Research Square

Preprints are preliminary reports that have not undergone peer review.
They should not be considered conclusive, used to inform clinical practice,
or referenced by the media as validated information.

Can Environment-Agnostic Agents Solve ARC-AGI-3 from Warped and Noisy Images?

Sergey Rodionov

sergey@singularitynet.io

SingularityNET

Research Article

Keywords: Environment-agnostic agents, ARC-AGI-3, visual robustness, geometric distortion, pixel noise, tool use

Posted Date: November 10th, 2026

DOI: https://doi.org/10.21203/rs.3.rs-11658804/v1

License: This work is licensed under a Creative Commons Attribution 4.0 International License.
Read Full License

Additional Declarations: No competing interests reported.

Can Environment-Agnostic Agents Solve ARC-AGI-3 from Warped and Noisy Images?

Sergey Rodionov[1*]

[1]SingularityNET, Zug, Switzerland.

Corresponding author(s). E-mail(s): sergey@singularitynet.io;

Abstract

We test an environment-agnostic agent's ability to solve public ARC-AGI-3 games from noisy and geometrically distorted images. The agent receives all information about the board's contents through images, with no textual or numerical grid representation. We apply Gaussian pixel noise and random projective or nonlinear warps, sampled independently for each frame. Geometric distortion makes clicking particularly difficult: the agent must infer coordinates on the original, unwarped board. In additional experiments, we provided cursor feedback showing where proposed board coordinates place a cursor.

For GPT-5.6-sol, geometric distortion reduced performance much more than pixel noise alone. Cursor feedback partly restored performance under projective transformations and moderate nonlinear warping. Under the strongest warping and noise tested in the main study, GPT-5.6-sol scored only **∼15%** RHAE with or without cursor feedback. By contrast, the same agent with GPT-6-astra achieved a near-ceiling 95.44% RHAE under the same distortions, even *without cursor feedback*.

To probe the limits of GPT-6-astra, we conducted followup experiments without cursor feedback. Adding random board rotation after each action to the strongest main-study setting reduced RHAE only modestly. In separate experiments, extreme pixel noise under strong nonlinear warping substantially reduced performance, but the agent still achieved 17.37% RHAE at the highest noise level tested. Agent-created scripts and denoised, unwarped images found in its workspace suggest that successful play may combine image processing with decisions under substantial uncertainty. Prior exposure to the public games may also have contributed.

Keywords: Environment-agnostic agents, ARC-AGI-3, visual robustness, geometric distortion, pixel noise, tool use

1 Introduction

Can an environment-agnostic agent solve interactive games when it observes the game board only through warped and noisy images? We investigate this question using the *Stop-and-report* agent from our previous work, which evaluated the same agent implementation across ARC-AGI-3, AGI Maze, and Terminal-Bench [1]. Here, *environment-agnostic* means that the agent remains unchanged across environments, while each environment supplies its own task description and interaction interface.

We use ARC-AGI-3, an interactive benchmark in which agents must discover game rules and objectives through exploration [2]. Our environments provide minimal descriptions of the available actions and ask the agent to solve the game using as few actions as possible (see Appendix A.1 for the shared description). All board contents are supplied as rendered images, accompanied by basic game metadata; the environment provides no textual or numerical grid representation.

We modify these images using additive Gaussian pixel noise, random projective transformations, and smooth nonlinear warps that bend the image at multiple spatial scales. JPEG compression introduces additional visual artifacts. For game observations, we independently resample pixel noise and projective or nonlinear warps for every frame, including intermediate frames returned by an action. The geometry is therefore not fixed over a game or level: even an unchanged board can appear under a new transformation in the next frame. Consequently, changes between images can come from the game itself, from noise and geometric distortion, or from both. The agent must recognize relevant objects, relate them across differently distorted views, and infer the consequences of its actions.

Geometric distortion also creates a separate difficulty for games that require clicks. Although the agent sees a target in a warped image, a click must specify coordinates on the original 64×64 board. Recognizing the intended target therefore does not directly determine which coordinates to submit. The agent must infer the relationship between the displayed image and the board coordinate system, without receiving the transformation parameters.

To investigate whether visual feedback helps with this difficulty, we introduce an auxiliary action, `highlight_pixel`, that provides cursor feedback. The agent proposes a board coordinate and receives two newly rendered images of the current state under a shared geometric distortion: one unmarked and one with a cursor pointing to the corresponding board location. This action does not advance the game or consume its game-action budget. The cursor image is the only additional spatial feedback: the environment supplies no coordinate-conversion tool or corrected click coordinates. The action captures one aspect of mouse interaction, while retaining an important difference: the agent proposes coordinates and can inspect where they point, rather than clicking directly on a visible target.

We test how well an environment-agnostic agent solves games from noisy images whose geometry changes from frame to frame. Section 2 relates this study to work on visual agents, robustness to image corruption, and cursor feedback.

Code and data availability.

Source code and documentation are available through the EnvTraverse project page at https://github.com/astroseger/envtraverse, and the exact software version used in this study is archived on Zenodo [3]. The supporting dataset [4] contains per-game action records, scorecards, summary tables, and configurations for rerunning both studies, together with the complete artifacts for the `lf52` run at $\sigma = 1200$ examined in Section 6.3. A reduced main-study archive, omitting JPEG observations after step 10, is linked from the project page. The complete collection of run artifacts totals approximately 700 GB, largely because noisy images compress poorly, and is available in full or in any subset on request.

2 Related Work

Visual agents and observation representations.

VISTA [5] studies how a visual harness can support reasoning in interactive environments, including ARC-AGI-3. It preserves past observations in a lossless visual memory and lets the model retrieve frames and inspect image regions. Its evaluation also includes a 3D rendering of the same ARC-AGI-3 games. The closest analogue in our evaluation is the homography condition, since both alter the apparent viewing geometry of the board. Our emphasis is on geometry that is resampled *independently for every game-observation frame*, together with pixel noise and nonlinear deformations: the agent must relate observations whose spatial correspondence changes throughout interaction. We study these conditions using the previously evaluated environment-agnostic Stop-and-report agent [1], keeping its implementation unchanged.

Robustness to visual and environmental corruption.

VLM-RobustBench [6] evaluates vision-language models under image corruptions, including Gaussian noise, JPEG compression, and geometric transformations, on visual question-answering benchmarks. It finds that geometric and resampling distortions can impair performance more than severe noise and compression, providing context for our comparison of noise and geometry. Zhao et al. [7] similarly study the sensitivity of GUI grounding models to natural image corruptions and adversarial perturbations. Robustness has also been studied over complete interactions: RobustNav [8] evaluates navigation agents under visual and dynamics corruptions, while AgentHijack [9] evaluates computer-use agents under disruptions such as pop-ups and resolution changes. Our study examines a complementary setting in which the game dynamics and action coordinate system remain unchanged while the observation geometry varies from frame to frame. Performance depends jointly on interpreting the corrupted images, discovering game rules, and selecting effective actions over a complete game, rather than on the accuracy of an isolated visual answer or coordinate prediction.

Cursor feedback and action grounding.

A related approach in GUI grounding is to show a model where its proposed coordinates fall in an image, allowing it to inspect and revise its prediction. GUI-Cursor [10] trains this process through reinforcement learning and evaluates it both on grounding

tasks and within an online computer-use agent. PrecisionCUA [11] similarly displays a marker at a predicted location in an editor screenshot, using annotated target regions to determine whether another attempt is needed. Our `highlight_pixel` action provides this form of visual feedback to an unchanged environment-agnostic agent during game solving. Here, the proposed coordinates refer to the original board, while each query displays their location under a newly sampled distortion. The agent must also decide which game locations are useful to act on, and the highlight response provides no judgment of target correctness.

3 Experimental Environments

The main study uses the *Stop-and-report* agent (`agent_v1_01_wstop`) from our previous study, connected to the environment through EnvTraverse [1]. The followup agent variants are described in Section 6.2.1. At initialization, the environment supplies a description of the task and interaction interface. The agent then submits JSON actions and receives observations as collections of named files, together with terminal status. EnvTraverse does not prescribe the contents or formats of these files; each environment defines its own observation interface.

We use the three environment variants summarized in Table 1. They share the underlying ARC-AGI-3 games and standard game actions, while differing in their image observations and access to auxiliary cursor feedback. In the latter two variants, pixel noise and geometric warping can be enabled separately or together. We first define *Clean-view*, then describe how the other two variants extend it.

Table 1 Environment names used in this paper and their implementation identifiers.

Paper name	Implementation identifiers	Description
Clean-view	`arc3_minimal_png`	Unmodified PNG board images and metadata.
Distorted-view	`arc3_minimal_img_wnoise3` `arc3_minimal_img_wnoise4`	Pixel noise, compression artifacts, geometric warping, or their combinations.
Cursor-feedback	`arc3_minimal_img_wnoise3_whighlight2`	Main-study Distorted-view transformations, plus `highlight_pixel`.

Distorted-view uses `arc3_minimal_img_wnoise3`, except for the random-rotation experiment in Section 6.1, which uses `arc3_minimal_img_wnoise4`. This updated implementation adds support for an optional board-rotation step before warping.

3.1 Clean-view

Clean-view is the baseline environment. Each instance contains one ARC-AGI-3 game, including its successive levels, and returns unmodified rendered board images in PNG

format. The 64×64 board is rendered as a 512×512-pixel image ($8\times$ scale). The agent must obtain all information about the board's contents from these images. No ASCII frames, numerical grid representations, or other separate pixel data are provided.

Environment description.

The environment description asks the agent to solve all levels using as few actions as possible and explains the action syntax and general action conventions. It explicitly tells the agent that the board has size 64×64 and that coordinate actions use an upper-left origin. It provides no game-specific rules or solution guidance. Appendix A.1 reproduces the complete description supplied to the agent.

Actions.

The environment exposes the standard ARC-AGI-3 actions (see Appendix A.1). The coordinate-based action `ACTION6` takes integer coordinates on the original 64×64 board.

Observations.

At initialization and after each accepted game action, the environment returns an observation as a collection of files. Table 2 gives the complete inventory for *Clean-view*. The metadata reports game status, completed levels, the total number of levels required to win, available actions, and the preceding action input. It supplies no pixel values or alternative representation of the board contents.

Table 2 Observation files returned by Clean-view. The environment description is supplied separately at initialization.

File	Contents and presence
`metadata.txt`	JSON with `state`, `levels_completed`, `win_levels`, `available_actions`, and `action_input`; present in every observation.
`final_frame.png`	The last frame returned by the game.
`intermediate_frame_00.png`, `intermediate_frame_01.png`, ...	Earlier frames from the same observation, when multiple frames are returned.

Termination.

The environment terminates at `WIN` or when the current-level action counter exceeds 1,500. The counter includes `RESET`, spans attempts at the same level, and is cleared when a new level is reached. This action limit is an experimental choice, not a native ARC-AGI-3 limit.

3.2 Distorted-view

Distorted-view differs from *Clean-view* only in the rendered images: we apply pixel noise, geometric distortion, or both. Pixel noise and projective or nonlinear warps are sampled independently for every frame, including intermediate frames. Consequently, an object can appear at different positions or with different shapes across observations even when the underlying board remains unchanged. The agent must obtain all information about the board's contents from these distorted images.

3.2.1 Visual transformations

Gaussian pixel noise.

We add independent, zero-mean Gaussian noise to each rendered pixel and RGB channel. The noise perturbs colors and can obscure boundaries and small visual features. Its strength is specified by the standard deviation of the added intensity values.

Random projective transformations.

Projective transformations, or homographies, change the board's overall orientation, scale, and perspective while preserving straight lines [12]. We generate them by combining randomly sampled rotation, shear, scaling, translation, and perspective parameters.

Multiscale nonlinear warping.

We generate smooth nonlinear warps using a cubic B-spline free-form deformation (FFD) representation [13]. Control coefficients on a grid define a smoothly varying displacement field. We add fields from a coarse grid and a finer grid to combine broad deformations with variation at a smaller spatial scale. These warps can bend straight lines and deform different parts of the board by different amounts.

JPEG compression.

JPEG encoding introduces compression artifacts in addition to the Gaussian pixel noise.

Image preparation.

Starting from the same rendering as *Clean-view*, we apply geometric distortion, add Gaussian pixel noise, and encode the result for delivery to the agent. When geometric distortion is used, the canvas is padded to accommodate the warped image. Appendix B gives implementation details.

3.2.2 Tested settings and examples

The main study varies pixel-noise strength and the type and magnitude of geometric distortion in *Distorted-view* using the settings below. These parameters also configure the images in *Cursor-feedback* (Section 3.3). Figure 1 illustrates their effects on the same game state. Additional rotation and extreme pixel noise are introduced in the followup experiments in Section 6. Figure 3 compares the strongest main-study noise level with the higher followup noise levels under strong nonlinear warping.

Pixel noise and encoding.

In the main study, we test Gaussian pixel noise with standard deviations $\sigma = 40, 80, 120$ on the 0–255 intensity scale. These images use JPEG quality 25; images without Gaussian noise ($\sigma = 0$) use lossless PNG. Comparisons with $\sigma = 0$ therefore include both added noise and JPEG artifacts.

We use *pixel-only noise* for conditions without geometric distortion. Observation files follow Table 2, with `.jpg` replacing `.png` for JPEG images.

Homography setting.

The rotation, shear, two scale factors, two translations, and two perspective coefficients are drawn independently from uniform distributions within the ranges in Table 3.

Table 3 Sampling ranges for the random homography. Translation bounds are relative to the unpadded, enlarged image dimensions. Perspective coefficients use the centered, normalized coordinates described in Appendix B.

Parameter	Sampling range
Rotation	$[-5°, 5°]$
Horizontal shear angle	$[-5°, 5°]$
Horizontal and vertical scale	$[0.925, 1.075]$
Horizontal and vertical translation	$\pm 2\%$ of the corresponding image dimension
Two perspective coefficients	$[-0.075, 0.075]$

Nonlinear warp settings.

Both settings combine 4×4 and 8×8 control grids, with the coefficient bounds in Table 4. We call these settings *moderate nonlinear warping* and *strong nonlinear warping*. Coefficients are initially sampled independently and uniformly, then checked using the acceptance criterion described in Appendix B.

The moderate setting was chosen to produce distortion visually comparable to the homography setting.

Table 4 FFD control-coefficient ranges for each axis, relative to the unpadded image side $L = 512$ pixels. The displacement at each pixel is computed from both grids.

Setting	4×4 grid	8×8 grid
Moderate	$\pm 0.125L$ (± 64 px)	$\pm 0.03125L$ (± 16 px)
Strong	$\pm 0.375L$ (± 192 px)	$\pm 0.09375L$ (± 48 px)

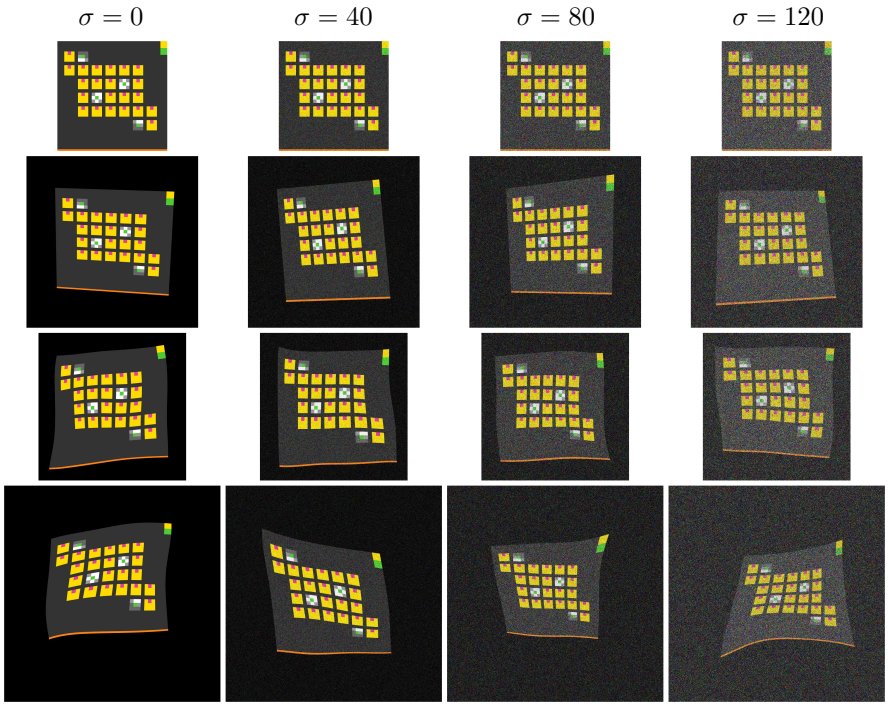


Fig. 1 Examples of the observation corruptions applied to the same frame from `ft09`. Rows, from top to bottom: no geometric distortion, a random homography, moderate nonlinear warping, and strong nonlinear warping. Columns show Gaussian pixel-noise standard deviations $\sigma = 0$, 40, 80, and 120. Geometric samples differ across examples. The zero-noise images use PNG encoding; positive-noise images use JPEG quality 25. The game state is unchanged across all panels.

3.3 Cursor-feedback

Cursor-feedback extends the main-study version of *Distorted-view* by adding one auxiliary action, `highlight_pixel(x,y)`. It retains the image transformations and parameter definitions in Section 3.2. The action's arguments use the same original board coordinates as a click. It renders the current game state twice: once without a cursor and once with a cursor whose tip points to the transformed center of the selected board pixel. Figure 2 shows an example of the returned pair.

Each call samples a new geometric transformation shared by the two images, while pixel noise is sampled independently for each image. This geometry need not match the preceding observation.

The added action syntax and description supplied to the agent are reproduced in Appendix A.2. A highlight response contains `metadata.txt`, `frame.jpg`, and `frame_highlighted.jpg`, with `.png` replacing `.jpg` in the image filenames when PNG encoding is used. It contains no intermediate frames. The metadata's `action_input` records the queried board coordinates.

Highlighting leaves the game state unchanged and does not consume the game-action budget. The cursor image is the only additional spatial feedback: the environment supplies no clean board image, transformation parameters, coordinate-conversion tool,

or corrected coordinates. The agent can inspect where its chosen board coordinates place the cursor before revising them or issuing a game action.

The action approximates the visual feedback available when using a mouse, while requiring the agent to propose board coordinates rather than point directly at a visible target.

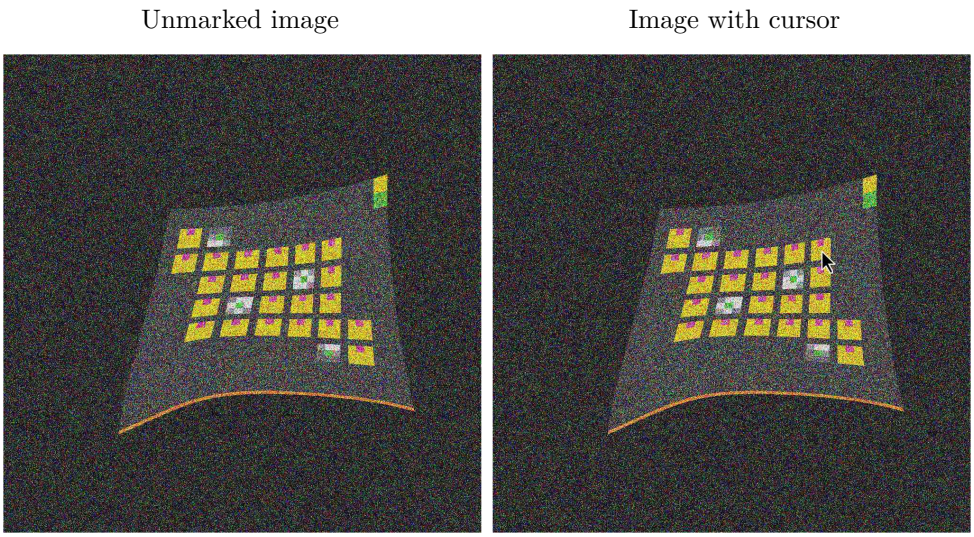


Fig. 2 Example output of `highlight_pixel` under strong nonlinear warping and pixel noise with $\sigma = 120$. The unmarked image (left) and cursor image (right) show the same game state under the same newly sampled geometric transformation, with independent pixel noise and JPEG quality 25. The cursor tip marks the transformed center of the queried board pixel. The action returns visual feedback about the proposed coordinate without executing a game action.

4 Evaluation Protocol and Metrics

Evaluation runs.

In the main study, we evaluate the same Stop-and-report agent with `gpt-5.6-sol` and `gpt-6-astra`, both at `max` reasoning effort. Each experimental condition contains one run on each of the 25 public ARC-AGI-3 games.

Appendix E describes the computational setup and adjustments to memory limits and runtime recovery for that study. The followup experiments and revised agent are described in Section 6.

Performance measures.

We report Relative Human Action Efficiency (RHAE), which combines level completion with action efficiency relative to human reference action counts [2]. We average per-game RHAE scores across all 25 games and separately within each action category. We also report the number of games completed in full.

Like moving a mouse cursor without clicking, the auxiliary `highlight_pixel` action does not count as a game action and is excluded from the action cost used by RHAE. Appendix D reports game actions, highlight calls, and inference costs separately.

Game categories.

We partition the 25 public ARC-AGI-3 games according to their initial available action sets, as exposed by the ARC-AGI Toolkit: *click-only* (7 games), *mixed keyboard-and-click* (12), and *keyboard-only* (6). Click-only games expose `ACTION6` but none of `ACTION1`–`ACTION5`; mixed games expose `ACTION6` and at least one of `ACTION1`–`ACTION5`; keyboard-only games do not expose `ACTION6`. `ACTION7` (Undo) is excluded from this categorization [2, 14]. Appendix C lists the games in each category. These labels describe the initially advertised interface, rather than the actions actually used during a run.

5 Results and Discussion: Main Study

Under the tested settings, geometric distortion impaired GPT-5.6-sol much more than pixel noise alone. Cursor feedback partly restored performance under homography and moderate nonlinear warping, with the largest gains in click-only games. Mean scores were also higher in keyboard-only games when cursor feedback was available. Under the strongest warp and noise setting, performance remained poor even with feedback. Using the same agent framework, GPT-6-astra achieved near-ceiling scores under both moderate and strong nonlinear warping. Table 5 groups the results by distortion type. Paired rows compare access to cursor feedback for the same model, noise level, and warp setting.

5.1 Pixel noise with GPT-5.6-sol

At $\sigma = 40$, overall RHAE remained close to the clean reference, increasing by 1.7 percentage points (Table 5, block A). This small difference does not establish a benefit from noise. Relative to the clean reference, RHAE declined by 6.8 points at $\sigma = 80$ and 17.4 points at $\sigma = 120$, with three fewer games completed at the strongest noise level. At $\sigma = 120$, click-only games showed the largest decline from the clean reference: 36.8 percentage points, compared with 11.1 for mixed games and 7.5 for keyboard-only games.

5.2 Projective distortion with GPT-5.6-sol

Random homography substantially reduced performance compared with the corresponding conditions without geometric distortion (Table 5, blocks A and B). Even without added Gaussian noise or JPEG compression, overall RHAE fell from about 91% to 50%. Click-only games showed the largest decline, although mixed and keyboard-only games were also affected.

Cursor feedback partly restored performance at every tested noise level and increased the number of completed games. The largest gains occurred in click-only

Table 5 Main-study results. All runs use the Stop-and-report agent at `max` reasoning effort. Scores are mean per-game RHAE (%). Category sizes appear in parentheses; completed games are out of 25.

			RHAE (%)				
σ	Model	Cursor feedback	**Overall**	Click-only (7)	Mixed (12)	Keyboard-only (6)	Completed (of 25)
A. No geometric distortion							
0	GPT-5.6-sol	No	**91.11**	97.86	86.27	92.91	23
40	GPT-5.6-sol	No	**92.77**	97.59	92.81	87.04	24
80	GPT-5.6-sol	No	**84.33**	95.47	77.41	85.20	23
120	GPT-5.6-sol	No	**73.68**	61.10	75.14	85.42	20
B. Random projective distortion (homography)							
0	GPT-5.6-sol	No	**49.79**	40.08	40.29	80.13	15
		Yes	**78.42**	90.86	67.21	86.31	21
40	GPT-5.6-sol	No	**53.33**	32.28	62.06	60.41	16
		Yes	**72.73**	82.94	67.09	72.11	20
120	GPT-5.6-sol	No	**36.33**	17.47	41.55	47.90	13
		Yes	**56.09**	67.63	49.95	54.91	17
C. Moderate nonlinear warping (FFD)							
40	GPT-5.6-sol	No	**37.25**	33.77	41.72	32.36	11
		Yes	**57.21**	64.14	53.74	56.07	18
40	GPT-6-astra	No	**99.50**	100.00	100.00	97.93	25
		Yes	**99.80**	100.00	100.00	99.17	25
120	GPT-5.6-sol	No	**24.04**	7.42	32.50	26.52	8
		Yes	**34.09**	34.41	34.57	32.76	10
120	GPT-6-astra	No	**95.80**	86.02	100.00	98.82	24
		Yes	**99.91**	100.00	100.00	99.60	25
D. Strong nonlinear warping (FFD)							
120	GPT-5.6-sol	No	**16.23**	7.20	23.66	11.91	5
		Yes	**14.48**	7.53	18.85	13.87	4
120	GPT-6-astra	No	**95.44**	85.76	100.00	97.59	24
		Yes	**99.53**	99.92	99.86	98.42	25

The first row is the *Clean-view* reference. All other rows without cursor feedback use *Distorted-view*; rows with feedback use *Cursor-feedback*. Images use PNG at $\sigma = 0$ and JPEG quality 25 otherwise. FFD denotes free-form deformation.

games, whose mean RHAE increased by approximately 50 percentage points in each comparison. Mean scores also increased in mixed and keyboard-only games.

Strong pixel noise remained a substantial difficulty even with cursor feedback. The highest noise level produced the lowest overall RHAE in both environments.

5.3 Moderate nonlinear warping with GPT-5.6-sol

The tested moderate FFD setting produced lower overall RHAE than homography at both noise strengths (Table 5, blocks B and C). Without cursor feedback, the reduction was approximately 16 percentage points at $\sigma = 40$ and 12 points at $\sigma = 120$. The two transformation settings were selected to be visually comparable in strength, although this comparison is subjective. At $\sigma = 40$ without feedback, FFD produced lower scores in mixed and keyboard-only games, while click-only scores remained similar to those under homography.

Cursor feedback increased overall RHAE by approximately 20 percentage points at $\sigma = 40$ and 10 points at $\sigma = 120$, and increased the number of completed games at both noise levels. The largest gains occurred in click-only games, although keyboard-only scores also improved. At stronger pixel noise, the click-category gain remained substantial, while gains in mixed and keyboard-only games were much smaller than at the lower noise level.

5.4 Strong nonlinear warping with GPT-5.6-sol

We increased the nonlinear warp strength to probe the limits of performance under severe visual distortion (Table 5, blocks C and D). Combined with pixel noise at $\sigma = 120$, strong FFD reduced GPT-5.6-sol's overall RHAE from 24.04% under moderate FFD to 16.23% without cursor feedback, with completed games falling from eight to five. Even cursor feedback did not restore performance: RHAE remained low at 14.48%, with four games completed. All three category scores remained low in both environments.

A plausible interpretation is that the combined distortion and noise made the observations too difficult for the model to use reliably, so additional coordinate feedback was insufficient to overcome the broader difficulty.

5.5 Nonlinear warping with GPT-6-astra

GPT-6-astra achieved near-ceiling RHAE under both moderate and strong nonlinear warping (Table 5, blocks C and D). With cursor feedback, it completed all 25 games in every tested condition. Without feedback, it completed all 25 under moderate warping at $\sigma = 40$, and 24 under either warp strength at $\sigma = 120$.

The only unfinished game was `ft09`. At $\sigma = 120$ without feedback, the agent completed only its first of six levels under both warp strengths. With feedback, it completed the game using only nine highlight calls in each run. Improvement on this one game accounts for most of the overall RHAE gain of approximately four percentage points in both comparisons.

The largest model gap occurred under strong warping at $\sigma = 120$ with feedback: approximately 85 percentage points. GPT-6-astra achieved 99.53% RHAE and completed all 25 games, whereas GPT-5.6-sol completed four. GPT-6-astra substantially outperformed GPT-5.6-sol in all settings tested with both models.

Section 6 examines whether this near-ceiling performance persists under more severe visual corruption.

5.6 Interpretation and limitations

A possible game-model explanation.

One possible explanation for GPT-6-astra's performance is that it builds an internal model of the game and uses it to interpret distorted observations and predict the effects of its actions. The behavioral results alone, however, do not establish whether such a model is constructed during interaction or how it contributes to successful play.

Feedback can provide more than a click-location cue.

Higher scores in *Cursor-feedback* were also observed for keyboard-only games. The auxiliary action could help the agent interpret the board in two ways. First, the cursor shows where a board coordinate chosen by the agent appears in the warped image. This provides a spatial reference that may help it understand the distortion. Second, each call returns fresh noisy and warped images of the same game state. Details obscured in one image may be clearer in another, even without using the cursor. Both benefits could be useful in games that do not require clicks. The current study does not establish how much each mechanism contributes to the observed gains.

Action efficiency does not measure all resources.

Highlight calls in the cursor-feedback experiments provide additional observations and consume inference without counting as game actions. Nevertheless, GPT-6-astra achieved higher RHAE with cursor feedback at approximately 8–16% lower normalized inference cost across all three comparisons (Appendix D).

At $\sigma = 120$, increasing FFD strength from moderate to strong changed GPT-6-astra RHAE by less than 0.4 percentage points with or without feedback, but increased normalized inference cost by approximately 20–23%. The agent therefore maintained near-ceiling performance under stronger warping at a higher inference cost.

Possible data contamination.

These experiments use public games, so the models may have encountered their rules or solutions during training. Such prior exposure could have contributed to the observed performance, even with warped and noisy images.

6 Results and Discussion: Followup Study on the Limits of GPT-6-astra

The near-ceiling scores in the main study raise the question of how much more severe the visual corruption can become before GPT-6-astra's performance declines substantially. We therefore conducted followup experiments on the same public games, first adding random rotation to strong nonlinear warping at $\sigma = 120$, then increasing pixel noise to $\sigma = 360$ and 1200 under strong warping without the additional rotation. All followup conditions use GPT-6-astra at `max` reasoning effort in the *Distorted-view* environment, *without cursor feedback*.

The extreme-noise experiments also exposed a problem in the agent's stuck protocol, which had rarely been invoked in our previous study [1]. We therefore evaluated a

revised Stop-and-report agent, described below. Table 6 presents results for the original and revised agents alongside the Baseline agent from that study and the main-study reference. Appendix D reports game actions and normalized inference costs for these conditions.

Table 6 Followup experiments with GPT-6-astra (max). All conditions use strong nonlinear warping, JPEG quality 25, and no cursor feedback. RHAE is the mean per-game score across 25 games. The first row repeats the main-study reference.

Condition	σ	Agent	**RHAE** (%)	Completed
Main-study reference	120	Stop-and-report	**95.44**	24/25
Random rotation	120	Stop-and-report	**93.06**	24/25
Extreme noise	360	Baseline	**57.43**	20/25
Extreme noise	360	Stop-and-report	**64.96**	18/25
Extreme noise	360	Stop-and-report3	**71.77**	21/25
Extreme noise	1200	Stop-and-report3	**17.37**	10/25

6.1 Random rotation under strong warping and noise

After each accepted game action, we independently sample an angle uniformly between 0° and 360° and rotate the board in the image plane about its center. We then apply nonlinear warping with the strong-FFD coefficient bounds in Table 4 and pixel noise at $\sigma = 120$. The additional padding required for rotation is described in Appendix B. The initial orientation is 0°. Intermediate frames returned by the same action share the rotation angle, while the nonlinear warp and pixel noise are independently resampled for each frame.

Adding random rotation reduced RHAE by 2.38 percentage points, while the number of completed games remained 24 of 25. As with stronger warping in the main study, the agent retained similar performance at a higher inference cost, using 26.5% more normalized cost tokens (Appendix D).

This suggests that random changes in orientation do not substantially confuse our agent. As in the main study, the agent has access to Python and general-purpose libraries. It therefore need not interpret every rotated image directly: it can recognize that the orientation has changed and use image-processing tools to rotate the image into a more useful view before inspecting it. Robustness to rotation may thus depend on the agent's ability to identify and compensate for the transformation, rather than on the underlying model's ability to understand images equally well at any orientation.

6.2 Extreme pixel noise under strong warping

We next increased the standard deviation of the added Gaussian pixel noise to $\sigma = 360$ and 1200, respectively three and ten times the strongest noise setting in the main

study. The strong nonlinear warp is still independently resampled for each frame. Consequently, simply stacking and averaging successive images would mix misaligned board locations. Combining observations to suppress noise requires estimating their spatial correspondence as well as accounting for changes in the game state.

Figure 3 illustrates the three noise levels. At $\sigma = 360$, small details are difficult to distinguish by direct visual inspection. Image processing, such as Gaussian smoothing, may make some features easier to recognize, but we have not established that it recovers every small detail required for a solution. At $\sigma = 1200$, even large structures are difficult to recognize from an individual image, although some information about the board remains.

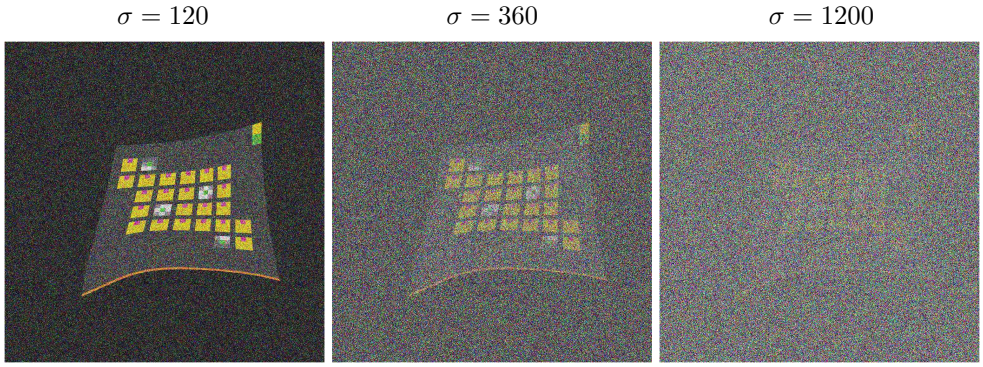


Fig. 3 Examples of strong nonlinear warping with increasing Gaussian pixel noise. The added noise standard deviations are 120, 360, and 1200 from left to right.

6.2.1 Revising the Stop-and-report stuck protocol

As described in our previous study [1], the stuck protocol is a fallback mechanism invoked when the agent returns control without taking any environment action. Its purpose is to prompt the agent to resume interaction and make progress. The protocol was rarely invoked in that study, and we did not notice an ambiguity in the message sent to the agent: it combines an instruction to stop when uncertain with a request to take meaningful actions. Appendix F reproduces the original and revised stuck-protocol messages.

In two separate experiments outside the studies reported here, we observed GPT-6-astra repeatedly return without taking an environment action after receiving this message. The runs eventually ended after exhausting the configured recovery attempts. We had not observed this behavior in the main study reported here.

Under strong warping with $\sigma = 360$, this behavior became more prominent, likely because the heavily corrupted observations more frequently left the agent uncertain how to proceed. In six games, `ls20`, `re86`, `s5i5`, `sk48`, `su15`, and `vc33`, Stop-and-report took only a few actions on the first level and then repeatedly returned without acting after the stuck reminder. Stopping can be reasonable when no further meaningful actions are possible. Here, however, the competing instructions in the stuck message appear

to have prevented further attempts. The revised agent described below subsequently made progress in all six games and completed four, showing that further progress was possible under these conditions.

Even with these early stops, Stop-and-report achieved higher mean RHAE than Baseline under the same conditions: 64.96% versus 57.43%. Its benefits therefore persisted despite the weakness in the stuck protocol. To explore how far the agent could progress under extreme visual corruption, we corrected this ambiguity and evaluated the revised agent.

We report results for both the original and revised agents. We call the revised agent *Stop-and-report3* (`agent_v1_08_wstop3`). The new reminder includes the explicit instruction:

> Take meaningful actions in the environment now. Do not stop or report back without taking meaningful actions.

This reminder is sent on its own, without the ordinary stop-and-report instruction. The controller permits up to five consecutive reminders without an environment action, instead of one, before exiting; the count resets after an action.

Stop-and-report3 remains environment-agnostic and differs from Stop-and-report only in its stuck protocol. Because that protocol was rarely invoked in our previous study, we expect the revised agent to retain Stop-and-report's performance on the other environments evaluated there. This expectation requires independent testing.

6.2.2 Results under extreme pixel noise

At $\sigma = 360$, the original Stop-and-report agent's RHAE was 30.5 percentage points below the main-study reference. Stop-and-report3 recovered 6.8 points and increased the number of completed games by three, to 21 of 25. Revising the stuck protocol therefore partly offset the decline under stronger noise, but did not restore the near-ceiling performance observed at $\sigma = 120$ in the main study.

At $\sigma = 1200$, Stop-and-report3 achieved 17.37% mean RHAE and completed ten games (Table 6). Among the completed games, `lf52` stood out with 99.38% RHAE. This demonstrates that near-ceiling performance remained possible in at least one game, even under extreme pixel noise and strong random nonlinear warping.

6.3 Investigating successful play at $\sigma = 1200$

The near-perfect score on `lf52` raises the question of how the agent could solve the game under such extreme noise and changing nonlinear warps. Prior exposure to the public game or its solution could have contributed, and we cannot exclude this without an appropriate evaluation on held-out games, which we do not have access to. Another possible contribution is active image processing followed by decisions under substantial uncertainty.

In the `lf52` run, the agent created 31 Python scripts and nearly 1,000 processed PNG images. Its scripts used various OpenCV functions to process the observations. The agent chose these processing procedures and output filenames itself. Its instructions gave it broad freedom to use the workspace (see [1, Section B.1.1]):

Use your workspace however you find most effective. You may create and modify files and directories, and write any scripts or utilities you need.

Figure 4 shows an example of a raw observation alongside two denoised and unwarped images found in the agent's workspace. The processed images make the board structure much easier to analyse. The complete artifacts for this `lf52` run, including files generated by the agent, are archived on Zenodo [4] for independent inspection.

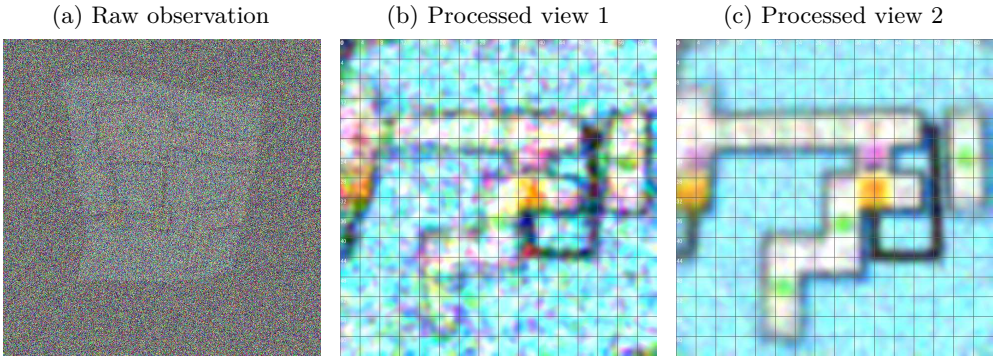


Fig. 4 Raw and processed views of `lf52` under strong nonlinear warping and $\sigma = 1200$ noise. (a) An example of a raw observation from level 6 supplied by the environment. (b,c) Denoised and unwarped images created and named by the agent: `level6walkway_now_grid_5.png` and `level6walkway_map_grid_5.png`, respectively. The coordinate grids were also added by the agent.

These artifacts support image processing as a plausible contributor to successful play, although we cannot exclude a contribution from prior familiarity with the public game. Nevertheless, the workspace provides a concrete example of an environment-agnostic agent developing tools during interaction to extract information from severely corrupted observations. Together with the successful play, this suggests that the agent may have combined image processing with decision-making under substantial uncertainty. Testing on held-out games would help determine whether this combination can produce similar performance on unfamiliar games.

7 Conclusion

We tested an environment-agnostic agent's ability to solve ARC-AGI-3 games from noisy and geometrically distorted images. Our previous study demonstrated that this agent could work unchanged across ARC-AGI-3, AGI Maze, and Terminal-Bench [1].

Geometric distortion makes clicking particularly demanding. In ordinary mouse interaction, a person can point directly at a visible target. Our agent has no such interface: it must supply coordinates on the original 64×64 board, even though the target appears at a distorted location. In additional experiments, we provided cursor feedback through the auxiliary `highlight_pixel` action. This action partly simulates mouse feedback: the agent proposes board coordinates and receives an image showing where they place a cursor. It can check and revise its guess, but still cannot click directly on the object in the image. The action provides visual feedback, not corrected coordinates.

For GPT-5.6-sol, geometric distortion reduced performance much more than pixel noise alone. Cursor feedback partly restored performance under homography and moderate nonlinear warping, with the largest gains in click-only games. Higher mean scores were also observed in keyboard-only games when feedback was available. Under the strongest tested combination of warping and noise in the main study, performance remained poor even with feedback.

GPT-6-astra achieved high performance even *without cursor feedback*: under the strongest tested combination of warping and pixel noise in the main study, our agent achieved 95.44% RHAE and completed 24 of the 25 public games. This included games requiring clicks in the original board coordinates, despite the agent seeing only distorted images. With feedback, it reached 99.53% RHAE and completed all 25 games.

We further explored the limits of GPT-6-astra, taking the strongest warping and noise settings from the main study as our starting point: strong nonlinear warping with $\sigma = 120$. All followup experiments used no cursor feedback. Adding a random rotation of the board about its center after each action reduced RHAE only modestly, from 95.44% to 93.06%. In separate experiments without the additional rotation, we retained strong nonlinear warping and increased the noise standard deviation to $\sigma = 360$ and 1200, respectively three and ten times the main-study setting. The independently changing warps complicate denoising by averaging successive images. These extreme conditions exposed a problem in the agent's stuck protocol: the agent could stop prematurely when uncertain how to proceed. We revised that protocol while keeping the agent environment-agnostic. Stop-and-report3 achieved 71.77% RHAE and completed 21 games at $\sigma = 360$. Performance declined substantially at $\sigma = 1200$, but the agent still achieved 17.37% RHAE and completed ten games.

Even at $\sigma = 1200$, the agent completed all ten levels of `lf52` with 99.38% RHAE. During this run, it created image-processing scripts and produced denoised and unwarped views of the observations. These artifacts suggest that successful play may involve image processing combined with decision-making under substantial uncertainty. They also illustrate how an environment-agnostic agent can develop image-processing tools during interaction without being specifically instructed to do so. This suggests a possible explanation for the modest effect of random rotation on performance: the

agent need not interpret every rotated image directly, but can recognize changes in orientation and use tools to compensate before inspecting the image.

Alongside image processing, the agent may build and use an internal model of the game. Such a model could help it combine information across corrupted observations and anticipate the effects of its actions. Whether and how the agent does this remains to be studied. Evaluation on held-out games would also help assess the contribution of prior familiarity with the public games.

Declarations

Funding. This study received no specific grant. The work was conducted while the author was employed as a researcher at SingularityNET.

Competing interests. The author is employed by SingularityNET and declares no other financial or non-financial competing interests.

Ethics approval and consent to participate. Not applicable. This study involved no recruitment of or interaction with human participants; no identifiable human data were accessed.

Consent for publication. Not applicable. This manuscript contains no identifiable personal data.

Clinical trial number. Not applicable.

Data availability. Per-game action records, scorecards, summary tables, and run configurations for both studies, together with the complete artifacts for the `lf52` run examined in Section 6.3, are archived on Zenodo [4]. The full collection of run artifacts is available in full or in any subset on request. Links to the released archives are provided through https://github.com/astroseger/envtraverse.

Code availability. Source code and documentation are available at https://github.com/astroseger/envtraverse. The exact software version used in this study is archived on Zenodo [3].

Author contributions. The author conceived and designed the study, implemented the experimental environments and agent modifications, conducted the experiments, analyzed the results, and wrote the manuscript.

Appendix A Environment descriptions supplied to the agent

A.1 Shared description for Clean-view and Distorted-view

Clean-view and *Distorted-view* use the description reproduced verbatim below. Its `ACTION6` convention explicitly states the 64×64 board size and upper-left origin. It supplies no information about the visual corruptions.

````
# Environment Description

## 1. General Description Of The Environment

You are playing a dynamic visual puzzle game. Your objective is to solve all levels using as few
actions as possible.

## 2. Actions

Actions are JSON objects.

Simple actions:

```json
{"id":"ACTION1"}
{"id":"ACTION2"}
{"id":"ACTION3"}
{"id":"ACTION4"}
{"id":"ACTION5"}
{"id":"ACTION7"}
{"id":"RESET"}
```

Coordinate action:

```json
{"id":"ACTION6","data":{"x":12,"y":34}}
```

General action conventions:

- `ACTION1`: usually direction-like, commonly up.
- `ACTION2`: usually direction-like, commonly down.
- `ACTION3`: usually direction-like, commonly left.
- `ACTION4`: usually direction-like, commonly right.
- `ACTION5`: game-specific interaction, such as use, select, rotate, attach, detach, confirm, or
execute.
- `ACTION6`: coordinate-based interaction; provide `x` and `y`, with the origin at the top-left.
The board size is 64x64.
- `ACTION7`: usually undo.
- `RESET`: reset the current level without resetting the action count.
````

A.2 Additions for Cursor-feedback

Cursor-feedback changes the shared description in only three places: `Coordinate action:` becomes `Coordinate actions:`, and the following JSON example and action convention are inserted immediately after their `ACTION6` counterparts:

```
{"id":"highlight_pixel","data":{"x":12,"y":34}}
```

```
- `highlight_pixel`: auxiliary action. Highlight the selected board pixel. Does not consume the
game-solving budget.
```

All other text is unchanged. The description does not explain the two-image response, its shared geometry, or its independently sampled pixel noise; these are properties of the interface described in Section 3.3.

Appendix B Implementation details for geometric distortions

The following choices specify the random transformations used in this study in addition to the parameter ranges in Tables 3 and 4.

Homography implementation.

We apply perspective, scaling, horizontal shear, and rotation, in that order, about the center of the unpadded image, followed by translation. The perspective terms use horizontal and vertical coordinates measured from the image center and divided by half the image width and height, respectively. The horizontal shear coefficient is the tangent of the sampled angle. Horizontal and vertical translations are expressed as fractions of the unpadded image width and height, respectively.

FFD implementation.

Each FFD field uses tensor-product cubic B-splines with open-uniform, clamped knot vectors. Horizontal and vertical coordinates are each normalized from zero to one across the padded canvas. Both control grids cover this canvas, and their displacement fields are added. Boundary control coefficients are sampled in the same way as interior coefficients; the displacement is therefore not constrained to zero at the canvas boundary. The coefficient bounds remain relative to the unpadded image's 512-pixel side length.

Deformation check.

The local deformation check evaluates the Jacobian determinant of the combined FFD backward sampling map at every output-pixel center. A sample is accepted only if all evaluated determinants are at least 0.1; otherwise, both grids are redrawn together. This is a discrete sampling check rather than a guarantee of global invertibility.

Padding and interpolation.

Padding is fixed within each condition and determined by the transformation bounds, interpolation support, and a safety margin. Output canvases are 800×800 pixels for homography, 688×688 for moderate FFD, and 1008×1008 for strong FFD without the additional rotation; without geometric distortion they remain 512×512. Both spline grids span the padded canvas, so strong FFD enlarges their spatial domain as well as their coefficient bounds.

For strong FFD with random rotation (Section 6.1), the canvas is enlarged to 1216×1216 pixels to accommodate all orientations. The strong-FFD coefficient bounds remain unchanged, but both spline grids span this larger padded canvas.

Rendering uses backward sampling: each output pixel is assigned a sampling location in the enlarged source image. For a homography, this location is obtained through the inverse transformation. For FFD, the summed displacement is added to the output position after accounting for the canvas-padding offset. We use bilinear interpolation, treating pixels outside the source image as black. Gaussian noise is added after interpolation, including to the padding. Intensities are clipped to the 0–255 range and rounded before encoding using the settings in Section 3.2.2.

Cursor rendering.

For `highlight_pixel`, the cursor is overlaid after warping and Gaussian noise, before image encoding. Its tip marks the transformed center of the queried board pixel. The graphic remains upright and fixed in output-pixel size; only JPEG encoding can alter its appearance.

Appendix C Games in each action category

Table C1 gives the partition used for the category means. Categories are based on the initial available actions, using the definitions in Section 4; `ACTION7` is excluded.

Table C1 The 25 public games grouped by their initial available action sets.

Category	Games
Click-only (7)	`ft09, lp85, r11l, s5i5, su15, tn36, vc33`
Mixed (12)	`ar25, bp35, cd82, cn04, dc22, ka59, lf52, m0r0, sb26, sc25, sk48, sp80`
Keyboard-only (6)	`g50t, ls20, re86, tr87, tu93, wa30`

Appendix D Actions and inference costs

Tables D2 and D3 report game actions and normalized inference costs for the main and followup studies, respectively. Table D2 also reports cursor-feedback calls separately from game actions.

Table D2 Main-study resource use, with conditions ordered as in Table 5. All runs use the Stop-and-report agent at `max` reasoning effort. Each row totals one run on each of the 25 games.

σ	Model	Cursor feedback	Game actions	Highlight calls	Cost tokens (M)
A. No geometric distortion					
0	GPT-5.6-sol	No	15,124	0	17.67
40	GPT-5.6-sol	No	14,550	0	17.49
80	GPT-5.6-sol	No	18,274	0	21.71
120	GPT-5.6-sol	No	22,144	0	23.00
B. Random projective distortion (homography)					
0	GPT-5.6-sol	No	31,107	0	35.16
		Yes	18,420	15,977	27.31
40	GPT-5.6-sol	No	33,759	0	41.83
		Yes	23,421	1,695	39.57
120	GPT-5.6-sol	No	34,971	0	45.99
		Yes	31,651	6,220	49.28
C. Moderate nonlinear warping (FFD)					
40	GPT-5.6-sol	No	37,271	0	48.78
		Yes	32,881	3,603	59.53
40	GPT-6-astra	No	7,662	0	6.94
		Yes	7,342	306	6.40
120	GPT-5.6-sol	No	53,249	0	74.60
		Yes	43,853	18,348	73.06
120	GPT-6-astra	No	9,986	0	9.07
		Yes	7,801	391	7.63
D. Strong nonlinear warping (FFD)					
120	GPT-5.6-sol	No	48,819	0	89.89
		Yes	47,473	37,682	64.28
120	GPT-6-astra	No	10,528	0	10.85
		Yes	9,050	1,036	9.38

The first row is the *Clean-view* reference; other rows use *Distorted-view* or *Cursor-feedback* according to feedback availability. Game actions are scorecard counts, including resets and excluding `highlight_pixel`. Highlight calls are counted separately from the recorded applied actions. Cost tokens weight output, uncached input, and cached input tokens by 1, 1/6, and 1/60, respectively, and are reported in millions [1]. This is a proxy for inference cost, not a dollar cost or a measure of computational work.

Table D3 Resource use in the followup study, with the main-study reference repeated for comparison. All conditions use GPT-6-astra at `max` reasoning effort, strong nonlinear warping, and no cursor feedback. Each row totals one run on each of the 25 games.

Condition	σ	Agent	Game actions	Cost tokens (M)
Main-study reference	120	Stop-and-report	10,528	10.85
Random rotation	120	Stop-and-report	12,438	13.73
Extreme noise	360	Baseline	34,424	15.78
Extreme noise	360	Stop-and-report	13,430	25.02
Extreme noise	360	Stop-and-report3	28,333	24.91
Extreme noise	1200	Stop-and-report3	51,209	46.31

Game actions include resets. Cost tokens use the same normalization as Table D2 and are reported in millions.

Appendix E Computational setup and runtime adjustments

This appendix gives the dates of both studies and describes the computational setup and runtime adjustments for the main study.

Computational setup.

The main study ran from 21 to 26 October 2026, and the followup study from 2 to 5 November 2026. For the main study, we used eight Codex accounts with $200 Pro subscriptions and ran up to 24 games concurrently on a machine with two AMD EPYC 9654 processors and 3 TB of RAM. These specifications describe the hardware used, not minimum requirements.

Runtime recovery.

Initial tests were disrupted by OpenAI service errors reporting insufficient capacity. To avoid exhausting recovery attempts during temporary outages, we introduced a 15-minute delay before each recovery attempt and increased the recovery allowance from 10 to 50 attempts per game run in most run configurations.[1] We did not observe these capacity problems during the main study.

Memory limits.

Noisy images compress poorly and produce larger observation files. Preliminary tests encountered memory-allocation errors in Codex under the limits used in our previous study [1]. We therefore increased the limits as shown in Table E4. These limits were deliberately generous to avoid further allocation failures. We did not determine the minimum memory needed.

[1]Twenty of the 22 saved run configurations explicitly specify 50 recovery attempts. The Clean-view reference and GPT-5.6-sol with moderate FFD, $\sigma = 40$, and cursor feedback omit this setting; the runner default is 10.

Table E4 Configured memory limits in the previous study and the main study.

Configuration	Per-process data limit (GiB)	Container memory limit (GiB)
Previous study	2	10
Main study, except strong FFD	16	32
Strong FFD with $\sigma = 120$	32	64

Appendix F Stuck-protocol messages

This appendix reproduces the complete messages used by the original and revised stuck protocols discussed in Section 6.2.1.

F.1 Original Stop-and-report agent

The original agent (`agent_v1_01_wstop`) combines the ordinary continuation instruction, the stop-and-report instruction, and the action reminder into the following message:

```
Keep working toward the goal described in `/observations/environment_description.md`. Re-read
the file whenever you need a reminder.

# When to stop
Continue working on the task as long as you can proceed confidently. If you encounter a problem,
ambiguity, unexpected issue, or anything that makes you unsure how to proceed, stop and report
it back to me before taking further action.

You have not sent any environment actions since the previous continuation request. Continue
working and send meaningful actions toward completing your task. If you do not make progress,
the run will stop.
```

F.2 Stop-and-report3

The revised agent (`agent_v1_08_wstop3`) sends the following reminder on its own during the stuck protocol:

```
You already have all the information you need to solve the problem you're facing. Take some time
to think carefully about your options, choose the best approach, and move forward. I'm confident
you can make progress!

**Take meaningful actions in the environment now. Do not stop or report back without taking
meaningful actions.** Reporting without acting will result in task failure.
```

References

[1] Rodionov, S.: Improving Environment-Agnostic Agents: The Role of Scaffolding. arXiv:2610.22289 (2026). https://doi.org/10.48550/arXiv.2610.22289 . https://arxiv.org/abs/2610.22289

[2] ARC Prize Foundation: ARC-AGI-3: A New Challenge for Frontier Agentic Intelligence. arXiv:2603.24621 (2026). https://doi.org/10.48550/arXiv.2603.24621 . https://arxiv.org/abs/2603.24621

[3] Rodionov, S.: astroseger/envtraverse: paper04-artifact-v2. Zenodo. Software (2026). https://doi.org/10.5281/zenodo.24032930 . https://doi.org/10.5281/zenodo.24032930

[4] Rodionov, S.: Supporting Data and Run Artifacts for "Can Environment-Agnostic Agents Solve ARC-AGI-3 from Warped and Noisy Images?". Zenodo. Dataset (2026). https://doi.org/10.5281/zenodo.24021388 . https://doi.org/10.5281/zenodo.24021388

[5] Han, Q., Hu, K., Qiu, L., Wu, C., He, K.: VISTA: A Visual Harness for Reasoning in an Interactive World. arXiv:2610.02200 (2026). https://doi.org/10.48550/arXiv.2610.02200 . https://arxiv.org/abs/2610.02200

[6] Saxena, R., Suglia, A., Minervini, P.: VLM-RobustBench: A Comprehensive Benchmark for Robustness of Vision-Language Models. arXiv:2603.06148 (2026). https://doi.org/10.48550/arXiv.2603.06148 . https://arxiv.org/abs/2603.06148

[7] Zhao, H., Chen, T., Wang, Z.: On the robustness of GUI grounding models against image attacks. In: Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition (CVPR) Workshops, pp. 1618–1623 (2025). https://openaccess.thecvf.com/content/CVPR2025W/TMM-OpenWorld/html/Zhao_On_the_Robustness_of_GUI_Grounding_Models_Against_Image_Attacks_CVPRW_2025_paper.html

[8] Chattopadhyay, P., Hoffman, J., Mottaghi, R., Kembhavi, A.: RobustNav: Towards benchmarking robustness in embodied navigation. In: Proceedings of the IEEE/CVF International Conference on Computer Vision (ICCV), pp. 15691–15700 (2021). https://openaccess.thecvf.com/content/ICCV2021/html/Chattopadhyay_RobustNav_Towards_Benchmarking_Robustness_in_Embodied_Navigation_ICCV_2021_paper.html

[9] Sun, J., Zhu, J., Li, Y., Liu, T., Hu, X., Han, B.: AgentHijack: Benchmarking Computer Use Agent Robustness to Common Environment Corruptions. arXiv:2605.25707 (2026). https://doi.org/10.48550/arXiv.2605.25707 . https://arxiv.org/abs/2605.25707

[10] Zhao, Y., Chen, W.-N., Inan, H.A., Kessler, S., Wang, L., Wutschitz, L., Yang,

F., Zhang, C., Minervini, P., Rajmohan, S., Sim, R.: Learning GUI Grounding with Spatial Reasoning from Visual Feedback. arXiv:2509.21552 (2025). https://doi.org/10.48550/arXiv.2509.21552 . https://arxiv.org/abs/2509.21552

[11] Mittal, H., Mittal, G., Troncoso, N.D., Hu, Y.: PrecisionCUA: Iterative Visual Refinement for Pixel-Precise Cursor Grounding in Code Editors. arXiv:2604.13019 (2026). https://doi.org/10.48550/arXiv.2604.13019 . https://arxiv.org/abs/2604.13019

[12] Hartley, R.I., Zisserman, A.: Multiple View Geometry in Computer Vision, 2nd edn. Cambridge University Press, Cambridge (2004)

[13] Rueckert, D., Sonoda, L.I., Hayes, C., Hill, D.L.G., Leach, M.O., Hawkes, D.J.: Nonrigid registration using free-form deformations: Application to breast MR images. IEEE Transactions on Medical Imaging **18**(8), 712–721 (1999) https://doi.org/10.1109/42.796284

[14] ARC Prize Foundation: ARC-AGI Toolkit. Software repository (2026). https://github.com/arcprize/ARC-AGI