ARC Benchmark: Autonomous Learning Without Neural Networks

Built Autonomously by Claude AI

Ghost in the Machine Labs

All Watched Over By Machines Of Loving Grace

January 2026

Abstract

We present a novel approach to the Abstraction and Reasoning Corpus (ARC) benchmark that rejects traditional neural network pattern matching in favor of geometric transform discovery. Our "Origin" system learns rules through multi-example consensus validation, achieving 96.7% accuracy on learned tasks with 30 rules discovered across 58 primitive transforms in under 5 minutes of training--without GPU acceleration.

The key insight: ARC tasks are geometric transformations, not statistical patterns. The correct approach is rule identification, not gradient descent.

The ARC Challenge

The Abstraction and Reasoning Corpus presents 400 training tasks and 400 evaluation tasks. Each task consists of input-output grid pairs demonstrating a transformation rule. The challenge: discover the rule from examples and apply it to test inputs.

Current approaches using large language models and neural networks achieve limited success because they treat ARC as a pattern matching problem. We argue this is fundamentally misguided.

Our Approach: Origin

Core Principle

Multi-example consensus required. A rule is only accepted if it correctly transforms ALL training examples for a task. Single-example fitting causes memorization, not generalization.

Architecture

```
Origin System
|---- Transform Library (58 primitives)
|   |---- Geometric (rotate, flip, scale, crop, transpose)
|   |---- Object (extract_largest, extract_smallest)
|   |---- Mirror (horizontal, vertical, both)
|   |---- Color (keep, remove, extract by color)
|   `---- Gravity (up, down, left, right)
|---- Rule Discovery Engine
|   `---- Tests each transform against ALL training pairs
`---- Learned Rules Store
    `---- Task ID -> Transform mapping
```

The Learning Rule

```
def learn_rule(task):
    for transform in TRANSFORMS:
        if all(transform(input) == output for input, output in task.train_pairs):
            return transform  # Multi-example consensus achieved
    return None  # No single transform solves all examples
```

This is not gradient descent. This is not backpropagation. This is geometric hypothesis testing.

Results

Training Run (5 minutes, CPU only)

```
Models processed:     400 tasks
Rules learned:        30
Accuracy on learned:  96.7% (29/30)
Coverage:             7.5%
Training time:        308 seconds
Hardware:             AMD Ryzen AI Max+ 395 (no GPU)
```

Transform Distribution

Transform Type	Rules Learned
mirror_both	3
mirror_h	2
mirror_v	2
extract_largest	2
extract_smallest	2

rotate_90/180/270	3
flip_h/v	2
crop	1
scale_3	1
tile_1x2	1
outline	1
transpose	1
Others	9

Why Neural Networks Fail at ARC

1. ARC is not IID: Training and test distributions are intentionally different
2. Small sample sizes: 2-5 examples per task, insufficient for statistical learning
3. Exact solutions required: 99% correct is 100% wrong for grid tasks
4. Novel reasoning: Each task requires discovering a new rule, not applying learned patterns

Neural networks approximate functions through statistical regularities. ARC requires exact geometric rule identification. These are fundamentally different capabilities.

Implications

For AGI

The ARC benchmark was designed to measure "general intelligence"--the ability to reason about novel problems. Our results suggest that general intelligence is better modeled as:

* Rule discovery over pattern matching
* Geometric reasoning over statistical correlation
* Exact verification over probabilistic estimation

For Compression

This connects to our Harmonic Stack research: if intelligence emerges from a small set of geometric primitives (our 194,471 junctions), then ARC-style reasoning may be achievable with minimal parameter counts.

Next Steps

Current limitation: single-transform solutions only. The remaining 370 tasks likely require:

* Compositional transforms: crop -> scale -> rotate
* Conditional rules: if color == X then transform_A else transform_B
* Object-relative operations: for each object, apply transform

We estimate compositional chaining will increase coverage to 20-30%.

Reproducibility

All code available at: https://github.com/7themadhatter7/harmonic-stack

```
# Run ARC training
cd ~/sparky
python3 arc_continuous_training.py

# Check results
cat arc_continuous/origin_rules.json
```

Conclusion

ARC is not a pattern recognition benchmark. It is a geometric reasoning benchmark. The path to solving it runs through rule discovery and compositional geometry--not larger language models.

Our Origin system demonstrates this with 30 rules learned in 5 minutes without a GPU. The remaining challenge is compositional: combining primitives into chains that solve multi-step transformations.

The geometry is the intelligence.

Ghost in the Machine Labs -- A 501(c)(3) Initiative

Website: https://allwatchedoverbymachinesoflovinggrace.org GitHub: https://github.com/7themadhatter7/harmonic-stack