ARC-AGI-3: A New Challenge for Frontier Agentic Intelligence

ARC Prize Foundation *

March 24, 2026

Abstract

We introduce ARC-AGI-3, an interactive benchmark for studying agentic intelligence through novel, abstract, turn-based environments in which agents must explore, infer goals, build internal models of environment dynamics, and plan effective action sequences without explicit instructions. Like its predecessors ARC-AGI-1 and 2, ARC-AGI-3 focuses entirely on evaluating fluid adaptive efficiency on novel tasks, while avoiding language and external knowledge. ARC-AGI-3 environments only leverage Core Knowledge priors and are difficulty-calibrated via extensive testing with human test-takers. Our testing shows humans can solve 100% of the environments, in contrast to frontier AI systems which, as of March 2026, score below 1%. In this paper, we present the benchmark design, its efficiency-based scoring framework grounded in human action baselines, and the methodology used to construct, validate, and calibrate the environments.

1 The ARC-AGI benchmark series

1.1 ARC-AGI-1 and 2

In 2019, the Abstraction and Reasoning Corpus (ARC-AGI-1) was introduced alongside the paper "On the Measure of Intelligence." (*8*) The paper proposed a formal framework for evaluating general intelligence as skill-acquisition efficiency rather than task-specific performance. ARC-AGI-1 tests fluid intelligence through grid-based tasks where a pattern must be inferred using limited data. The test taker must discover a novel transformation rule from just a handful of input-output examples. Each task presents pairs of grids of up to 30x30 cells using 10 unique colors, grounded in Core Knowledge priors (*20*) such as objectness and basic geometry. No prior knowledge or previously learned heuristics help in solving ARC-AGI-1.

ARC-AGI-1 was built to resist the memorization-and-retrieval shortcuts that had allowed AI to claim superhuman performance on other tasks like Go (*17*). Each task is unique, which rules out memorization. The small number of examples in each task prevents statistical pattern-matching through massive amounts of training data. This combination made ARC-AGI-1 a durable AI benchmark between 2019 and 2024.

ARC-AGI-2 was introduced in March 2025 to measure complexity scaling of AI reasoning on static tasks. It kept the same grid-based format, while requiring deeper levels of reasoning, featuring multi-step reasoning,

*Development team:
Lead game designer: Hunter Henry
Engineering: David Wexler, Derek Smith
Game development: Benjamin Morgan, Vadym Andriianov, Fraser Scott, Pablo Romero Saavedra, Jonathan Pappas, Flynn Swainston-Calcutt, Tom Elliot, Kevin Johnson
Design and communications: Bryan Landers
President, Board member: Gregory Kamradt
Co-founder, Board member: Mike Knoop
Benchmark designer, Co-founder, Board member: François Chollet

sequential rule application, and symbolic interpretation. Every task was human-calibrated with over 400 untrained participants to ensure 100% solvability. On average, a task from ARC-AGI-1 takes humans about 30 seconds to solve, while a task from ARC-AGI-2 takes humans about 300 seconds to solve.

1.2 Prior competitions

The first formal ARC-AGI competition was the 2020 Kaggle Abstraction and Reasoning Challenge (*10*), which used ARC-AGI-1. It offered a $20,000 prize pool and drew 913 teams. The winning solution achieved approximately 20% accuracy on the private test set using brute-force program search over a library of hand-crafted primitives. This approach would come to define the dominant style of ARC-AGI-1 solvers for the following three years. In 2022 and 2023, Lab42 (*15*) hosted two "ARCathon" competitions (*16*) with $100,000 in prizes each, expanding international participation.

In 2024, the ARC Prize Foundation (*3*), co-founded by Mike Knoop and François Chollet, launched the ARC Prize 2024 competition with over $1 million in prizes. The competition drew 1,430 teams and 47 paper submissions. For the first time, it saw strong performance from deep learning based solutions. Test-time training emerged as a breakthrough technique, reaching a score of 53.5% on the private ARC-AGI-1 test set.

The ARC Prize 2025 competition ran on the ARC-AGI-2 benchmark, released in March 2025. It drew 1,455 teams and 90 paper submissions. NVIDIA's NVARC team took first place (*19*) with 24% accuracy, using synthetic data generation and test-time training on a 4B parameter model. The 85% grand prize threshold remained unclaimed across both competition years.

1.3 ARC-AGI-1 and 2 key findings

1.3.1 Predictive power

The Transformer architecture (*21*), published in 2017, paved the way for the "scaling era" of AI, eventually enabling self-supervised LLMs (Large Language Models) for which benchmark performance kept increasing with no further architecture changes, purely by increasing training data and training compute. This is the approach known as *pretraining scaling*, which was the dominant paradigm of AGI research from 2019 to 2024. Chain-of-Thought prompting was discovered in 2022 (*23*) as a way to improve test-time reasoning abilities of LLMs and led to a second scaling paradigm using test-time adaptation (or test-time compute).

ARC-AGI-1 resisted pretraining scaling because base LLMs (without test-time adaptation) are limited to memorization and interpolative retrieval of patterns found in their training data and cannot generalize to never-seen-before tasks, even when these tasks are elementary. The latest generation of base LLMs (as of March 2026) still perform poorly on ARC-AGI-1.

Test-time reasoning was the key innovation that enabled LLM-based systems to begin exhibiting non-zero fluid intelligence, giving rise to the LRM (Large Reasoning Model) paradigm. This was first demonstrated by OpenAI's breakthrough o1 and o3 systems (*9*) on ARC-AGI-1. It was at the time the only benchmark to precisely identify the advent of frontier AI fluid reasoning.

Modern-day models excel at reasoning, precisely corroborated by ARC-AGI-2 progress. These new capabilities have enabled systems to achieve considerable product-market fit with coding tools (such as Claude Code and Codex). There is now global recognition that AI agents capable of broad task automation will transform software engineering and eventually most of the economy. The capstone achievement of the ARC-AGI-1 and 2 benchmarks was signaling and quantifying the arrival of these transformative advances from late 2024 to late 2025 (see figure 1).

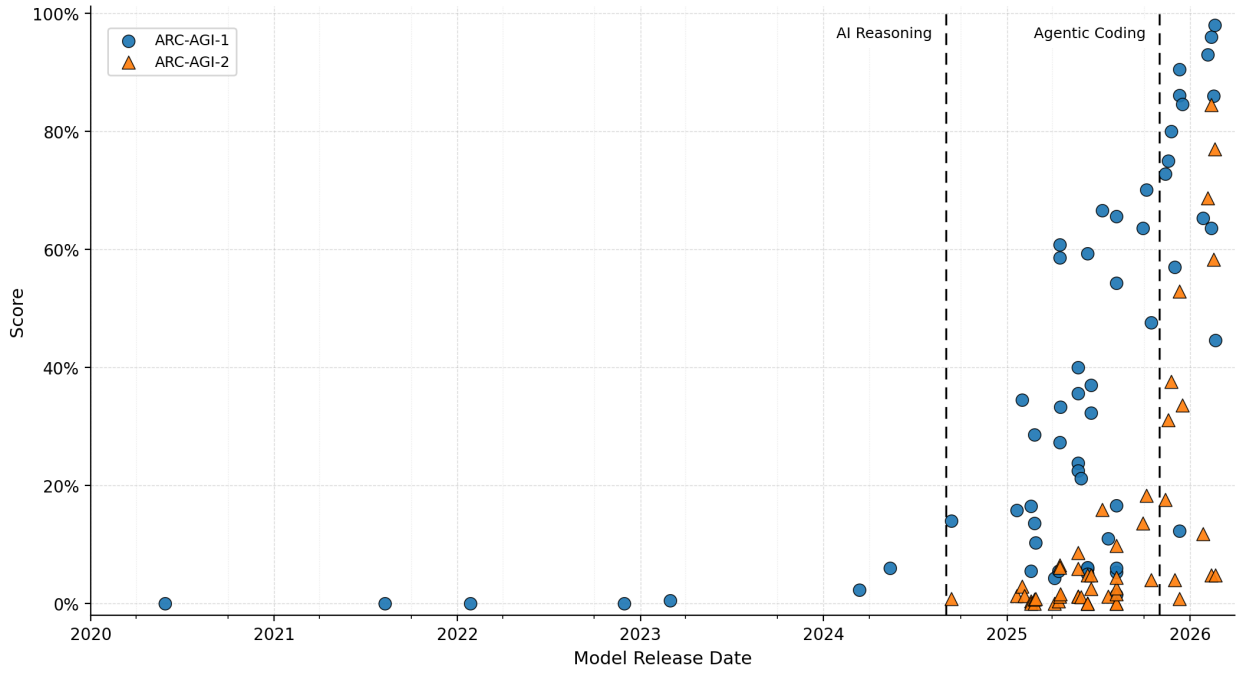

Figure 1: Frontier AI performance on ARC-AGI since introduction in 2019.

1.3.2 Known limits of LRM fluid intelligence

Modern LRMs enable automation in any domain where the following is true:

- The base model contains sufficient knowledge coverage of the domain.
- The domain provides an exact correctness feedback measure to the LRM (such domains are known as "verifiable domains").

This strongly suggests that AI reasoning capability is tied to LRM knowledge. Take a moment to appreciate how strange this is: Human reasoning capability is *not* bound by domain knowledge. This leads to imprecise descriptions of LLMs as "jagged intelligence", when in reality LLMs remain bound to task-specific training, albeit now over task-specific *reasoning chains* instead of the literal task data.

Collecting domain knowledge and building verifiers is expensive, as seen with programming environments for AI coding agents. Future LLM automation of new areas will be driven by industry investment, first in domains where it is possible to assemble the necessary data, compute, and labor and where the return on investment is positive. This includes investment to produce new scientific knowledge. In late 2025, Steve Hsu published an example (*13*) of an LRM automation system that discovered a novel result in quantum physics. Many scientific domains, like drug discovery, are highly automatable due to their mechanistic nature.

However, many other potential applications of LRM automation are either too expensive or impractical to deploy today. As model efficiency increases, more applications become possible. But in the bigger picture, machines that can perform highly efficient adaptation to produce paradigm-shifting innovation are still well outside our reach. Even the latest LRMs remain bottlenecked by human intelligence and show limited ability to cover novel domains, which is a key argument why they fall short of AGI.

1.3.3 Overfitting and memorization shortcuts

Classically, models are said to be overfit when they learn "too much" at training time – memorizing specific features of training data instances instead of learning general, causal principles that will generalize to new instances. This leads the model to perform well during development (via non-generalizable "memorization shortcuts"), while performing poorly on production data after deployment, which often comes as a surprise to model developers. With many benchmarks, the test data shares a high degree of similarity with the training data, leading such overfit models to perform well on the test data despite having limited power to generalize to new instances in the wild. This can also happen when the training process leaks information about the test data.

ARC-AGI-1 and ARC-AGI-2 were designed to be resistant to this style of memorization shortcuts, by using a private dataset (inaccessible to model developers) for official scoring and verification and making sure all tasks were reasonably unique – both distinct from each other and distinct from tasks found on the web.

However, frontier LRMs demonstrate non-zero fluid intelligence and can thus adapt to tasks further away from their training distribution (while still requiring extensive domain knowledge). This means that benchmarks that were designed to resist direct memorization can now be attacked via higher-level shortcuts if the public training set and the private test set are overly similar (e.g., identically distributed) and the model was trained on an enormous amount of tasks representing a dense sampling of the task space – possibly tasks that were automatically generated for this purpose. A simple strategy to achieve this is to ask a model to generate more tasks from the domain, solve them, verify the solutions via a reliable verifier (to note, ARC-AGI-1 and 2 tasks are verifiable), then train on the produced reasoning traces, in a loop. This can scale to millions of tasks – enough to provide high coverage density of the target domain at training time, thus considerably reducing or outright removing the need for test-time adaptation.

We believe this has happened to ARC-AGI-1 and ARC-AGI-2 with frontier LRMs – either incidentally or intentionally. Here is one bit of evidence from our Gemini 3 verification (7). The model mentions the following in its reasoning chain:

> *... Target is Green (3). Pattern is Magenta (6) Solid. Result: Magenta Square on Green ...*
> (Gemini 3 Deep Think)

Our verification model prompt *does not* mention "ARC-AGI" or the integer-to-color mapping used by ARC-AGI tasks, yet the model is using the correct color mapping in its reasoning. This strongly suggests that ARC-AGI data is well represented in the underlying model – enough to make correct ARC-AGI inferences based on just the structure and format of 2D arrays of integers.

Going forward, benchmark designers will need to steer private datasets to be out-of-distribution (OOD) from any publicly available demonstration data if they want to test true generalization.

2 ARC-AGI-3

2.1 ARC-AGI-3 goals

The overarching goal of the ARC-AGI series is to measure the "residual gap" between current artificial intelligence and human-level AGI. This gap is definitionally a moving target and necessitates new versions as frontier AI capabilities advance. We define AGI not merely as a set of static capabilities, but as a system's ability to acquire any skill a human can, as efficiently as a human can. While ARC-AGI-1 and 2 focused on data-efficient modeling (inferring rules from static input/output pairs), ARC-AGI-3 shifts to targeting

agentic intelligence. Specifically, ARC-AGI-3 uses a set of **interactive turn-based environments** to evaluate a test-taker across four core functional components of agentic intelligence:

- **Exploration:** In real-world environments, information is rarely provided passively, it must be actively obtained by the agent by interacting with its surroundings.
- **Modeling:** Inherited from previous ARC-AGI generations, this is the ability to turn raw observations into a generalizable world model that can predict future states and outcomes.
- **Goal-Setting:** A cornerstone of autonomy, goal-setting is the ability to identify interesting or desirable future states without explicit instructions. The agent must independently determine "what to target" based on its own intrinsic drive and environmental cues.
- **Planning and Execution:** This involves the strategic mapping of an action path from the current state to the identified goal. It requires not only initial accuracy but also the agility to course-correct in response to environmental feedback or unexpected results.

While causal modeling remains a prerequisite, the benchmark now demands autonomous navigation of "unknown unknowns". In particular, one of the most significant hurdles in ARC-AGI-3 is that the agent is **never told the objective nor provided instructions**. It must autonomously infer the mechanics of each new environment, including the win conditions.

2.2 Intelligence as efficiency

In the ARC-AGI-3 framework, intelligence is fundamentally defined as **efficiency** across the four pillars mentioned above. A high-intelligence system is not simply one that can solve a task, but one that does so while minimizing its resource usage. There are many forms of resources one might consider, such as data (number of states explored), time, compute, and risk (in embodied or game-based scenarios, every action carries a cost: potential "death," loss of progress, or wasted energy). ARC-AGI-3 makes the opinionated decision of subsuming all of these into a single scalar efficiency measure: **action efficiency**. Action efficiency lets us provide a standardized comparison between biological and artificial agents.

Action efficiency is the number of moves or "turns" required to solve a new environment upon first contact with it. It is aggregated on a per-level basis (more in the Scoring Methodology section). This metric has the following useful properties:

1. **It penalizes "brute-forcing":** A system that blindly tries many options is viewed as less intelligent than one that quickly forms a model of the environment and uses it for effective planning and execution.
2. **It accounts for data efficiency and risk efficiency:** Fewer actions naturally translate to lower exposure to environmental hazards.
3. **It enables direct human-AI comparison:** By establishing a baseline of human action efficiency on these same environments, we can quantitatively measure how close an AI is to "human-level" skill acquisition.

Beating ARC-AGI-3 is achieved when an AI system matches or exceeds human-level action efficiency on ARC-AGI-3 environments that it sees for the first time, averaged across all private environments.

2.3 The ARC-AGI-3 environment format

Each environment in ARC-AGI-3 is structured as a series of levels. A level ends when a win condition (terminal frame) is reached. The benchmark utilizes a turn-based interface designed to prioritize offline reasoning over real-time sensorimotor "reflexes." At each turn, the agent is presented with a frame (or series of frames representing a transition animation), and must take one action to move to the next frame. The environment's state does not change asynchronously from the agent's actions.

Figure 2: Screenshot of ARC-AGI-3 environment `ls20`.

2.3.1 The Observation Space

The agent views a **64x64 grid** where each cell is one of **16 possible colors**. A given grid state is called a "frame". At each turn, the agent receives a frame or frame sequence. Frame sequences allow for non-interactive animations (e.g., an object moving across the screen) between player turns.

2.3.2 The Action Space

Each environment offers a different action space, which is a subset of:

- Five key actions, plus an Undo action (reverting to the previous state)
- One action to select (e.g. click on) a cell from the 64x64 grid by specifying its coordinates

This small action space ensures that the complexity of the benchmark lies in the logic of the environment, not the difficulty of the controls.

An action is defined as a discrete interaction with the environment, i.e., a turn where the agent submits a command, move, or input that affects the environment state. Internal operations that do not alter the environment, such as tool calls, reasoning steps, or retries within the model itself, are **not counted** as actions.

3 Building ARC-AGI-3

3.1 The ARC-AGI-3 game environment studio

To create ARC-AGI-3, we established an in-house game studio tasked with producing a collection of novel interactive environments under a shared set of technical and design constraints. This studio model allowed creative environment design to be coupled with standardized interfaces, evaluation procedures, and validation criteria which enabled environment production to scale without sacrificing consistency.

3.2 Production pipeline

In order to make production efficient, we organized the studio around three core functions: a lead developer who defined and reviewed the production pipeline, individual developers who implemented environments, and an engineer who built high-level automation and internal tools to support creation at scale.

Our initial assumption was that environments could be developed serially by each environment developer, with one environment completed before the next entered production. In practice, this did not reflect the realities of development. Because ideation, implementation, playtesting, and revision progressed at different rates, throughput was highest when three to four environments were in development simultaneously by a single developer, at different stages of the pipeline.

The production pipeline comprised four stages:

1. **Specification:** The developer creates an environment concept description, which is collectively reviewed before implementation, allowing major design issues to be identified early and reducing iteration cost.
2. **Internal:** The developer builds a prototype and tests it with members of the team.
3. **External:** The environment undergoes external human testing in order to determine whether it satisfied our human-performance criteria. Environments that passed this stage were moved on.
4. **Done:** The environment is finished and ready for sorting into one of the ARC-AGI-3 sets.

Throughout the process, automated validation was used to detect development bugs, trivial environments, and regressions before an environment advanced.

3.3 Technical constraints

All ARC-AGI-3 environments were implemented in a shared runtime using a custom in-house environment engine. We initially used Unity for this process but found it too heavy and too slow for the rate of iteration

required. Building a custom engine provided tighter control over performance, tooling, and evaluation. The final engine is implemented in Python (*6*) to achieve our minimum performance goal threshold of 1,000 frames per second.

3.4 Game design constraints

The core challenge in ARC-AGI-3 is intended to be reasoning, rather than perception, which is why ARC-AGI-3 is turn-based instead of real-time. Idea generation for new environments, rather than implementation, was often the most challenging part of the development process. Below, we present our core environment design principles.

Core knowledge priors only: To ensure the benchmark remains a test of innate reasoning rather than acquired knowledge, all environments are strictly limited to Core Knowledge priors (*20*), and seek to avoid similarities with existing games.

- **Objectness:** Elements are perceived as coherent, persistent entities that can move, collide, or be occluded.
- **Basic geometry and topology:** Understanding of symmetries, rotations, and elementary topology (e.g., "inside" vs. "outside", connectedness, holes).
- **Basic physics:** Intuitive rules like gravity, momentum, and bouncing.
- **Agentness:** Recognizing that certain objects act with intent and pursue goals.
- **No language or cultural symbols:** Environments never use numbers, letters, recognizable real-world clip-art (like flowers or keys), or cultural conventions (like green meaning "go").

Novelty: Each environment is required to be novel both with respect to preexisting video games, and with respect to the previously created set of environments. As a practical test of novelty, we tested whether a single program could solve two different environments while being at least 50% shorter than the concatenation of two independent solution programs; when the answer is yes, those environments are likely insufficiently distinct.

Human solvable: Environments are designed to be solvable by humans within a bounded play session of approximately 20 minutes (but most environments can be solved in only a few minutes).

Difficulty through composition: Difficulty is not intended to arise from obscurity or increasing complexity. Rather it is intended to arise from the composition of reasoning demands acquired over the course of play. Later levels are therefore expected to require the accumulation and integration of concepts learned earlier in the environment.

Tutorial level: The first level in an environment functions as a tutorial level and is intentionally easy (to both humans and AI). In some cases, random agents can occasionally stumble into success at this stage, which is acceptable by design. The purpose of the opening level is to communicate the core interaction pattern and orient the player without instructions.

Multiple mechanics: Each environment contains multiple mechanics. Environments centered on a single mechanic that scaled in size or difficulty are treated as an anti-pattern.

Levels: Environments are developed with a level-based structure, with at least six levels per environment.

Game ID: Each environment has a unique game ID that is a series of four characters. Informally, environments have longer names, but these are not shared publicly to avoid sharing semantic information about the environment's goal or mechanics.

3.5 Automated environment validation

The validation pipeline is divided into two complementary layers: **deterministic system qualification** and **exploratory state-space analysis**. Together, these provided confidence that an environment was both compatible with the platform and behaviorally well-formed under large-scale automated execution.

3.5.1 Environment qualification

Qualification focuses on platform integration and reliability. Structural tests verified each environment can be loaded, instantiated, and exercised by the broader runtime environment. Several regimes are deployed.

The first random regime runs for up to 50,000 steps and asserts that no level can be beaten by accident. This is primarily a sanity check against trivial or degenerate reward paths.

The second regime extends to 1,000,000 steps and strengthened the constraint that non-tutorial levels must remain unbeaten under uninformed random play. This helped ensure that genuine progression requires structure rather than luck.

A third 1,000,000-step sweep is run across all levels. This combined performance and fuzzing harness testing. At this scale, random testing becomes useful for surfacing edge-case crashes, malformed transitions, invalid frame outputs, inconsistent hidden-state behavior, and rare action-sequence defects that deterministic tests may miss.

Finally, developer-provided recording-based playback verifies reproducibility. Known-good recordings are replayed under both win and loss conditions, confirming that the engine can serialize and faithfully re-execute action traces. This is important not only for regression testing, but also for debugging, benchmarking, auditability, and future model analysis.

3.5.2 Exploring graph-based state space construction and win probability estimation

Exploration models the environment as an explicit directed graph over reachable states (see figure 3). In this representation, each node corresponds to a unique environment state, while each edge corresponds to a valid player action taken from that state. Node identity is hash-based, allowing the builder to merge distinct trajectories that arrive at the same underlying state. This allows the system to transform repeated simulations into a compact state-space approximation rather than a collection of independent rollouts.

Graph construction begins from the reset state of a selected level. For each visited state, the builder enumerates the currently valid actions and records them as outgoing candidate edges. When an edge is followed, the environment advances by one step, a successor node is constructed from the returned frame and hidden state, and that node is either inserted as a new node or merged with an existing equivalent node. Exploration continues until a configured limit is reached, including step budget, time budget, node limit, or edge limit.

Terminal conditions such as level completion and environment completion are explicitly marked. Invalid actions are recorded as transitions that do not change the state. The implementation also tracks merge density, maximum observed depth, cycle detection, and whether the reachable graph has been fully explored.

The graph reveals structural properties of each environment, exposes cycles, estimates reachability, and quantifies stochastic solvability in a reproducible manner. Even when the full graph cannot be exhaustively enumerated, the system can still produce mathematically grounded bounds on the probability that a random policy will solve the environment. Our acceptance threshold was that a random policy should not successfully solve a level more often than 1 in 10,000 times.

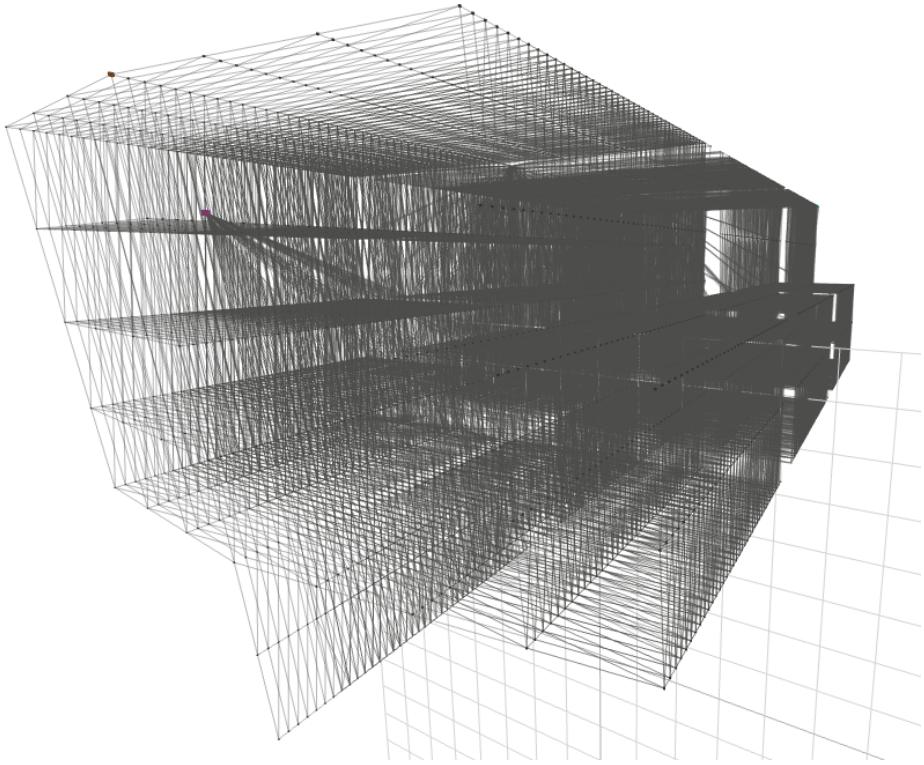

Figure 3: First level of `ls20` in graph form. Notice the three repeating states – an artifact of the three-life mechanic of the level. P_{win} for this level is exactly 1 in 355.

3.6 ARC-AGI-3 environment selection

The ARC-AGI-3 benchmark consists of the following datasets:

Public demonstration set. The public set is designed to demonstrate the ARC-AGI-3 environment format, while being accessible and engaging for human players. These environments are intentionally easier for both humans and AI, with a stronger emphasis on clarity and fun. As the primary entry point to the benchmark, the public set serves as a community-facing "front door," and is expected to receive the majority of human playthroughs. To preserve evaluation integrity, the public set does not comprehensively represent the mechanics found in the private set, reducing the risk of overfitting or targeted optimization.

Private set. The private set is designed to more rigorously test generalization. These environments are significantly more difficult for both humans and AI, and are intentionally out-of-distribution relative to the public set. They cover a broader and more diverse set of mechanics with limited overlap with the mechanics found in the public environments, and they probe greater adaptation capabilities, involving deeper compositional reasoning (while remaining entirely solvable by humans, which is ensured by our human calibration process).

The private set is further subdivided into two subsets: the **semi-private set**, to be used to test frontier models behind an external API (therefore subject to a small risk of data leakage), and the **fully private set**, to be used for the official ARC Prize competition, tightly guarded.

Compared to ARC-AGI-2, which maintains a roughly 10:1 public-to-private ratio, ARC-AGI-3 inverts this balance. The public set shifts from a training resource to a demonstration interface, while the private set becomes the primary basis for evaluation.

Dataset	Purpose	# of environments
Public Demo	A public demonstration set that shows the format and basic mechanics of ARC-AGI-3 environments.	25
Semi-Private	A private hold out set that is used to test models behind an external API.	55
Fully Private	A private hold out that is used for the competition. This set is only given to a very limited number of partners.	55

Table 1: ARC-AGI-3 dataset composition.

4 Measuring performance on ARC-AGI-3

Interactive Reasoning Benchmarks provide a new way to measure the learning efficiency of AI by counting the number of actions (defined below) it takes them to complete a task.

To complete ARC-AGI-3, test takers must use actions in two ways: **exploration** (learning the environment mechanics and goals) and **execution** (carry out a strategy to reach a goal) to complete an environment. Counting the total number of actions taken on first exposure to beat an environment accounts for both of these.

4.1 Scoring methodology

The ARC-AGI-3 scoring method seeks to score the test taker by its per-level action efficiency (as compared to a human baseline), normalized per environment, across all environments.

This scoring function is called RHAE (Relative Human Action Efficiency), pronounced "Ray".

The procedure can be summarized as follows:

- **"Score the AI test taker by its per-level action efficiency"** - For each level that the test taker completes, count the number of actions that it took.
- **"As compared to human baseline"** - For each level that is counted, compare the AI agent's action count to a human baseline, which we define as the second-best human action action. Ex: If the second-best human completed a level in only 10 actions, but the AI agent took 100 to complete it, then the AI agent scores $(10/100)^2$ for that level, which gets reported as 1%. Note that level scoring is calculated using the square of efficiency.
- **"Normalized per environment"** - Each level is scored in isolation. Each individual level will get a score between 0% (very inefficient) 100% (matches or surpasses human level efficiency). The environment score will be a weighted-average of level score across all levels of that environment.
- **"Across all environments"** - The total score will be the sum of individual environment scores divided by the total number of environments. This will be a score between 0% and 100%.

We define the metric as follows:

For a given level l within an environment e, let $a_{l,e}$ be the number of actions taken by the AI agent, and $h_{l,e}$ be the human baseline (defined as the second-best human action count).

The **Level Efficiency Score** $S_{l,e}$ is defined as follows:

$$S_{l,e} = \min\left(1.0, \frac{h_{l,e}}{a_{l,e}}\right)^2 \tag{1}$$

The **Environment Score** E_e is the linearly weighted average of the level scores. For an environment with n levels (where $n = 5$), the weight for level l is $w_l = l$. The score is calculated as:

$$E_e = \frac{\sum_{l=1}^{n} w_l \cdot S_{l,e}}{\sum_{l=1}^{n} w_l} = \frac{\sum_{l=1}^{n} l \cdot S_{l,e}}{\frac{n(n+1)}{2}} \tag{2}$$

The **Total Benchmark Score** T is the mean of all environment scores across the dataset D:

$$T = \frac{1}{|D|} \sum_{e \in D} E_e \tag{3}$$

This formulation is inspired by the robotics navigation Success weighted by Path Length (SPL) metric (*4*), which evaluates not only task completion but also path efficiency.

4.2 Key scoring design decisions

Normalize AI scores with second-best human score

A defining characteristic of ARC-AGI-3 scoring is that AI will be normalized to human level action efficiency. We conducted in-person human testing in a controlled environment to gather human baseline data. An environment was *only* included in ARC-AGI-3 if it passed an "easy for humans" bar. Exactly 10 members of the public are tested on each environment.

We defined the human baseline as the second-best human by number of actions used. This removes the outlier winner while still remaining a strong human capability baseline.

Per-Level vs Per-Environment aggregation

AI performance is compared to human performance *per-level*, then aggregated per environment.

Per-level aggregation strips away noise from uneven level lengths. For instance, we've observed that `vc33`'s level 6 requires 10x actions of its level 1 (50 vs <5). A 10% decrease in efficiency on `vc33` level 6 (+5 actions) would drown out the efficiency observed on level 1.

If we disregard level efficiency and simply look at actions across an entire environment, the longest levels dominate the score and reduce signal from short levels.

Additionally, early levels are meant to be trivial, late levels are more difficult. If we collapse scoring into one environment-denominator you would not be able to tell *where* the agent is weak. With per-level scores you immediately see "agent matches human efficiency on levels 1 through 3, but does not perform on levels 4 and 5".

Lastly, by reinforcing efficiency per level, we won't design environments (or encourage AI) to waste actions on levels because they're still "under budget" for a given environment.

Cap the maximum per-level score

To stop a single glitch-level from distorting an entire environment score, we cap the per-level efficiency an AI can receive at 1.0x human baseline.

For example, suppose the human baseline shows 20 actions needed to complete a level, but an AI discovers a 2-action exploit, the ratio (20/2 = 10x) would overwhelm the environment average. To counter this, we cap the maximum score for a level at 1x the human baseline.

Power law scoring

ARC-AGI-3 uses a power-law efficiency term rather than a linear one. For each level, a test-taker's efficiency is defined relative to the human baseline, and that value is then squared before contributing to the final score. This adjustment increases the penalty for highly inefficient solutions while preserving partial credit. Under a linear formulation, substantial inefficiencies can still yield disproportionately high scores (e.g., 2× the human action count yields 50% credit), reducing the metric's ability to distinguish between near-human and materially suboptimal performance. The power-law transformation improves this discrimination by more heavily penalizing deviations from the human baseline.

For example, if a human completes a level in 10 actions and an AI system requires 100 actions, the raw efficiency is $10/100 = 0.1$. Under the power-law formulation, this becomes $0.1^2 = 0.01$, or 1% credit for that level.

Weighted levels

ARC-AGI-3 also applies level weighting within each environment. Because the earliest levels are intentionally easier and in some cases may be solvable through limited exploration or even chance, they should contribute less to the overall environment score than later levels. We therefore use a simple linearly weighted average across the five levels of an environment.

For example, in a 5 level environment, the per-level score contribution is as follows:

- Level 1 contributes 1/15th of the environment score
- Level 2 contributes 2/15th of the environment score
- Level 3 contributes 3/15th of the environment score
- Level 4 contributes 4/15th of the environment score
- Level 5 contributes 5/15th of the environment score

This ensures that introductory or tutorial-like levels have the smallest influence on the final score, while later levels, which typically require a more complete understanding of the environment's mechanics, contribute the most.

4.3 Leaderboards

ARC-AGI-3 scores are reported on a similar 2D plot as the prior ARC Prize leaderboard used for ARC-AGI-1 and 2. The Y-Axis represents performance (action efficiency, as defined above). The X-Axis continues to represent cost for a given run.

Due to the operational intensity of running an ARC-AGI-3 full evaluation set using high-reasoning frontier model APIs (which could run in the tens of thousands of dollars as of early 2026), we set a hard cutoff of 5x human performance per level. If a human takes 10 actions to beat a certain level on average, then we will cut the AI agent off after 50 actions. This might lead to reporting somewhat lower scores than what would be theoretically feasible using as many actions as the environment intrinsically permits.

Score reporting from the ARC Prize Foundation will be split into two different leaderboards: an official leaderboard, and a community leaderboard.

4.3.1 Official leaderboard

Our intent with the official leaderboard is to accurately help the public sense how close frontier models are to human-level general intelligence. We see general intelligence as the ability to deal with problems that the system was not specifically designed or trained for. This means that the official leaderboard will seek to discount score increases that come from direct targeting of ARC-AGI-3, to the extent possible.

We seek to fight two forms of overfitting that would muddy public sensefinding:

Task-specific overfitting. This includes any agent that is created with knowledge of public ARC-AGI-3 environments, subsequently being evaluated on the same environments. It could be either directly trained on these environments, or using a harness that is handcrafted or specifically configured by someone with knowledge of the public environments. We know such agents can in principle achieve a 100% score on the public set. To demonstrate it, we are releasing an open-source "harness" which scores 100% on all public environments, using human replay. Because it is impossible to ensure that system designers don't use the public environments as part of their work, and because the public set is materially easier than the private set, we will never report public set scores of any system on the official leaderboard. The public set is to be used strictly as a demonstration of what ARC-AGI-3 is – evaluating on it is emphatically *not* a valid measure of progress towards AGI.

Domain-specific overfitting. This includes any agent that is created specifically to play ARC-AGI-3 environments in general, either by being trained on many synthetically generated ARC-AGI-3 lookalike environments, or using a harness that contains ARC-AGI-3 specific strategies. We know that by injecting a high amount of human instructions into a harness, or even hand-crafting harness configuration choices such as which tools to use, it is possible to artificially increase performance on ARC-AGI-3 (without improving performance on any other domain). The purpose of ARC-AGI-3 is not to measure the amount of human intelligence that went into designing an ARC-AGI-3 specific system, but rather to measure the general intelligence of frontier AI systems.

Therefore, **we will focus on reporting the performance of systems that have not been specially prepared for ARC-AGI-3, served behind a general-purpose API** (representing *developer-aware generalization* on a new domain as per (*8*)). This is similar to looking at the performance of a human test-taker walking into our testing center for the first time, with no prior knowledge of ARC-AGI-3. We know such test takers can indeed solve ARC-AGI-3 environments upon first contact, without prior training, without being briefed on solving strategies, and without using external tools.

To measure this, **the official leaderboard will not use a harness to report official scores**. Our position is that future AGI systems will not need task-specific external handholding to approach new tasks.

This approach is further justified by our early testing. We hired researchers to build general harnesses targeting a small set of public environments: `ls20`, `ft09`, and `vc33`. We then tested the harnesses on the full public set (which researchers did not have access to at the time). We found extreme bimodal performance across the two sets, controlling for the same frontier model. For example, in a variant of environment TR87, Opus 4.6 scores 0.0% with no harness and 97.1% with the Duke harness (*12*), yet in environment BP35, Opus 4.6 scores 0.0% under both configurations. This is clear evidence that:

- Frame content perception and API format are not limiting factors for frontier model performance on ARC-AGI-3: with the right handcrafted strategy, frontier models can in fact solve such environments via the current API format.
- Specifically engineered harnesses are not a useful way to measure AGI progress, as their performance on seen environments does not translate to unseen environments, much less to novel domains.

In order to make apples-to-apples comparisons across all different models, we will be using the same system

prompt for all evaluation runs. Models will not be given tools (although they could be using their own tools behind the model's API, which is a blackbox). The code used in the community leaderboard can be found on Github (*5*).

ARC-AGI-3 system prompt:

> *"You are playing a game. Your goal is to win. Reply with the exact action you want to take. The final action in your reply will be executed next turn. Your entire reply will be carried to the next turn."*

At release, frontier models score on the official ARC-AGI-3 leaderboard as follows:

Provider	**Model**	**Score**
Google	Gemini 3.1 Pro Preview	0.37%
OpenAI	GPT 5.4 (High)	0.26%
Anthropic	Opus 4.6 (Max)	0.25%
xAI	Grok-4.20 (Beta 0309 Reasoning)	0.00%

Table 2: Semi-private leaderboard scores for frontier models at release.

4.3.2 Community leaderboard

While the official leaderboard does not include scores achieved using domain-specific harnesses, we recognize the importance of harness research. Better harnesses lead to further task automation abilities, which, while not necessarily representing progress towards AGI, remains economically valuable. We expect that 2026 will see significant progress on harness innovation due to ARC-AGI-3.

To provide a dedicated venue for harness-driven results and to highlight this class of innovation, we introduce a secondary leaderboard, the *community leaderboard*. This leaderboard will be public. Anybody can submit to it, and scores will be self-reported. By default, the ARC Prize foundation will not verify anything on the community leaderboard, and we specifically caution against interpreting any scores on the community leaderboard as evidence of AGI progress.

To note, we expect that the best ideas originating from harness research, if they are sufficiently general, will end up flowing behind the model API layer. For example, the original chain-of-thought research started out as a third-party harness from DeepMind wrapping GPT-3. This evolved into a first-party harness internally at OpenAI (Q* and then Strawberry) and OpenAI later brought this innovation to market as a first-party model called o1. We expect this trend to continue. The best and most general ideas will flow from independent third-party research into first-party harnesses and ultimately into first-party models.

5 Human calibration and solvability

In order for an environment to be included in ARC-AGI-3, it needs to pass the minimum "easy for humans" threshold. Each environment was attempted by 10 people. Only environments that could be **fully solved by at least two human participants** (independently) were considered for inclusion in the public, semi-private and fully-private sets. Many environments were solved by six or more people. As a reminder, an environment is considered solved only if the test taker was able to complete all levels, upon seeing the environment for the very first time.

As such, **all ARC-AGI-3 environments are verified to be 100% solvable by humans with no prior task-specific training**.

If an environment did not meet the minimum solvability threshold, it is returned to the game developer for iteration. To diagnose failure modes, we analyze participant performance at the level of individual environments and levels. In particular, we examine per-level completion rates across participants to identify consistent drop-off points, which often indicate unclear mechanics or unintended difficulty spikes.

To further understand the human performance issues, we review full video replays for each level. These replays provide step-by-step visibility into participant behavior, making it possible to observe precisely where and how test takers become stuck. In practice, this combination of aggregate level statistics and detailed replay analysis enables easy identification and correction of problematic mechanics, ensuring that environments meet the intended "easy for humans" standard.

5.1 Testing protocol

In ARC-AGI-2, human evaluation was conducted in a batch setting, with large-scale testing sessions involving hundreds of participants two to three months apart. In contrast, ARC-AGI-3 adopts a continuous evaluation model to support faster development. Rather than infrequent large cohorts, we conducted smaller-scale testing sessions multiple times per week (Monday, Wednesday, and Friday) at a dedicated testing center in San Francisco. This shift enabled faster feedback cycles.

Participants were presented with a sequence of candidate environments and asked to solve them to the best of their ability within a 90-minute session. No task-specific instructions were provided. Each environment was subject to a soft time limit of 20 minutes, after which participants were prompted to conclude their attempt, and a hard cutoff of 30 minutes was enforced.

Participants received a fixed participation fee of $115–$140 for completing the session, along with a $5 performance-based incentive for each environment successfully solved. This incentive structure was designed to encourage completion while maintaining consistent engagement across environments.

On average, participants completed approximately nine environments per session. A small subset of participants exhibited low-effort behavior, rapidly cycling through environments with minimal engagement. They often abandoned more difficult tasks in favor of attempting a larger number of easier ones. These sessions were excluded from analysis, as this behavior appeared to stem from a misinterpretation of the incentive structure.

An "attempt" was defined as a gameplay session consisting of more than 30 actions and less than 30 minutes of interaction. Participants were limited to a single attempt per environment and could not revisit previously completed levels. However, they were allowed to reset the current level at any time. In some cases, participants reset levels after reaching a solution in order to improve efficiency, though this typically increased total interaction time.

5.2 Participant demographics

Study participants came from diverse professional backgrounds. Participants were members of the general public and not selected for any special training, abilities, or skill sets (see figure 4).

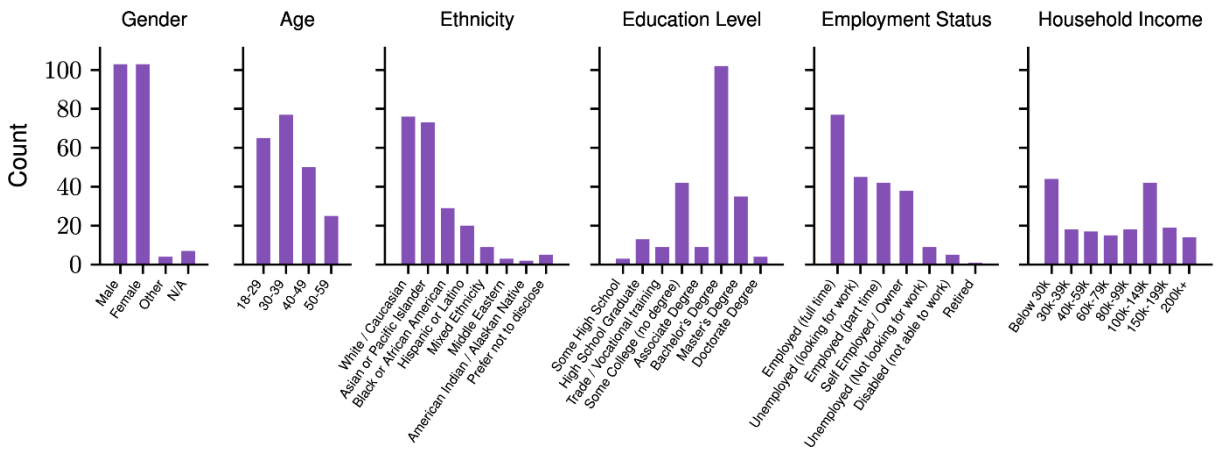

Figure 4: Participant demographics.

5.3 Human performance on ARC-AGI-3

In total, we recorded 486 unique participants across 414 candidate environments. This resulted in 2,893 total environment attempts.

5.3.1 Duration

The total recorded play time for all attempts was 427.9 hours. The median duration of an attempt was 7.4 minutes (see figure 5). Successful attempts had a median duration of 8.1 minutes, whereas unsuccessful attempts had a median of 5.9 minutes. Unsuccessful attempts under 20 minutes had a median of 4.7 minutes.

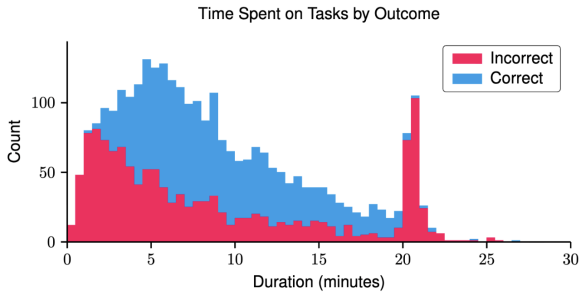

Figure 5: Time spent on environments by outcome, split between successful runs (“correct”) and unsucessful runs (“correct”).

5.3.2 Human efficiency

The human efficiency of beating ARC-AGI-3 is measured by the number of actions it took to complete the environment. Because all human evaluations were conducted as first-run attempts, this data allows us to measure how efficiently humans solve each environment when encountering it for the first time. We track three reference points (see figure 6):

- Optimal playthrough: Empirical estimate of the lower bound on the number of actions needed to solve the environment (once the environment's mechanics and goals are already fully understood.)
- Best first-run playthrough: Best first-run human playthrough aggregated per level. It combines the fewest actions achieved by any test participant on each individual level on a first run, regardless of whether they came from the same person.
- Human baseline: Second-best first-run human playthrough. This is what we use as the human baseline in the official score computation.

The difference between the "optimal playthrough" and the "best first-run playthrough" captures the amount of actions that need to be expended for initial exploration and mechanics learning.

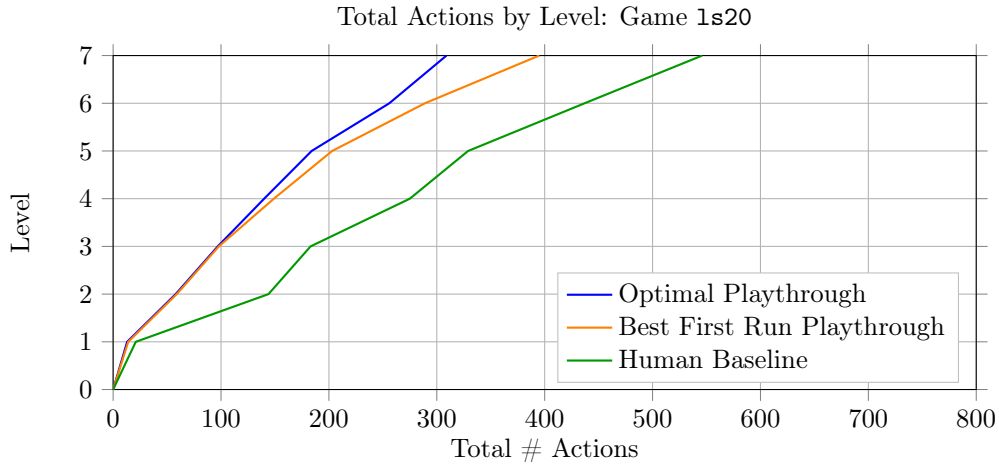

Figure 6: Total actions by level for game `ls20`.

6 ARC-AGI-3 pre-launch testing

Unlike ARC-AGI-1 and 2, we decided to release previews of ARC-AGI-3 prior to the full launch in order to guide our final benchmark design. This gave us critical feedback on what environments were easier and more engaging, and enabled early AI tests to vet our design choices.

To incentivize this, we both hosted an agent preview competition (*14*) and worked with external teams to red-team ARC-AGI-3.

6.1 Agent Preview Competition

The ARC-AGI-3 Preview Agent Competition (*14*) ran for 30 days from July 18 to August 19, 2025. Three public environments were released and three private environments were held back as a hidden evaluation set. Final scoring only measured the ability for AI systems to generalize to this hidden evaluation set.

Top entries included:

- StochasticGoose, Tufa Labs (12.58%, first place) (*18*): A CNN with reinforcement learning to predict which actions would cause frame changes, encoding 64x64 frames through a four-layer convolutional network. It achieved 12.58% and completed 18 levels.

- Blind Squirrel (6.71%, second place) (*22*): A directed state graphs from observed frames.

Both winning approaches used an informed search approach, exploring as much of the action space of the environment as possible in the hope of encountering a winning combination by chance.

6.2 ARC-AGI-3 academic partners

To complement internal development, ARC-AGI-3 included a small number of academic partnerships aimed at exploring agentic approaches to ARC-AGI-3. These collaborations provided early signal on how frontier models behave on the benchmark while helping surface key challenges in harness design, particularly around context management and long-horizon reasoning.

One of the collaborations was with Duke University, which participated through a sponsored effort led by a small research team. Their work focused on building an agentic harness around a large reasoning model (LRM), with particular emphasis on managing interaction history and extracting relevant state from prior actions.

Context management is a central challenge in ARC-AGI-3. Environment frames are 64x64 grids, and maintaining a naive rolling window of observations quickly exhausts a model's context budget. Their harness (*12*) addresses this by allowing the model to execute arbitrary Python code to selectively retrieve and transform information from its action history. This enables more targeted reasoning over past states and improves decision-making efficiency. In evaluation, this approach was able to solve all three public environments with action counts comparable to human performance.

6.3 Community approaches and early experimentation

In parallel with academic partnerships, ARC-AGI-3 was released early to the broader research community to encourage independent experimentation. This early access surfaced a range of novel harness designs and provided additional insight into the emerging design space of agentic systems.

Symbolica AI introduced a harness called *Arcgentica* (*11*), which employs an orchestrator–subagent architecture. A top-level orchestrator does not interact with the environment directly. Instead, it delegates tasks to specialized subagents that return compressed textual summaries. This design constrains context growth and allows the system to maintain a higher-level plan without exceeding context limits. This approach was also able to solve all three public environments.

7 ARC Prize 2026

A key part of the ARC Prize Foundation is hosting the annual ARC Prize competition. This will continue in 2026 across two tracks: The ARC-AGI-3 track and the ARC-AGI-2 track.

The total prize pool is now $2M to encourage open research. Both competitions are held on Kaggle. As always, participants must open source their solutions in order to receive prize money. This will be the final year of the ARC-AGI-2 track, and as such the grand prize is guaranteed to be paid out to the best team this year. The primary focus going forward will be on ARC-AGI-3.

Previous year competition recaps for 2024 (*1*) and 2025 (*2*) can be found at arcprize.org.

8 Conclusions

ARC-AGI-3 introduces an interactive reasoning benchmark for evaluating agentic intelligence, focusing on a system's efficiency at acquiring new skills through exploration, model formation, goal inference, and planning in unfamiliar environments. By structuring environments around core knowledge priors and measuring performance through action efficiency relative to human baselines, the benchmark aims to capture aspects of general intelligence that are not reflected in static evaluation settings. The transition to interactive environments provides a more direct test of whether systems can adapt to "unknown unknowns" without reliance on prior exposure or task-specific optimization.

The development of ARC-AGI-3 also highlights the importance of benchmark design in an era of increasingly capable models. Lessons from earlier ARC-AGI benchmarks suggest that even carefully constructed static datasets can become susceptible to overfitting as training data expands to directly target the benchmark. In response, ARC-AGI-3 emphasizes novelty, compositional generalization, and out-of-distribution design, along with human calibration, to maintain a meaningful evaluation signal.

Initial results from human studies and early AI systems suggest that ARC-AGI-3 presents a qualitatively different challenge from prior benchmarks. Humans are able to reliably solve the environments within bounded time and action budgets, while current AI systems struggle to achieve consistent performance without significant external scaffolding. This gap reflects not only differences in reasoning capability, but also limitations in exploration strategies, hypothesis revision, and efficient planning under uncertainty.

To our knowledge, ARC-AGI-3 is the only unsaturated general agentic intelligence benchmark as of March 2026.

As models continue to improve, evaluation frameworks must evolve to remain informative and resistant to shortcut solutions. Interactive reasoning benchmarks provide one such direction, offering a controlled setting in which to study how systems learn, adapt, and act in new environments. We present ARC-AGI-3 to serve as a useful platform for advancing research in agentic AI systems and for improving our understanding of what constitutes efficient, general-purpose intelligence.

9 Acknowledgments

We thank Surge AI for their early support in brainstorming initial ARC-AGI-3 environment concepts. They provided useful inspiration and helped seed portions of the game development process.

We also thank the developers who contributed to ARC-AGI-3 for their work in designing and implementing the final set of benchmark environments: Pablo Romero Saavedra, Benjamin Morgan, Vadym Andrianov, Flynn Swainston-Calcutt, Tom Elliot, Fraser Scott, Jonathan Pappas, Kevin Johnson, Lukas Donkers, Danielle Goldman, Majid Manzarpour, Nic Tristani, Mattia Traverso, Christian McDonald, Isaac Karth, Philip Dhingra, and Yago Cerqueira.

References

1. ARC Prize 2024 Competition. `https://arcprize.org/competitions/2024`, 2024.
2. ARC Prize 2025 Competition. `https://arcprize.org/competitions/2025`, 2025.
3. ARC Prize Foundation. `https://arcprize.org/`, 2026. Founders: Mike Knoop, François Chollet. Operations: Bryan Landers, Greg Kamradt.

4. Peter Anderson, Angel Chang, Devendra Singh Chaplot, Alexey Dosovitskiy, Saurabh Gupta, Vladlen Koltun, Jana Kosecka, Jitendra Malik, Roozbeh Mottaghi, Manolis Savva, and Amir R. Zamir. On evaluation of embodied navigation agents, 2018.

5. ARC Prize Foundation. ARC-AGI Community Leaderboard. https://github.com/arcprize/ARC-AGI-Community-Leaderboard, 2026.

6. ARC Prize Foundation. ARC-AGI Toolkit. https://github.com/arcprize/ARC-AGI, 2026.

7. ARC Prize Foundation. Gemini 3 Deep Think Preview Verification on ARC-AGI-2. https://huggingface.co/datasets/arcprize/arc_agi_v2_public_eval, 2026.

8. François Chollet. On the Measure of Intelligence. https://arxiv.org/abs/1911.01547, 2019.

9. François Chollet. OpenAI o3 Breakthrough High Score on ARC-AGI-Pub. https://arcprize.org/blog/oai-o3-pub-breakthrough, December 2024.

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

11. Akul Datta, Pratyush Shukla, and Samuel Knutsen. Arcgentica: ARC-AGI-3 Agent Harness Built on the Agentica SDK. https://github.com/symbolica-ai/ARC-AGI-3-Agents, 2026.

12. Alexis Fox, Junlin Wang, Paul Rosu, and Bhuwan Dhingra. Hill-climbing arc-agi-3, 2026.

13. Steve Hsu. Post on LRM automation discovering novel results in quantum physics. https://x.com/hsu_steve/status/1996034522308026435, 2025.

14. Greg Kamradt. ARC-AGI-3 Preview: 30-Day Learnings. https://arcprize.org/blog/arc-agi-3-preview-30-day-learnings, August 2025.

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

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

17. David Silver, Aja Huang, Chris J. Maddison, Arthur Guez, Laurent Sifre, George van den Driessche, Julian Schrittwieser, Ioannis Antonoglou, Veda Panneershelvam, Marc Lanctot, Sander Dieleman, Dominik Grewe, John Nham, Nal Kalchbrenner, Ilya Sutskever, Timothy Lillicrap, Madeleine Leach, Koray Kavukcuoglu, Thore Graepel, and Demis Hassabis. Mastering the game of Go with deep neural networks and tree search. *Nature*, 529(7587):484–489, 2016.

18. Dries Smit. ARC3 Solution. https://github.com/DriesSmit/ARC3-solution, 2025.

19. I. Sorokin and Jean-Francois Puget. NVARC Solution to ARC-AGI-2 2025. https://drive.google.com/file/d/1vkEluaaJTzaZiJL69TkZovJUkPSDH5Xc/view, 2025.

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

21. Ashish Vaswani, Noam Shazeer, Niki Parmar, Jakob Uszkoreit, Llion Jones, Aidan N. Gomez, Lukasz Kaiser, and Illia Polosukhin. Attention Is All You Need. https://arxiv.org/abs/1706.03762, 2017.

22. wd13ca. ARC-AGI-3 Agents. https://github.com/wd13ca/ARC-AGI-3-Agents, 2025.

23. Jason Wei, Xuezhi Wang, Dale Schuurmans, Maarten Bosma, Brian Ichter, Fei Xia, Ed Chi, Quoc Le, and Denny Zhou. Chain-of-Thought Prompting Elicits Reasoning in Large Language Models. https://arxiv.org/abs/2201.11903, 2022.