The **Abstraction and Reasoning Corpus (ARC)** is a benchmark designed to evaluate general intelligence, not pattern memorization. Unlike typical vision tasks, ARC puzzles require humans to infer **abstract structural rules** — groupings, object relations, symmetry, color interactions — from a few input/output grid examples. Humans solve these tasks effortlessly, but AI systems struggle because ARC demands **conceptual reasoning**, not statistical learning.

The **ARGA** (Abstraction Reasoning Graph Analysis) framework approaches ARC by constructing **graph-based abstractions** and running a search-based solver over these abstractions. Each abstraction (e.g., ccg, nbccg, ccgbr, na) represents a different hypothesis about how to interpret objects and structures in the grid.

Goal of the Project

We explore a simple but fundamental question:

Can a Large Language Model (GPT-5) correctly predict which abstraction best describes an ARC task?

If true, LLMs could guide ARC solvers by providing strong priors on *how to view* the task — an ability that humans rely on naturally.

We constructed two analytical datasets:

Dataset A — Solver Performance Across All Abstractions

For each task in our subset (54 tasks):

1. **We modified ARGA so that it can be run with a manually selected abstraction.**
2. **We executed all nine abstractions independently on each task.**
3. **For each abstraction we recorded:**
    - **Whether it solved the task**
    - **Time taken (in seconds)**
    - **Program length (length of apply_call list)**
4. **We ranked the nine abstractions per task using:**
    - **Time-based rank**
    - **Program-length-based rank**
      using a 1-3-3-4 scheme (ties allowed below 1 second).

This gives the ground-truth notion of:

The "best" abstraction = fastest/shortest correct abstraction

Dataset B — LLM-Predicted Abstraction Rankings

For the same tasks:

1. **We crafted a detailed prompt describing:**
    - **ARC**
    - **ARGA**
    - **All 9 abstractions**
    - **JSON input/output format**
    - **Task-specific reasoning guidelines**
2. **Prompt included:**
    - **Persona priming**
    - **Structural reasoning focus**
    - **Explanations of object/group behaviors**
    - **A self-reflection "mirroring" step**
3. **GPT-5 returned:**
    - **Top-3 abstractions (predicted_1, predicted_2, predicted_3)**
    - **Justifications for each**

We then converted these to rank vectors:

- **predicted_1 → rank 1**
- **predicted_2 → rank 2**
- **predicted_3 → rank 3**
- **all others → rank 9**

Correlation Study:

Do GPT's Rankings Align With the Solver's Best Abstractions?

This is the central part of the project.

We compared, per task, the solver rank vector vs. the GPT-5 rank vector using:

Spearman's Rank Correlation Coefficient (ρ)

which measures monotonic agreement between rankings.

We compute two correlations:

1. **LLM ranks vs solver time-based ranks**
2. **LLM ranks vs solver program-length ranks**

Correlation Results

Comparison	Spearman ρ	p-value	#Pairs
LLM vs Solver (Time)	**0.275**	**1.03e-09**	**486**
LLM vs Solver (Program Length)	**0.272**	**1.11e-09**	**486**

Interpretation

A ρ ≈ 0.27 is:

- **Weak in magnitude**
- **Highly statistically significant**
- **Meaningful for 9-class ranking tasks**
- **Far above random baselines (~0)**

What the p-value Signifies

A p-value answers the question:

If the true correlation were zero (no relationship), how likely is it to observe a correlation as strong as ours just by random chance?

In our case:

- **$p \approx 1 \times 10^{-9}$ (for both time and program length)**
- **This means the probability that our correlation happened by random noise is about: 0.000000001(one in a billion).**

Thus:

The correlation is extremely unlikely to be accidental.

GPT-5's abstraction rankings weaklu align with the solver's performance-driven rankings.

Even though ρ ≈ 0.27 is moderate, the statistical certainty of the relationship is extremely high.

Solver Top-3 Abstractions (Ground Truth)

Frequency of abstractions solving at least one task:

abstraction	solved_count
nbccg	38
ccgbr2	17
ccg	16
ccgbr	16
mcccg	15
lrg	15
nbhcg	11
nbvcg	10
na	7

From these:

Solver Performance Summary

- **nbccg alone solves 70.37%**
- **nbccg or ccgbr2 solves 77.78%**
- **nbccg, ccgbr2, or ccg solves 79.63%**

GPT-5 Abstraction Prediction Performance

Across all tasks:

Metric	**Value**
Top-1 accuracy	**57.43%**
Top-2 accuracy	**83.33%**
Top-3 accuracy	**88.89%**

Thus:

GPT-5 (89.89%) outperforms the solver's own 3 most frequent abstractions (79.63%).

Qualitative Analysis: How GPT-5 Reasons

We studied rationales across predicted_reason_1/2/3.
Patterns include:

Strong Behaviors:

- **Detects connected components reliably**
- **Sensitive to background vs non-background**
- **Recognizes structural patterns like symmetry, alignment**
- **Applies abstraction rules correctly most of the time**
- **Differentiates pixel-level (na) from object-level abstractions**

Weaknesses / Overgeneralization:

- **Sometimes over-prefers mcccg due to the presence of multi-color patches**
- **Slight bias toward high-level abstractions even when pixel-level ones work**
- **Occasionally describes transformations too literally**

Conclusion:

This project shows that:

1. **GPT-5 can accurately infer the correct abstraction for an ARC task, with 88.89% success in its top-3 predictions.**
2. **These predictions correlate weakly with the abstractions that ARGA solves the fastest or with the shortest programs (Spearman $\rho \approx 0.27$, $p < 1e\text{-}9$).**
3. **GPT-5 outperforms ARGA's own empirically strongest top-3 abstractions.**
4. **The correlation suggests that:**
    - **GPT-5 may capture structural priors like in humans to some extent**
    - **GPT-5 could serve as an abstraction selector inside ARC solvers**
    - **This could somewhat reduce ARGA's search time**