Hint-on-Failure Program-of-Thought: A Novel Approach to Improve LLM Fluid Intelligence

Category: General Machine Learning

University of Southern California

David Kalin dkalin@usc.edu

Chiddy Golden chiddygo@usc.edu

Ethan Chiang kechian@usc.edu

Magen Mozeh mozeh@usc.edu

Neel Gude ngude@usc.edu

Ameya Deshmukh ameyad@usc.edu

Abstract

The Abstraction and Reasoning Corpus for Artificial General Intelligence (ARC-AGI), introduced in 2019, established a challenging benchmark for evaluating the general fluid intelligence of artificial systems via a set of unique, novel tasks only requiring minimal prior knowledge. It was designed as a "first concrete attempt to measure the definition of intelligence" (Chollet et al., 2024). For five years, little progress was made to improve upon ARC-AGI-1 benchmarks until 2024 when OpenAI's o3-preview model achieved 75% on ARC-AGI-1 (low compute) & 87% accuracy (high compute). ARC-AGI-2 was introduced as the next iteration designed to test efficiency and capability of frontier AI-reasoning models (ARC Prize. (n.d.)).

1. Project Domain & Goals

1.1. Motivation

Humans solve ARC puzzles nearly perfectly, while LLMs remain far below human-level on ARC-2 (~20%). Our research investigates whether a structured, step-by-step reasoning scaffold, augmented with targeted on-failure hints, can boost an LLM's reasoning accuracy and efficiency on challenge ARC-style reasoning tasks over standard prompting methods, and to analyze which task/operation families benefit most

1.2. Goals

Based on our analysis and research into related works in the ARC-AGI/Fluid Intelligence domain (see Section 3), we have identified the following primary and secondary goals for our research.

1.2.1. Improve Upon Current Best-in-Class Performance on ARC-2 Challenge

Our primary goal is to improve upon current best-inclass performance on ARC-AGI-2 benchmarks (ARC Prize (n.d.)) by applying our step-by-step Hint-on-Failure Program-of-Thought (HoF-PoT) methodology (with prompted model error reflection) to a frontier LLM. Ideally, our approach will achieve lift over existing ARC-AGI-2 benchmarks. However, we recognize this is ambitious given the substantial compute and resources involved.

1.2.2. Lift Over Baselines on ARC-AGI-1

Given that our primary goal is highly ambitious, we plan to evaluate whether HoF-PoT can outperform established prompting baselines on ARC-AGI-1 tasks. Thus, even if ARC-AGI-2 improvement is infeasible, we hope to demonstrate that our method yields significant gains over the model's own baseline performance on ARC-AGI-1 tests.

2. Research Design

2.1. Models

We plan to leverage existing frontier LLM models, specifically OpenAI's GPT-5 series (GPT-5, GPT-5 mini & GPT 5 nano) via API. We propose using GPT-5 mini for its advanced reasoning capability, efficiency and optimal cost compared with full GPT-5. However, if GPT-5 mini underperforms, we will switch to the full GPT 5-model. Experiments will proceed in two phases using ARC-AGI benchmark datasets.

2.2. Dataset(s)

We propose using both datasets in two phases for testing and evaluation. Phase I will utilize the ARC-AGI-1 dataset and Phase II will utilize the ARC-AGI-2 dataset. We propose using the ARC-1 Public Training Tasks/Evaluation dataset(s) for development and testing. If we can show lift upon baseline metrics using our model and executor program, we can then move to Phase II to evaluate the model's ground-truth performance and ensure our improvement is not simply a matter of overfitting to the ARC-AGI-1 dataset.

2.2.1. Phase I – Test/Dev (Arc-1)

(Chollet et al., 2024) ARC-AGI-1 is a dataset consisting of various reasoning tasks presented as pairs of grids of discrete symbols. In total, ARC-AGI-1 is a dataset split into four subsets (public and private), but we will only use the public datasets to develop our methodology and evaluate model performance over baselines.

Public Training Tasks (400, easy) – Intended to demonstrate task format and allow the model to learn Core Knowledge Priors.

Public Evaluation Tasks (400, hard) – Intended to allow for local evaluation of research models.

2.2.2. Phase II – Evaluation (ARC-2)

ARC-AGI-2 consists of a much larger, more challenging dataset and presents a much greater challenge to current LLM systems, but it still holds the same fundamental principles and format as ARC-AGI-1. We propose using ARC-AGI-2 as an evaluation metric to test the robustness and our model's ability to generalize. If we cannot surpass ARC-AGI-2 benchmarks, we will still evaluate our method on a verifiable subset of ARC-AGI-2 to ensure ARC-AGI-1 improvements are not due to overfitting.

3. Methodology & Evaluation

Our research seeks to implement and evaluate a new LLM prompting methodology to show lift over baseline metrics on ARC-AGI-1 testing and, if possible, improve upon current best-in-class performance on ARC-AGI-2 puzzles.

3.1. Novel Method: Hint-on-Failure Program-of-Thought (HoF-PoT)

Our HoF-PoT framework proposes a stepwise hypothesis

and evidence-guided decomposition for solving ARC-style tasks. It proposes a hypothesis for each transformation and only upon the model's first "step-failure" do we provide a "hint" to help correct its error. The hint is an input-output grid pair (distinct from the test input) that illustrates the correct transformation step, contrasted against a plausible distractor. The model then explains its adjusted approach and retries the step before continuing its reasoning chain until it provides final output.

3.1.1. HoF-PoT Step-by-Step Breakdown

The HoF-PoT prompting methodology operates as follows:

1. *Stepwise Hypothesis:* The model proposes a hypothesis for the next transformation step (instead of upfront prediction only).
2. *Hint-on-Failure:* If the model's predicted step is inaccurate, we provide an on-failure hint at that exact point. Specifically, we show the model an example of an "initial" input grid and a "post-transformation" grid (unrelated to the current task but requiring the same transformation) to illustrate the correct operation.
3. *Re-Evaluate with Hint:* The model then uses the hint, adjusts its reasoning (providing new "chain of thought" (COT) reasoning) and re-attempts the step again at that point.
4. *Iterative Hints:* If the model fails again, a new hint (using a different example from the prior hint) is provided. We propose using a maximum of three hints for a given step, after which we will provide the model with the correct transformation.
5. *Continuation*: After identifying the correct transformation (via hint or direct exposure), the model continues with the remaining steps until final output is produced.

3.1.2. Technical Requirements

1. *Strict JSON Schema*: For parsing for outputs (per official ARC guidelines – see Kaggle. (n.d.)).
2. *Input/Output Format:* Each task is given to the model as input-output grid pairs (with grids represented as 2D python lists of integers) and one test input grid.
3. *Output Format:* Predictions must be flattened into strings with list rows delimited with an "|".

3.1.3. Primary Evaluation: Accuracy Lift Over Baselines

Our evaluation methodology will focus on the overall performance lift HoF-PoT provides relative to baselines on ARC-AGI-1 as the primary evaluation metric. These aspects include:

1. *By-Category Accuracy:* Performance categorized by transformation type or task (to gain insight into which tasks benefit most).
2. *Attempts Per Task:* The number of attempts (including retries after hints) for the model to solve each task.
3. *Hint Effectiveness:* The impact of hints on the model's success rate (how often a task is solved post-hint).
4. *Efficiency (Time per Task):* The time or compute required to reach a solution, comparing HoF-PoT with baseline prompting to assess efficiency and accuracy.
5. *Stability:* Evaluate whether improvements from Hof-Pot persist across tasks and experiments (indicating robustness).

3.1.4. Secondary Metric (Confirmation): Use Model on ARC-2 Dataset

As a secondary evaluation, we will test our best performing HoF-PoT setup on AGI-ARC-2 tasks. This will confirm the model's generalizability and ensure ARC-1 gains are not due to overfitting, even if we do not surpass ARC-2 benchmarks.

3.1.5. Prompting Baselines to Evaluate Our Model

For comparison, we will implement the following baseline prompting strategies.

1. *Baseline 0 – Direct Answer:* Provide model with the task and ask for output grid directly (no step-by-step reasoning).
2. *Baseline 1 – Chain-of-Thought:* Prompt model to verbalize reasoning before giving final output.
3. *Baseline 2 – Self-Consistency:* Run multiple CoT trials and use the output grid that reaches a trial majority (popular vote).
4. *Baseline 3 – Program-of-Thought:* Prompt model to generate a pseudocode program (sequence of operations) to transform the input grid, then execute the program to obtain the output (no hints given).

4. Literature Review: Comprehensive Survey of Related Works

(Kaggle. (n.d.)) ARC is a competition that provides a publicly accessible dataset and benchmark that assesses a participant's ability to generalize and do abstract reasoning. Past solutions serve as good baselines from which we can improve upon.

(ARC Prize (n.d.)) describes ARC-AGI's history and evolution. Most notably, it emphasizes the philosophy of ARC-AGI as a benchmark for evaluating fluid intelligence on tasks that are "easy for humans, hard for AI." The highest performance rating was at 53%, leaving room for improvements, which is the basis of our project.

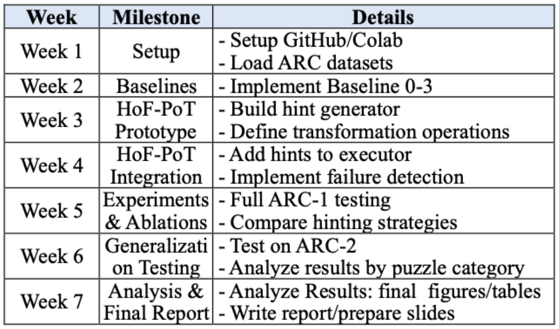
(Chollet et al., 2024) emphasized that early approaches to ARC-AGI relied on symbolic search techniques that failed to handle complexity.

(Ishay, A., et al., 2023) Neural network methods have improved flexibility but have been found to overfit training data.

(DeLong, L. N., et al., 2023) Most recently, neuro-symbolic hybrid methods have emerged, but often suffer from computational costs.

5. Milestones

Week Milestone Details



Week	Milestone	Details
Week 1	Setup	- Setup GitHub/Colab - Load ARC datasets
Week 2	Baselines	- Implement Baseline 0-3
Week 3	HoF-PoT Prototype	- Build hint generator - Define transformation operations
Week 4	HoF-PoT Integration	- Add hints to executor - Implement failure detection
Week 5	Experiments & Ablations	- Full ARC-1 testing - Compare hinting strategies
Week 6	Generalizati on Testing	- Test on ARC-2 - Analyze results by puzzle category
Week 7	Analysis & Final Report	- Analyze Results: final figures/tables - Write report/prepare slides

Appendix

Chollet, F., Knoop, M., Kamradt, G., Landers, B. ARC Prize 2024 Technical Report. https://arxiv.org/pdf/2412.04604, 2024.

Chollet, F., Knoop, M., kamradt, G., Landers, B., Pinkard, H., ARC-AGI-2: A New Challenge for Frontier AI Reasoning System. https://arxiv.org/html/2505.11831v1#bib.bib1, 2025

Bober-Irizar, M., Banerjee, S. Neural networks for abstraction and reasoning. *Sci Rep* **14**, 27823 (2024). https://doi.org/10.1038/s41598-024-73582-7

Simmons-Edler, R., Miltner, A., & Seung, S. (2018). *Program Synthesis Through Reinforcement Learning: Guided Tree Search*. arXiv. https://arxiv.org/pdf/1806.02932

ARC Prize. (n.d.). *What is ARC-AGI?* Retrieved from https://arcprize.org/arc-agi

ARC Prize. (n.d.). *ARC-AGI Leaderboard.* Retrieved from https://arcprize.org/leaderboard

Kaggle. (n.d.). *Abstraction & Reasoning Challenge*. Retrieved from https://www.kaggle.com/competitions/abstraction-and-reasoning-challenge

Ishay, A., et al. (2023). *Neuro-symbolic reasoning with large language models*. NSF Public Access Repository. https://par.nsf.gov/servlets/purl/10475328

DeLong, L. N., et al. (2023). *Neurosymbolic AI for reasoning over knowledge graphs: A survey*. arXiv preprint arXiv:2302.07200. https://arxiv.org/pdf/2302.07200.pdf