ARC Prize 2024: Technical Report

François Chollet, Mike Knoop, Gregory Kamradt, Bryan Landers

December 5, 2024

Abstract

As of December 2024, the ARC-AGI benchmark is five years old and remains unbeaten. We believe it is currently the most important unsolved AI benchmark in the world because it seeks to measure generalization on novel tasks – the essence of intelligence – as opposed to skill at tasks that can be prepared for in advance. This year, we launched ARC Prize, a global competition to inspire new ideas and drive open progress towards AGI by reaching a target benchmark score of 85%. As a result, the state-of-the-art score on the ARC-AGI private evaluation set increased from 33% to 55.5%, propelled by several frontier AGI reasoning techniques including deep learning-guided program synthesis and test-time training. In this paper, we survey top approaches, review new open-source implementations, discuss the limitations of the ARC-AGI-1 dataset, and share key insights gained from the competition.

1 Introduction: ARC-AGI

François Chollet first wrote about the limitations of deep learning in 2017 (5). In 2019, he formalized these observations into a new definition of artificial general intelligence (AGI), characterizing it as a system capable of efficiently acquiring new skills and solving novel problems for which it was neither explicitly designed nor trained. (7)

Alongside this definition, Chollet published the Abstraction and Reasoning Corpus (ARC) benchmark (6) (later renamed ARC-AGI to avoid name collisions with other AI benchmarks), as a first concrete attempt to measure this definition of intelligence. We will refer to this dataset as ARC-AGI-1. It is a set of independent "tasks" (see figure 1), each consisting of a number of "demonstration pairs" (two or more, with a median count of three) and one or more "test inputs". A test pair consists of an "input grid", a rectangular grid of variable size (up to a maximum size of 30 rows by 30 columns) where each cell can have one of ten distinct "values", and an output grid which should be fully inferable from the characteristics of the input grid. The goal is to use the demonstration pairs to understand the nature of the task, and use this understanding to construct the output grid corresponding to each test input. The test taker is allowed two attempts per test input.

The defining characteristic of the benchmark is that it should not be possible to prepare for any of the tasks in advance. Every task in the dataset follows a different logic. All tasks were created by humans to ensure a high degree of novelty and diversity.

ARC-AGI tasks do not require specialized world knowledge (e.g., historical facts) nor language to solve. The only assumed prior knowledge is Core Knowledge (7) – concepts such as objectness, basic topology, elementary integer arithmetic, etc. Human Core Knowledge has been investigated by Spelke et al. (22). These knowledge priors are acquired by children very early (typically before age four) and are universally shared by all humans. The ARC-AGI-1 public training tasks are designed to expose test-takers to all the Core Knowledge priors needed to solve ARC-AGI tasks.

1.1 Dataset Composition

ARC-AGI-1 consists of 1,000 tasks split into four subsets:

- Public training tasks (400, easy) - Intended to demonstrate the task format and allow for learning the Core Knowledge priors.
- Public evaluation tasks (400, hard) - Intended to let researchers locally evaluate their performance.
- Semi-private evaluation tasks (100, hard) - Intended to let us evaluate third-party approaches that rely on publicly-available commercial APIs. It is "semi-private" because while it hasn't been publicly released, it has been exposed to commercial APIs and thus suffers from a risk of leakage.
- Private evaluation tasks (100, hard) - Intended to let us evaluate standalone approaches. It is fully private and theoretically free of leakage.

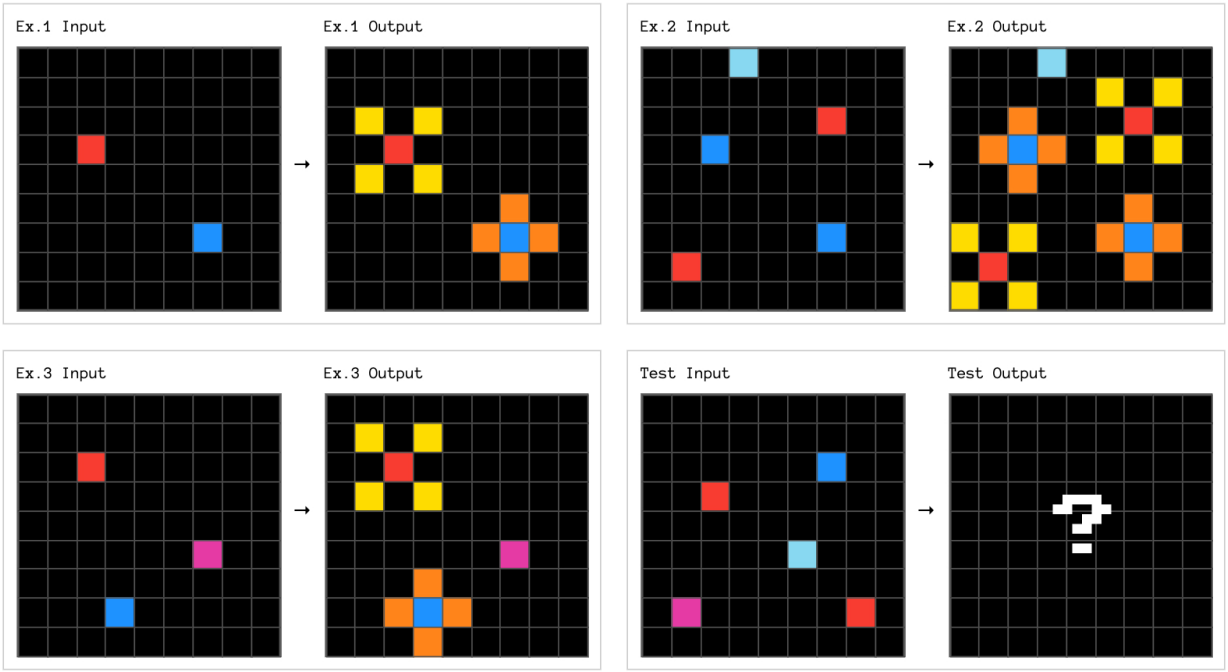

Figure 1: Example ARC-AGI task (0ca9ddb6).

State-of-the-art scores are only reported on the private evaluation task set in order to reduce the risk of overfitting and data contamination.

Another important characteristic of ARC-AGI tasks is that they are hard for AI systems, yet easy for humans. The original private evaluation tasks were originally tested by two people who scored 97% and 98%, and, together, solved all 100%. NYU published a recent study testing Mechanical Turk workers which showed 99% of public evaluation tasks were solved by at least one worker, with 10 workers assigned to each task. (18)

1.2 Pre-2024 Progress

ARC-AGI has been the target of three public competitions before ARC Prize 2024:

- 2020: First ARC-AGI Kaggle competition ($20,000 USD in prizes) (9)
- 2022: ARCathon 1 ($100,000 in prizes) (16)
- 2023: ARCathon 2 ($100,000 in prizes) (17)

In the years following the original release of ARC-AGI-1, pure deep learning approaches turned out to perform poorly on ARC-AGI, as the classic deep learning paradigm works by relating new situations to situations seen at training time, with no adaptation or knowledge recombination at test time, making it impossible for such models to apprehend entirely novel tasks at test time. In the first Kaggle competition (2020), no deep-learning based approach scored above 1%. The original GPT-3 model from OpenAI scored 0% on the public evaluation via direct prompting.

This is why, despite being created before large language models (LLMs), ARC-AGI has resisted the rise of LLMs in the 2022-2024 period.

The first ARC-AGI competition ran on Kaggle in 2020 (9) with a top score of 20%. Four years later, the top score had only increased to 33%. This lack of progress on ARC-AGI can be attributed to lack of progress towards AGI. From 2020 to early 2024, the field of AI research was dominated by the scaling up of deep learning systems, which increased task-specific skills but did not improve the ability to tackle tasks without available training data at training time (i.e., general intelligence). Our view is that progress towards AGI had stalled during this period – AI systems had been getting bigger and memorizing ever more training data, but generality in frontier AI systems had been roughly static.

2 ARC Prize 2024 Results

2.1 Kaggle Leaderboard

The poor performance of frontier AI systems on ARC-AGI at the start of 2024 was clear evidence of the conceptual limitations hindering AGI progress. In response, we launched ARC Prize (15) to inspire AI researchers to work on new ideas and share them openly. Most frontier AI research is no longer being published by industry labs, which is why ARC Prize incentivizes and promotes open sharing.

ARC Prize 2024 launched on June 11, 2024, and ended November 10, 2024. The competition ran both on kaggle.com and arcprize.org. Prizes included a Grand Prize of $600,000 USD for the first team to reach 85% on the private evaluation set, $50,000 in progress prizes tied to the Kaggle leaderboard, and $75,000 in prizes for the best paper submissions. The Grand Prize was not claimed.

The 2024 winners are are shown in Table 1. All scores are open source and reproducible on arcprize.org.

The winners competed on Kaggle, where their solutions attempted to solve the 100 tasks of the private evaluation set on a virtual machine that featured a single P100 GPU in under 12 hours with no internet access. Only those who open-sourced their solution could be named as a winner and claim prizes. MindsAI achieved the highest score of 55.5% on the private evaluation set during the competition but chose not to open source their solution, and was therefore ineligible for a prize.

Place	Name	Score (Private evaluation set)
1st	the ARChitects	53.5%
2nd	Guillermo Barbadillo	40%
3rd	alijs	40%
4th	William Wu	37%
5th	PoohAI	37%

Table 1: ARC Prize 2024 Winners.

2.2 Public Leaderboard

In addition to the Kaggle leaderboard, ARC Prize also featured a secondary leaderboard, ARC-AGI-Pub, which allowed internet access and relaxed compute constraints, in order to evaluate the performance achievable with closed-source frontier models. Due to the risk of data leakage, submissions were not verified on the private evaluation set, but rather on the "semi-private" evaluation set (100 tasks). We report results alongside the public evaluation set (400 tasks) to guard against overfitting. We consider scores overfit if semi-private and public evaluation set scores exceed ±10% absolute difference. All scores are open source and reproducible on arcprize.org.

Name	Semi-private eval	Public eval
Jeremy Berman	53.6%	58.5%
Akyürek et al.	47.5%	62.8%
Ryan Greenblatt	43%	42%
OpenAI o1-preview (pass@1)	18%	21%
Anthropic Claude 3.5 Sonnet (pass@1)	14%	21%
OpenAI GPT-4o (pass@1)	5%	9%
Google Gemini 1.5 (pass@1)	4.5%	8%

Table 2: ARC-AGI-Pub leaderboard.

This leaderboard offered entrants approximately 1,000 times more compute than the Kaggle leaderboard. ARC-AGI-Pub entries were allowed to consume up to $10,000 in API credits, whereas entries on Kaggle could only consume the equivalent of $10 of compute per entry. ARC Prize covered API fees for public leaderboard high score submission verification.

Final 2024 top scores on the public leaderboard are shown in Table 2. The "pass@1" commercial API results use a publicly available direct prompting approach (identical for all models.)

Surprisingly, both the competition and secondary public leaderboard top scores tracked closely. This suggests algorithmic improvements towards AGI hold significant power and that massive compute may not be necessary in order to beat ARC-AGI.

2.3 Paper Awards

ARC Prize 2024 also featured "Paper Awards" to reward novel concepts regardless of how their solutions scored. Prizes were awarded to the following papers. All papers are shared alongside open source code on arcprize.org.

- **First place**: Li et al., *"Combining Induction and Transduction for Abstract Reasoning"*
- **Second place**: Akyürek et al., *"The Surprising Effectiveness of Test-Time Training for Abstract Reasoning"*
- **Third place**: Bonnet and Macfarlane, *"Searching Latent Program Spaces"*
- **Runners up**:
  - Franzen et al., (the ARChitects): *"The LLM ARChitect: Solving the ARC Challenge Is a Matter of Perspective"*
  - Barbadillo, *"Omni-ARC"*
  - Fletcher-Hill, *"Mini-ARC: Solving Abstraction and Reasoning Puzzles with Small Transformer Models"*
  - Ouellette, *"Towards Efficient Neurally-Guided Program Induction for ARC-AGI"*
  - Puget, *"A 2D nGPT Model For ARC Prize"*

In total, 1,430 teams submitted 17,789 entries for ARC Prize 2024. Many well-funded startups have also shifted priorities to work on ARC-AGI – we've heard from seven such companies this year. Additionally, multiple large corporate labs have now spun up internal efforts to tackle ARC-AGI.

While we have a long way to go to reach AGI, we're excited that ARC Prize has catalyzed several new open source frontier AGI reasoning approaches, in particular test-time training, an approach we first observed in use by Jack Cole in 2023 and subsequently popularized this year by ARC Prize.

3 Top Approaches

Until 2024, all top ARC-AGI approaches relied on discrete program search, starting with the 2020 winning entry by icecuber (9), which exclusively leveraged brute-force program search to achieve 20% on the private evaluation set.

Over the next four years, progress was slow. Despite the advent of LLMs (e.g., GPT-3, 3.5, 4), attempts to use these systems to beat ARC-AGI were unsuccessful. Advancement primarily came in the form of improved domain-specific languages (DSLs), notably one created by Michael Hodel (12) which improved the performance of the program search process.

Progress re-accelerated, however, during ARC Prize 2024, catalyzed by three major categories of approaches:

- **Deep learning-guided program synthesis:** Leveraging deep learning models, particularly specialized code LLMs, to generate task-solving programs or guide the program search process beyond blind brute-force methods.
- **Test-time training (TTT) for transductive models[1]:** Fine-tuning an LLM at training time on a given ARC-AGI task specification in order to recombine the prior knowledge of the LLM into a new model adapted to the task at hand.
- **Combining program synthesis together with transductive models:** Merging together the two approaches above into a single super-approach, based on the observation that each approach tends to solve different kinds of tasks.

[1] A "transductive" model is a model that attempts to directly predict the output grid given a test input grid and a task specification, instead of first trying to write down a program that matches the task.

The first exciting accomplishment of the year came from Ryan Greenblatt, who reached 42% on the ARC-AGI-Pub leaderboard (11) using an LLM-guided program synthesis approach. His solution uses GPT-4o to search thousands of Python programs per task (and iteratively debug the most promising ones) to find a program that successfully maps the task input/output examples.

During the 5-month contest period, one of the top scoring teams, MindsAI, improved its score on ARC-AGI-1 private evaluation set from 33% (achieved by the same team at the end of the 2023 competition) to 55.5%. MindsAI pioneered test-time training for ARC-AGI, beginning in 2023. While they chose not to publicly share their 2024 TTT implementation, they inspired many teams to invent their own.

Notably, ARC Prize 2024 1st place winners, the ARChitects, used TTT to score 53.5% on the private evaluation and the 2nd place paper award winners, Ekin Akyürek and team, used TTT to score 47.5% on semi-private evaluation set. Both were open sourced as a result of the competition and are available on arcprize.org.

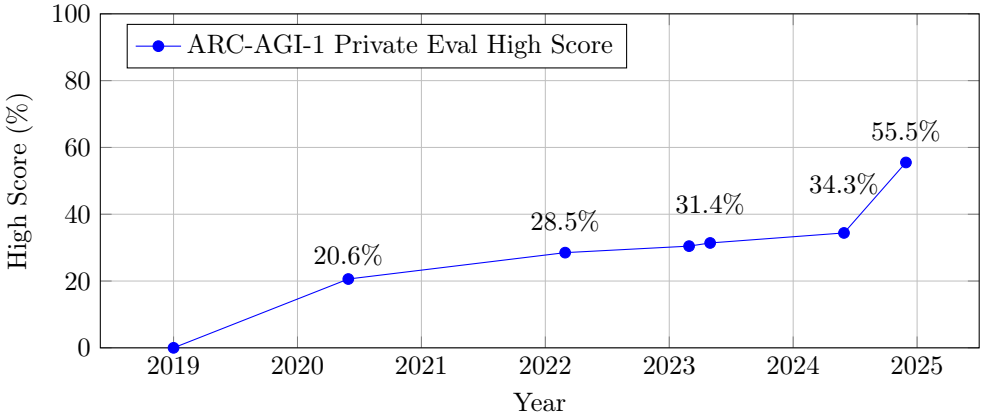

Figure 2: ARC-AGI-1 high scores over time

3.1 Deep Learning-Guided Program Synthesis

Upon releasing ARC-AGI in 2019, Chollet suggested that it could be understood as a program synthesis benchmark, and that it could be solved by leveraging deep learning models to guide a discrete program search process – thereby solving the bottleneck of program synthesis, combinatorial explosion. Such an approach was described in detail in Chollet's AAAI Fall Symposium talk in November 2020 (8). The 2020 competition was entirely dominated by brute-force program search technique, and the rise of LLMs capable of generating code from 2023 onwards led to more efficient program synthesis solutions that used LLMs to write candidate programs that would then be evaluated by a code interpreter.

Program synthesis for ARC-AGI has so far come in the following flavors:

- **Brute-force search over a Domain Specific Language (DSL)**: This approach involves exhaustively searching the space of possible programs within a predefined DSL. While theoretically complete, it cannot scale on its own to complex programs as it suffers from combinatorial explosion as the size of the DSL and the size of the desired program grow. This is the first approach to have yielded positive results on ARC-AGI, and as early as 2020 we had a proof of existence of a very simple, relatively low-compute brute-force strategy that could achieve 49% on the private evaluation set by ensembling all 2020 competition entries together. The highest score seen on any single Kaggle submission with this approach has been 40% on the private evaluation set (by Agnis Liukis, team name alijs.)

- **LLM-powered program generation in open-ended languages**: LLMs pretrained on programming-related data can be used to generate programs in general-purpose languages like Python. Greenblatt (11) demonstrated an approach that prompted GPT-4o with a task description (containing the task's demonstration pairs) to generate thousands of candidate Python programs to solve the task, which were then run by a code interpreter and selected based on their performance on the demonstration pairs. To work well, this approach requires significant prompt engineering efforts and relies on deterministic evaluation of potentially vast numbers of generated programs.
- **LLM-guided discrete program search over a DSL**: This approach combines some of the strengths of both DSL-based discrete program search and LLMs. Ouellette (21) demonstrated this strategy using an LLM to guide the search process within a DSL, effectively reducing the search space and improving efficiency.
- **LLM-powered iterative program debugging**: Instead of generating complete programs from scratch, LLMs can be used to iteratively debug and refine programs that are heuristically generated or close to correct. Greenblatt (11) also explored this approach, using LLMs to identify and rectify errors in the most promising incorrect programs (selected via a heuristic criterion), leading to improved performance on ARC-AGI.

One approach that has not been tried so far (likely because it is technically challenging) but that we expect to perform well in the future, is the use of specialist deep learning models to guide the branching decisions of a discrete program search process – similar to what can be seen in the AlphaProof system from Google DeepMind (2).

We expect that program synthesis, together with closely related test-time search techniques, will be adopted by every frontier AI system over the next 12 to 24 months. This will result in a stronger need for formal efficiency reporting with benchmark scores in the future, since any search-based approach can always score higher if provided with more compute, and it is thus no longer possible to assign a score to an approach, but only to the combination of an approach plus a compute budget. For instance, we estimate that an 85% ARC-AGI score could be achieved by an approach like Greenblatt's when generating, evaluating, and debugging approximately 100,000,000 programs per task, which would represent a multi-million dollar compute budget to solve 100 tasks.

We also note that deep learning-guided program synthesis does not currently decisively beat DSL-based brute-force program search – both score in the 40% range today with comparable compute budgets. We expect that more efficient program search techniques leveraging deep learning should be able to pull away from brute-force search in the future.

3.2 Test-Time Training

The classical deep learning paradigm, exemplified by LLMs from the 2022-2023 period, involves first training a model on a large dataset and then performing inference with a frozen version of the model. However, solving ARC-AGI requires going beyond simple fetching and application of memorized patterns – it necessitates the ability to adapt to the specific task at hand at test time. This has led to the emergence of test-time training (TTT), also known as test-time fine-tuning (TTFT), as a dominant approach in LLM-based solutions for ARC-AGI. Today, all top LLM-based transduction approaches for ARC-AGI leverage TTT, and there does not exist any *static inference*-style transduction solution that scores above 10%. This stark gap highlights the inability of the classical deep learning paradigm to generalize to novel tasks.

TTT, in this context, involves fine-tuning a pretrained LLM on the demonstration pairs of each task instance seen at test time, effectively creating a different variant of the base model for each task. The model is then prompted to directly predict the output grid (transduction).

Interestingly, TTT can be seen as conceptually similar to program search, albeit at the opposite end of the memorization/recombination spectrum. Both involve recombining pre-existing building blocks to solve a task. Program search typically employs deep recombination of a small set of generic programming primitives. TTT, on the other hand, utilizes a vast number of specialized "building blocks" (vector functions found in the weights of the pretrained LLM) and performs shallow recombination through test-time gradient descent.

Key aspects of the application of TTT to ARC-AGI include:

- **Data augmentation and alternative datasets**: Given the limited size of the ARC-AGI-1 dataset, successful TTT relies heavily on increasing the amount of ARC-AGI-like data available for pretraining, either by using alternative datasets like BARC (a dataset of 400,000 additional ARC-like tasks released by Ellis et al. (19), Re-ARC (a programmatic implementation of the ARC-AGI training set allowing infinite sampling of new instances of the 400 training tasks, released by Hodel (13)) or carefully designed data augmentations. The ARChitects (10), for example, introduced novel augmentations and a selection criterion based on the stability of generated solutions under these augmentations.
- **Fine-tuning strategies**: Both LoRA fine-tuning (14) and full fine-tuning have been explored for adapting LLMs at test time. Fine-tuning is performed on augmented demonstration pairs derived from the specific test instance considered.
- **Specialized 2D-aware architectures**: Effective TTT often leverages specialized transformer architectures tailored for visual reasoning. This includes employing 2D attention mechanisms or 2D position encodings (as seen in the Puget/NVIDIA "A 2D nGPT Model for ARC Prize" (4)) to better capture spatial relationships within the input grids.

Examples of TTT used in ARC Prize 2024:

- OmniARC (Barbadillo (3)): A Qwen2.5-0.5B-Instruct model pretrained on multiple program induction tasks and further fine-tuned at test time with augmented ARC data. They combine this with a program synthesis approach for a competitive ensemble.
- Akyürek et al. (1): An 8B parameter model using TTT achieved 53% accuracy on the public evaluation set.
- MindsAI (20): A Salesforce T5 series model pretrained on the public evaluation set and synthetic data, is further fine-tuned at test time on each private task.
- ARChitects (10): A NeMo-Minitron-8B foundation model with extensive data augmentation and a novel selection criterion based on solution stability under augmentations.

A variant of TTT involves searching in the latent space of an LLM as in Bonnet and MacFarlane (4). This approach uses random search and gradient descent to find better program representations within the model's latent space – a novel take on test-time adaptation that is neither fine-tuning nor discrete search.

Overall, we expect test-time training will be the primary technique for LLM-based AI systems to improve performance on tasks outside of what they were pretrained on. It's harder to incorporate TTT into production systems compared to program synthesis, so we expect it won't be productionized for a couple of years – but it, or a derivative, should become popular from 2026 onwards.

3.3 Combining Program Synthesis with Transduction

There are broadly two ways to approach ARC-AGI:

- **Program synthesis, or "induction"**: Based on the demonstration pairs of a test task, find a program or function that appears to turn the input grids into their corresponding output grids, then apply the program to the test input grid(s).
- **Transduction**: Based on the demonstration pairs of a test task and an input grid, directly predict the corresponding output, for instance, by prompting an LLM with the task description and the test input.

As soon as transduction-based approaches started scoring above zero (in late 2023, pioneered by Jack Cole) researchers noticed that program search and transduction were able to solve significantly distinct sets of tasks. This issue was later investigated in detail by Li et al. in "Combining Induction and Transduction for Abstract Reasoning" (19).

Today, all top scores (e.g., Akyürek and Berman on the public leaderboard, the ARChitects, Barbadillo, and MindsAI on the Kaggle leaderboard) use a combination of transduction and induction. The best transduction-only and induction-only single submissions score around 40%, so only an ensemble of both can compete for the state of the art.

4 Future

We have committed to running ARC Prize annually until the ARC-AGI benchmark is defeated and a public reference solution is shared. ARC Prize 2024 was a large-scale experiment that we consider highly successful, and we aspire to grow ARC Prize from its experimental origins into a durable north star for AGI. We are applying lessons learned during ARC Prize 2024 to inform future versions of both the competition and the benchmark.

4.1 ARC Prize: 2025 and Beyond

We're excited that ARC Prize catalyzed significant attention towards new ideas for AGI. We had expected ARC Prize to drive academics, independent researchers, and big labs alike to give renewed attention towards ARC-AGI. But we were surprised to the degree that well-funded AI research startups would change their roadmaps to prioritize beating the benchmark. We are now aware of at least seven distinct efforts to solve ARC-AGI by organizations that have greater than $1M in funding, including Basis AI (`basis.ai`), Tufa Labs (`tufalabs.ai`), Agemo (`agemo.ai`), and Symbolica (`symbolica.ai`).

All of these groups do not share the same incentives (e.g., progress prizes would not be sufficient to incentivize sharing for venture-funded startups and large corporate labs), and we plan to redesign the 2025 edition of the competition to account for this. Our goal is to provide the best directional compass towards AGI across the full spectrum of AI research actors, from academic labs to startups to big labs.

4.2 ARC-AGI-2

The ARC-AGI-1 private evaluation set has been unchanged since 2019, and it is known to suffer from a number of flaws. First of all, the private evaluation set is limited to only 100 tasks. These 100 tasks have been used by all four ARC-AGI competitions for reporting intermediate leaderboard scores, and, as a result, on the order of 10,000 private evaluation set scores have been reported to participants so far. This presents a significant risk of overfitting, since each score has the potential to extract a tiny but non-zero amount of information about the content of the hidden tasks. The benchmark's reliability can be improved by increasing the sample size and using two separate datasets: one for intermediate leaderboard scores (a larger

semi-private evaluation set) and another for final scoring (a larger private evaluation set). This approach eliminates the risk of overfitting to the private evaluation set.

Further, while 20% represented the highest score by any single submission in 2020, an analysis of all 2020 submissions found that 49% of the private evaluation set was solved by at least one team (all of which were using some variation of brute-force program search.) This suggests that a large fraction of the ARC-AGI-1 dataset is overly susceptible to these kinds of brute-force program search techniques and therefore does not carry a useful signal towards AGI. A sufficient fraction of the dataset has proven interesting enough, and that's why ARC-AGI remains unsolved.

Finally, anecdotal evidence suggests that the different evaluation datasets are not drawn from a consistent human difficulty distribution, which makes score comparisons across evaluations challenging.

To address these issues while retaining the familiar task format of ARC-AGI, we are actively working on ARC-AGI-2 and our intention is to launch the new dataset along with the 2025 competition.

5 Conclusion

ARC Prize 2024 was a highly successful experiment – awareness of the benchmark was raised significantly and several new approaches have emerged, bringing the state of the art from 33% to 55.5%. However, ARC-AGI remains undefeated – still by a considerable margin – especially considering that a score of 49% was technically achievable with basic brute-force program search as early as 2020. New ideas are still needed to build AGI. The fact that ARC-AGI survived five months of intense scrutiny with an outstanding $600,000 grand prize and hundreds of thousands of dollars in additional prizes is strong evidence that the solution does not yet exist. We're inspired, proud, and hopeful that ARC-AGI has played an important role in shifting attention towards new research ideas. Our belief is the team that will eventually build AGI is thinking about ARC-AGI today, and we're committed to stewarding this attention as a north star towards AGI.

6 Appendix

6.1 The ARC-AGI Ecosystem

ARC Prize has inspired a community of researchers and developers who have contributed valuable tools, datasets, and repositories to support both competition participants and the greater AI community.

- **ARC-DSL** - A domain-specific language for working with ARC-AGI tasks: GitHub Repository
- **ConceptARC** - Benchmark in the ARC-AGI domain that systematically assesses abstraction and generalization abilities on a number of basic spatial and semantic "concept groups": GitHub Repository
- **RE-ARC** - A repository to procedurally generate examples for the ARC-AGI training tasks: GitHub Repository
- **BARC** - Tools for generating synthetic ARC-AGI problems: GitHub Repository
- **arcsolver** - A Python library for automatically solving ARC challenges using Claude and object-centric modeling: GitHub Repository
- **ARC Interactive** - An interactive web-based tool for engaging with ARC-AGI tasks: Web Tool
- **Arckit** - Python and command-line tools for easily working with the Abstraction & Reasoning Corpus: GitHub Repository

- **The ARC Game** - UI interface for ARC-AGI Tasks: Web Interface
- **ARC Gym** - A data generation framework to help research and develop a solution to the problems of compositional generalization and efficient search: GitHub Repository
- **Open Source Kaggle Notebooks** - Hundreds of publicly shared Kaggle submissions for ARC Prize 2024: Kaggle Competition Code

Numerous other resources are listed in the official ARC Prize Technical Guide.

6.2 Acknowledgments

ARC Prize builds on the legacy of earlier ARC-AGI competitions: the 2020 competition on Kaggle as well as the 2022 and 2023 "ARCathons", which were a collaboration between François Chollet and the Davos-based non-profit AI lab, Lab42. We are grateful to the Lab42 team – Rolf Pfister, Oliver Schmid, and Hansueli Jud – for their expertise and support in facilitating a smooth transition for the ARC-AGI community. Their contributions were significant in facilitating a bigger, bolder competition and advancing initiatives like ARC-AGI-2.

We would also like to recognize the dedication of past ARCathon participants, who not only championed the benchmark but whose prior work brought new members of the community quickly up to speed. In particular, we thank Michael Hodel, Jack Cole, Mohamed Osman, and Simon Ouellette for their ongoing efforts to develop ARC-AGI solutions. Special recognition is due to Simon Strandgaard for his exceptional role as a community ambassador and prolific open-source contributor.

Finally, we extend our deepest gratitude to all participants in ARC Prize 2024, especially those who shared their work with the community. Your dedication advances the broader field of AI, bringing us closer to realizing the transformative potential of AGI for humanity.

References

1. Ekin Akyürek, Mehul Damani, Linlu Qiu, Han Guo, Yoon Kim, and Jacob Andreas. The Surprising Effectiveness of Test-Time Training for Abstract Reasoning. https://arxiv.org/abs/2411.07279, 2024.
2. AlphaProof and AlphaGeometry teams. AI achieves silver-medal standard solving International Mathematical Olympiad problems. https://deepmind.google/discover/blog/ai-solves-imo-problems-at-silver-medal-level/, July 2024. Accessed: 2024-12-04.
3. Guillermo Barbadillo. Solution Summary for ARC24. https://ironbar.github.io/arc24/05_Solution_Summary/, 2024. Accessed: 2024-11-27.
4. Clément Bonnet and Matthew V. Macfarlane. Searching Latent Program Spaces. https://github.com/clement-bonnet/lpn/blob/7f86b1d11ea37ba173700dbac8604393eac6da37/paper.pdf, 2024. Proposes Latent Program Network (LPN), an algorithm for program induction applied to the ARC-AGI benchmark.
5. François Chollet. *Deep Learning with Python*. Manning Publications Co., Shelter Island, NY, 2017.
6. François Chollet. Abstraction and Reasoning Corpus for Artificial General Intelligence (ARC-AGI). https://github.com/fchollet/ARC-AGI, 2019.
7. François Chollet. On the Measure of Intelligence. https://arxiv.org/abs/1911.01547, 2019.

8. François Chollet. AAAI Fall Symposium Talk. "Abstraction & Reasoning: What Deep Learning can do, what it can't, and what we can try next". https://docs.google.com/presentation/d/1yqE5AYS4l9mzn17WjnSKEmvqydgw-CsAYrs_TPClOFA, 2020.

9. François Chollet, Katherine Tong, Walter Reade, and Julia Elliott. Abstraction and Reasoning Challenge. https://kaggle.com/competitions/abstraction-and-reasoning-challenge, 2020. Kaggle.

10. Daniel Franzen, Jan Disselhoff, and David Hartmann. The LLM ARChitect: Solving the ARC Challenge Is a Matter of Perspective. https://github.com/da-fr/arc-prize-2024/blob/main/the_architects.pdf, 2024. Achieved 53.5 points on the hidden test set and solved 72.5/100 tasks in the public evaluation set during ARC-Challenge 2024.

11. Ryan Greenblatt. Submission for ARC Prize. https://www.kaggle.com/code/rgreenblatt/rg-basic-ported-submission?scriptVersionId=184981551, 2024. Achieved 42% on the public evaluation leaderboard using a LLM-guided program synthesis approach.

12. Michael Hodel. ARC-DSL: A Domain-Specific Language for Solving ARC Tasks. https://github.com/michaelhodel/arc-dsl, 2024. Contains a DSL for solving ARC tasks, proof-of-concept solvers, and a detailed write-up.

13. Michael Hodel. RE-ARC: Reverse-Engineering the Abstraction and Reasoning Corpus. https://github.com/michaelhodel/re-arc, 2024.

14. Edward J. Hu, Yelong Shen, Phillip Wallis, Zeyuan Allen-Zhu, Yuanzhi Li, Shean Wang, Lu Wang, and Weizhu Chen. LoRA: Low-Rank Adaptation of Large Language Models. https://arxiv.org/abs/2106.09685, 2021.

15. Mike Knoop, François Chollet, Bryan Landers, and Greg Kamradt. ARC Prize. https://arcprize.org/, 2024. Accessed in 2024. Hosts: Mike Knoop, François Chollet. Operations: Bryan Landers, Greg Kamradt.

16. Lab42. ARCathon 2022. https://lab42.global/past-challenges/2022-arcathon/, 2022.

17. Lab42. ARCathon 2023. https://lab42.global/past-challenges/2023-arcathon/, 2023.

18. Solim LeGris, Wai Keen Vong, Brenden M. Lake, and Todd M. Gureckis. H-ARC: A Robust Estimate of Human Performance on the Abstraction and Reasoning Corpus Benchmark. https://arxiv.org/abs/2409.01374, 2024.

19. Wen-Ding Li, Keya Hu, Carter Larsen, Yuqing Wu, Simon Alford, Caleb Woo, Spencer M. Dunn, Hao Tang, Michelangelo Naim, Dat Nguyen, Wei-Long Zheng, Zenna Tavares, Yewen Pu, and Kevin Ellis. Combining Induction and Transduction for Abstract Reasoning. https://arxiv.org/abs/2411.02272, 2024.

20. led by Jack Cole MindsAI Team. MindsAI Submission for ARC Challenge 2024. https://www.kaggle.com/jcole75, 2024. ARC Challenge 2024 submission by the MindsAI team.

21. Simon Ouellette. Towards Efficient Neurally-Guided Program Induction for ARC-AGI. https://drive.google.com/file/d/1sFlK3mhz8kH2agdE379o0ODQWYkrSD0b/view, 2024. Explores neurally-guided program induction techniques for the ARC-AGI benchmark.

22. Elizabeth S. Spelke and Katherine D. Kinzler. Core knowledge. *Developmental science*, pages 89–96, 2007.