ZeroQA and Relevant Subset Selection for AI2 Reasoning Challenge. *Procedia Computer Science*, 207: 2757–2766.

Renze, M.; and Guven, E. 2024. Self-Reflection in LLM Agents: Effects on Problem-Solving Performance. *arXiv preprint arXiv:2405.06682*.

Shneiderman, B. 2020. Human-centered artificial intelligence: Three fresh ideas. *AIS Transactions on Human-Computer Interaction*, 12(3): 109–124.

Xu, Y.; Khalil, E. B.; and Sanner, S. 2023. Graphs, constraints, and search for the abstraction and reasoning corpus. In *Proceedings of the AAAI Conference on Artificial Intelligence*, volume 37, 4115–4122.