# A Non-LLM Graph-Search Agent for ARC-AGI-3 with Novelty-Guided Exploration and Case-Based Memory

Mohammad Saadati  Laura Mascarell  Sundong Kim

Department of AI Convergence, Gwangju Institute of Science and Technology (GIST)

## Abstract

When an agent cannot query an external supervisor at inference time, what can it learn from and how far does that take it? We study this question in the context of **ARC-AGI-3**, where offline evaluation forbids internet access and rules out language-model prompting at play time. We propose an agent that derives four signals from observations, **novelty, sub-goal progress, anti-pattern, and exploration depth**, and uses them to drive both a tabular Q-function and case-based memory, backed by an Iterative Deepening A* (IDA*) planner over a masked-state graph. On six public environments, our agent is substantially more action-efficient than verified offline baselines on the levels each solves. Notably, the learned components contribute by **expanding which environments the agent can solve at all**, rather than by speeding up search on environments already within reach.

## Motivation & Research Questions

**The problem:** Agents deployed under compute or connectivity constraints cannot rely on a human labeler or frontier model at inference time. The agent must learn from the environment itself, using only signals computable from observations.

**Test bed:** ARC-AGI-3 Kaggle competition: turn-based games with unfamiliar mechanics and unannounced win conditions, both inferred through interaction. The competition forbids internet access at evaluation, ruling out LLM prompting at play time.

We address two research questions:

- **Q1:** Does a fully offline agent driven by observation-derived signals use *fewer actions* than baselines?
- **Q2:** Does adding a Q-function and memory improve *capability* beyond search alone?

## Problem Setting and Four World-Feedback Signals

Each environment exposes a $64 \times 64$ frame (16-color palette). The agent selects from five key actions, an Undo, and a coordinate-click on a $64 \times 64$ grid, yielding a nominal branching factor of $\approx 4{,}102$. Mechanics and goals must be inferred through interaction.

**We identify four signals computable from the frame stream alone:**

- **Novelty** — an action transitioning to a previously unobserved state; positive feedback for productive exploration.
- **Sub-goal progress** — an action changing the playground region (the gameplay area, distinct from the UI overlay); a weak positive signal subordinated to novelty.
- **Anti-pattern** — an action sequence leading to a characteristic failure marker (loss-of-life indicator, playground reset, or prolonged non-progression); penalizes future exact matches.
- **Exploration depth** — the number of actions from the trajectory's start; signals commitment to longer plans rather than shallow thrashing.

These signals are heterogeneous in time scale, sign, and noisiness, and require no external supervisor. They serve as the **shared supervisory interface** across both the Q-function and case-based memory.

## Agent Architecture

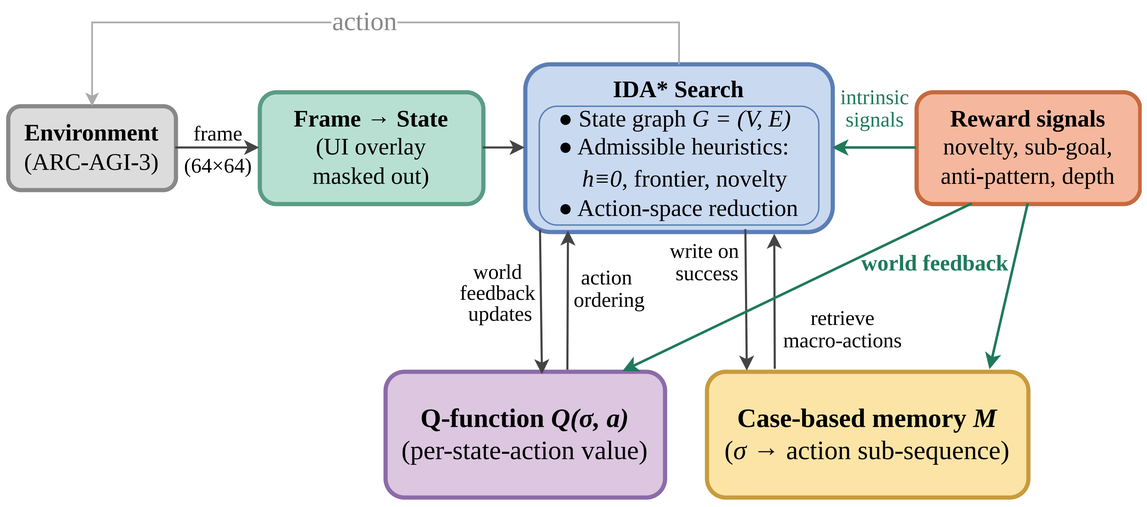


Figure 1. **Agent architecture and information flow.** Raw frames from the ARC-AGI-3 environment are masked to obtain state $s$ and signature $\sigma$, then reduced via action-space pruning. IDA* searches the resulting state graph $G = (V, E)$ using admissible heuristics ($h \equiv 0$, frontier, novelty). Four intrinsic signals — novelty, sub-goal progress, anti-pattern, and exploration depth — are computed from the masked-frame stream and combined into a single reward (world feedback). This reward updates the tabular Q-function $Q(\sigma, a)$ (used to reorder candidate actions during search) and writes to the case-based memory $M$ on level completion (used to retrieve macro-action shortcuts on subsequent levels).

## Per-Step Loop

The agent operates in a per-step loop:

1. **Perceive** — mask the UI overlay, compute state $s$ and signature $\sigma$.
2. **Reduce action space** — pixel-grouping, no-effect pruning, foreground-background segmentation.
3. **Plan via IDA*** — search with three heuristics ($h \equiv 0$, frontier-distance, pixel-novelty); $Q$ reorders primitive actions and $M$ supplies macro-actions.
4. **Execute** the first action of the recommended path.
5. **Update** $Q$ via standard tabular Q-learning ($\alpha = 0.1$, $\gamma = 0.95$) on every observed transition, including off-path expansions.
6. **On level completion**, segment the trajectory at sub-goal satisfaction events and add segments to $M$ for use on subsequent levels.

**Reward formulation:**

$$r(s, a, s') = w_n \cdot \mathbb{1}[s' \notin V] + w_p \cdot \mathbb{1}[\Delta_{\text{play}}(s, s')] - w_a \cdot \mathbb{1}[\text{match}_{\text{anti}}(\tau_{s'})] + w_d \cdot \text{depth}(s')$$

with $w_n = 1.0$, $w_p = 0.2$, $w_a = 2$, $w_d = 0.05$ — novelty as primary driver, anti-pattern as strong correction, sub-goal and depth as smaller biases.

## Evaluation Setup

We evaluate on **six environments** from the ARC-AGI-3 public set: `cd82`, `ft09`, `lp85`, `ls20`, `r11l`, and `vc33`. Execution mirrors the Kaggle regime: offline, with a per-environment action budget of 10,000 actions. Five seeds per (environment, configuration) cell.

Figure 2. The six public ARC-AGI-3 environments used in evaluation. They differ substantially in visual composition, spatial layout, and interaction modality, despite sharing the same input format ($64 \times 64$ frame, 16-color palette) and action space. Mechanics and win conditions must be inferred through interaction.

We compare against four references:

- **StochasticGoose** — CNN trained online to predict frame-change actions (1st in preview competition).
- **BlindSquirrel** — directed state graph with neural action-value model trained between levels (2nd in preview competition).
- **Search-Only ablation** — our agent with $Q$ and $M$ disabled.
- **Random policy** — uniform random action selection (lower bound).

## Main Results: Per-Level Action Counts

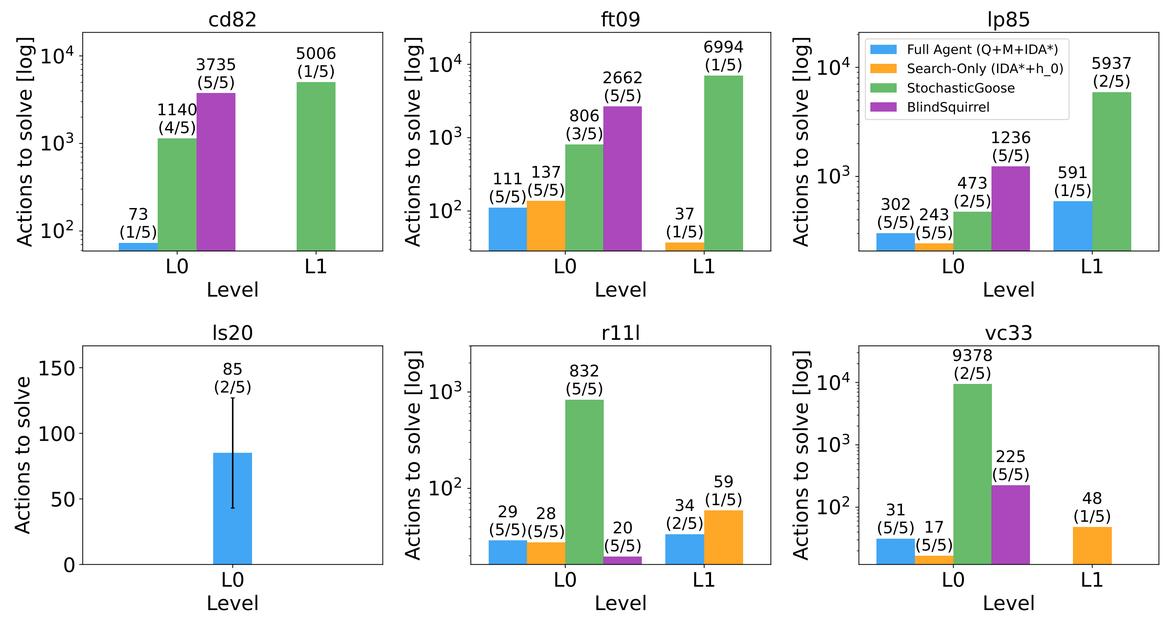


Figure 3. Per-level trajectory lengths (mean actions to solve) across configurations on six public ARC-AGI-3 environments. Each bar shows the mean over solving seeds, with $n/N$ annotated; panels with $>20\times$ range use log scale. The random baseline solves zero levels and is omitted.

**Key findings on level 0:**

- Full agent solves **29 to 302 actions** across five environments.
- **1.6× to 300×** faster than StochasticGoose; **4× to 51×** faster than BlindSquirrel where each baseline solves.
- On `ls20`, both StochasticGoose and BlindSquirrel *fail to solve*; the full agent solves 2 of 5 seeds.
- On `cd82`, BlindSquirrel achieves higher reliability (5/5 vs 1/5), but the full agent uses **51× fewer actions** when it solves; search-only solves none.

## Decomposition: Search vs. Learning

**Headline finding:** the contributions of search and learning decompose cleanly.

- **IDA* + action-space reduction drives per-level efficiency.** The search-only ablation matches the full agent's level-0 speed within error bars on the four environments where both solve.
- **$Q$ and $M$ expand solving capability.** The full agent solves `ls20` (2/5) and `cd82` (1/5) where search-only solves *neither*.
- **Level 1 reach is environment-dependent.** The full agent reaches L1 on `r11l` (34 actions) and `lp85` (591 actions); search-only reaches L1 on `vc33` (48 actions) and `ft09` (37 actions); offline baselines either fail or require many actions.

Learned representations in this regime serve *capability expansion*, not *search acceleration*.

## External Validation: Kaggle Leaderboard

On the **public ARC-AGI-3 Kaggle leaderboard**, our agent scores **0.44**, exceeding StochasticGoose's 0.25. Third-party verified validation on a held-out evaluation set, with the same unchanged agent.

## Conclusions and Future Work

- **World-feedback signals can substitute for external supervision** in deployment-constrained agentic settings. Four observation-derived signals serve as a shared supervisory interface for both Q-learning and case-based memory.
- **Search and learning decompose**: IDA* with action-space reduction drives per-level efficiency; the Q-function and case-based memory expand the set of environments the agent can solve.
- **Validated externally**: outperforms verified offline baselines on Kaggle leaderboard (0.44 vs. 0.25 for StochasticGoose).
- **Future work:** tighten state signatures with object-level semantics; meta-learn reward weights from cross-environment outcomes; replace tabular $Q$ with a function approximator; principled ablation over signal set and weights.



mohammadsaadati@gm.gist.ac.kr