Approach: ARC AGI Agent

1. Core System Components

The ARC AGI agent is built using modular components that work together to analyze and solve grid-based transformation tasks. At the center is the ToTArcAgent, which leverages Tree-of-Thought reasoning to explore possible transformation paths.

- ToTArcAgent: Main reasoning engine
- MetaProgrammer: Learns transformations from training examples
- PatternAnalyzer: Extracts grid structure and relationships
- prims: Library of transformation functions
- CNN: Extracts visual features from grids
- evaluate_all_enhanced: Runs multiple tasks and collects performance metrics

2. Tree-of-Thought Agent (ToTArcAgent)

The ToTArcAgent uses Tree-of-Thought search, which builds a decision tree of potential transformation steps. Each node in the tree represents a new state of the grid after applying a transformation. The agent uses:

- Beam search to keep only the best states
- Loop detection to avoid revisiting grids
- Scoring based on similarity to training outputs
- Confidence estimation to pick reliable solutions

3. Meta-Programming (MetaProgrammer)

The MetaProgrammer module is responsible for analyzing training input/output pairs and generating transformation programs. It uses transformation detection to classify tasks and create sequences of operations.

- Detects simple types (rotate, flip, color change)
- Synthesizes multi-step or adaptive strategies
- Generates reusable program templates for future tasks

4. Pattern Analysis (PatternAnalyzer)

This component examines the structure of input and output grids to detect patterns and identify differences. It extracts metrics such as:

- Grid shape and bounding box
- Number and distribution of colors
- Symmetry (horizontal, vertical, rotational)
- Sparsity and connectivity

This helps inform the MetaProgrammer about what type of transformation to attempt.

5. Transformation Library (prims)

The prims dictionary contains all the transformation functions used by the agent. These include simple geometric transformations and advanced operations.

- Geometric: rotate90, flip_h, transpose
- Morphological: dilate, erode
- Region-based: crop_nonzero, region_fill
- Task-specific: transform_custom, diagonal_propagate_transform

6. Feature Extraction

Feature extraction combines deep learning and handcrafted methods to describe grid properties.

- CNN: Multi-layer model that captures local, medium, and global patterns
- Handcrafted: Statistical and geometric features (color distribution, symmetry, connectivity)
- Combined feature vector used for scoring and pruning in ToT search

7. Task Solving and Evaluation

Each task is solved using the enhanced ToT approach. The agent selects the most promising transformation sequence, applies it, and compares the result with the expected output.

- solve_task_enhanced: Solves individual task
- evaluate_all_enhanced: Loads and solves all tasks from a folder
- Tracks accuracy, saves detailed results in JSON format

8. Summary of the Approach

The ARC AGI agent integrates symbolic reasoning and deep learning to solve transformation-based grid tasks. Its Tree-of-Thought mechanism allows for flexible exploration, and its MetaProgramming engine allows for adaptive solution generation.

- Tree-of-Thought enables flexible stepwise reasoning
- MetaProgramming enables learning from examples
- CNN and features ensure accurate evaluation
- Primitives and pattern analysis ensure modularity

This approach aims to solve complex ARC tasks in a general, human-like way.