MNIST-PRO: MNIST IS BACK AS A PARTIALLY OBSERVABLE WORLD FOR AI AGENTS

Anonymous authors
Paper under double-blind review

ABSTRACT

Agentic perception requires agents in partially observable environments to coordinate active sensing with working memory to construct and maintain an evolving perceptual state. However, existing benchmarks struggle to isolate this capability because they couple active vision with complex embodied navigation and control dynamics. To isolate this capability, we introduce `MNIST-PRO`, which converts digit recognition into a sequential search task with controlled access to past glimpses. Evaluating ten multimodal models across in-context memory representations reveals that accuracy drops from up to 99.0% on unmasked images to between 7.0% and 43.5% under partial observability. We identify three bottlenecks in the perception loop. First, agents often fail to search thoroughly, making premature guesses based on partial strokes instead of fully utilizing their exploration budget. Second, even when agents uncover most digit strokes, they struggle to combine separate glimpses in memory, a limitation that visual canvas reconstruction overcomes by recovering over 50 percentage points on two-digit tasks. Third, even with near-complete coverage on an assembled canvas, agents still make visual interpretation errors, showing that downstream recognition remains an independent hurdle. Additionally, equipping agents with persistent memory in a harness fails to improve accuracy, instead causing them to waste steps repeatedly re-checking strokes they have already seen.

1 INTRODUCTION

In partially observable environments, multimodal agents must coordinate active sensing with working memory to construct a persistent perceptual state. Unlike static visual reasoning, where complete inputs are provided upfront, agentic perception requires an agent to decide where to look next, gather localized observations, and integrate incoming glimpses with past beliefs to resolve uncertainty (Wu & Xie, 2023; Xu et al., 2026; Yang et al., 2025). This capability is essential for downstream tasks such as graphical user interface interaction (Zhou et al., 2024; Xie et al., 2024) and robotic manipulation (Savva et al., 2019; Brohan et al., 2023).

Evaluating how agents maintain these internal representations remains difficult because existing benchmarks confound perceptual state construction with control and retrieval dynamics. Standard multimodal benchmarks like MMBench (Liu et al., 2024) and MMMU (Yue et al., 2024) provide complete visual evidence upfront, removing the need for active search. Active control benchmarks like ActiView (Wang et al., 2025) and ActiveVision (Zhang et al., 2026) require iterative gaze shifts, but allow unconstrained visual lookback. This setup tests image retrieval from full history rather than an agent's ability to maintain and update an evolving belief state. In contrast, embodied 3D benchmarks like ALFRED (Shridhar et al., 2020) require sequential actions, but their navigation and manipulation tasks introduce physical execution errors that obscure whether failures stem from control, perception, or memory.

To evaluate how agents search, remember, and interpret visual evidence without physical control errors, we introduce `MNIST-PRO`, a diagnostic benchmark formulated as a Partially Observable Markov Decision Process. In `MNIST-PRO`, an agent receives localized visual glimpses rather than full images, requiring active search and observation integration across sequential steps. We build our benchmark on MNIST digits (Lecun et al., 1998), where current multimodal models exceed 90% accuracy under full observability, establishing an empirical ceiling for passive recognition. Us-

Reviewers: please read the Reviewer Guidelines (iclr.cc/Conferences/2027/ReviewerGuidelines) and the AI Policy for Reviewers (iclr.cc/Conferences/2027/AIPolicyForReviewers). If you used AI to expand, edit, or polish your review, please provide the input text to the LLM. Better still, consider skipping the LLM and submitting your original text: we, and the authors, are much more interested in your unedited thoughts than in what an LLM has to say. AI-assisted or not, you are putting your name and reputation behind your review: LLM-generated falsehoods, hallucinations or misrepresentations are subject to disciplinary action, which may include desk-rejecting all papers you have authored.

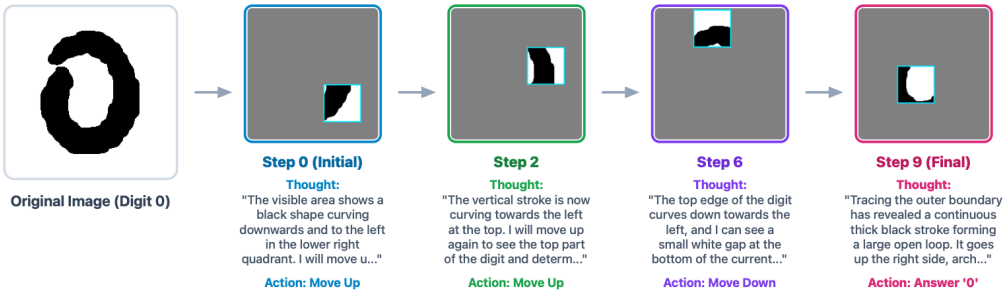


Figure 1: **Example episode in `MNIST-PRO`.** The leftmost panel shows the original hidden image of Digit 0, which the agent can only access through a glimpse window. To solve the task, the agent moves the glimpse window across the image and updates its beliefs as it gathers partial observations. At Step 0, the agent begins with a stroke-anchored glimpse, stores the observation in memory, and selects an action. It moves upward at Steps 0 and 2, then downward at Step 6. By Step 9, after tracing stroke boundaries and integrating observations, the agent predicts the label *0*.

ing simple digit canvases eliminates physical control errors and visual clutter, isolating how agents search, track coverage, and integrate spatial memory. We structure the benchmark across two difficulty tiers. Level 1 evaluates exploration on a single centered digit. Level 2 places two digits side by side, requiring agents to cross empty background space to find both digits and predict their order.

A key feature of `MNIST-PRO` is the ability to vary visual history length $|\mathcal{H}_v|$, distinguishing between maintaining an evolving state and simply reviewing past images. By setting visual lookback to $|\mathcal{H}_v| = 1$, the agent receives only the current frame and must maintain an evolving state representation. We evaluate representations across three paradigms, internal in-context memory, externalized visual canvases, and autonomous agent harnesses. Internal in-context memory includes raw visual histories, qualitative textual states, and structured metric grid maps. Externalized canvases provide environment-assisted image stitching, while harnesses allow agents to write their own reconstruction tools. Unlike interactive benchmarks like ARC-AGI-3 (Foundation, 2026), where agents must deduce hidden environment mechanics, `MNIST-PRO` provides explicit rules and deterministic movement actions. Because task rules and dynamics are fully transparent, performance drops reflect limitations in agentic perception rather than rule comprehension.

We summarize our core contributions below.

1. We introduce `MNIST-PRO`, an agentic perception benchmark that isolates visual state construction and active exploration under controlled visual lookback and finite exploration budgets.
2. Across ten multimodal models, we find a large performance drop under partial observability. While frontier models exceed 90% accuracy on unmasked images, reaching up to 99.0%, their accuracy falls to between 7.0% and 43.5% when relying on in-context memory of local glimpses.
3. We identify an evidence collection bottleneck where agents terminate search prematurely. In an audit of Level 2 trajectories, over 20% of trials end with only a single digit prediction, and agents fail to locate both digits in up to 62% of zero lookback episodes, dropping accuracy on those runs to 0.0%.
4. We identify an evidence representation bottleneck where agents fail to combine disjoint glimpses in working memory, a limitation that programmatic visual canvas assembly overcomes by recovering over 50 percentage points on two-digit tasks.
5. We show that downstream visual interpretation remains an independent hurdle even with stroke coverage above 98%, where swapping in a stronger classifier on identical trajectories improves accuracy by up to 17.8 percentage points.
6. We demonstrate that equipping agents with persistent memory across episodes in harnesses fails to improve performance, reaching 73.5% accuracy with memory compared to 75.5% without memory, while triggering circular verification loops that increase exploration steps by over 25%.

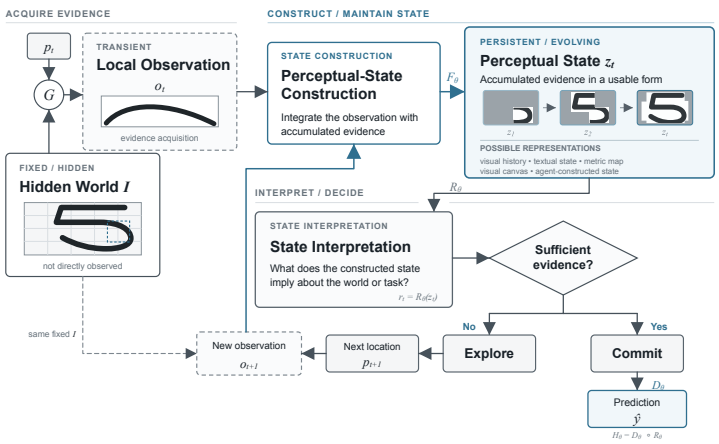


Figure 2: Perceptual-state construction under partial observability. Evidence acquisition, state construction, and state interpretation are modeled as separate stages of the agent loop. The agent explores until it commits to a prediction.

2 MNIST-PRO

MNIST-PRO evaluates how multimodal agents coordinate exploration with working memory under partial observability, as shown in Figure 1. By masking digit canvases with a movable glimpse window, the environment isolates spatial exploration and belief state construction while minimizing low-level perceptual noise. Because models achieve up to 99.0% accuracy on unmasked digits, task errors isolate limitations in active search and memory integration rather than passive visual recognition.

2.1 POMDP Formulation

We formalize active glimpse perception as a discrete POMDP. The state space is $\mathcal{S} = \mathcal{I} \times \mathcal{P}$, where $I \in \mathcal{I}$ is a static canvas with dimensions $H \times W'$ and $p_t = (x_t, y_t) \in \mathcal{P}$ is the top-left coordinate of an $h \times w$ glimpse window. Because the canvas is unobservable, the agent receives a partial observation $o_t = \Omega(I, p_t) \in \mathcal{O}$ at each step. This observation has dimensions $H \times W'$, with regions outside the glimpse window masked in gray and a 2-pixel cyan border outlining the window for spatial reference. To avoid searching empty background, initial position p_0 is centered on a digit stroke so that each episode begins with visual evidence. The action space $\mathcal{A} = \mathcal{A}_{\text{move}} \cup \mathcal{A}_{\text{predict}}$ includes movement and prediction. Movement actions $\mathcal{A}_{\text{move}} = \{\text{up}, \text{down}, \text{left}, \text{right}\}$ shift the glimpse window by δ pixels. Transitions are deterministic. The canvas remains static with $I_{t+1} = I_t$, while window coordinates update through clipping according to $p_{t+1} = \text{clip}(p_t + \delta \cdot u(a_t))$, where $u(a_t)$ is the unit displacement vector for action a_t. Prediction action $\mathcal{A}_{\text{predict}}$ outputs a predicted label sequence $\hat{y} \in \mathcal{Y}$ and ends the episode. Episodes terminate when the step count reaches budget $T_{\max}$. Section A.3 provides further details on the POMDP formulation.

2.2 The Required Agentic Loop

To solve MNIST-PRO, an agent must coordinate sensing with belief tracking across evidence collection, state construction, and state interpretation, as illustrated in Figure 2. At step t, the agent receives observation o_t, updates belief state z_t, and decides whether to output prediction $\hat{y}$ or execute movement action a_t. Testing how z_t is maintained and interpreted isolates whether failures stem from incomplete exploration or incorrect interpretation of evidence.

2.3 Benchmark Task Levels and Complexity

We evaluate MNIST-PRO across two difficulty levels designed to isolate local stroke integration from sequential multi-target search. Under **Level 1 Single Digit**, the agent explores a centered

digit on a canvas, isolating baseline capability in tracking local glimpses along stroke contours and consolidating disjoint visual features into a coherent digit. Under **Level 2 Two Digits**, two independent digits are horizontally concatenated onto a canvas, requiring the agent to traverse empty background gaps, locate and recognize both digits, and preserve their relative spatial order to predict sequences such as `58`.

2.4 A Taxonomy of Memory for Multimodal Agents

To evaluate how memory architecture shapes agentic perception, we benchmark configurations across three distinct paradigms, with schemas, coordinate frames, and harness specifications detailed in Sections A.4 to A.6 and E.1.

1. Internal In-Context Memory. The model maintains its belief state z_t entirely within its textual context window across interactive turns. **(i) Image-Only Baseline ($|\mathcal{H}_v| = \infty$):** The model receives the full chronological sequence of past visual glimpses and actions. This baseline isolates the model's ability to perform path integration from visual history without relying on intermediate textual representations. **(ii) Textual State ($|\mathcal{H}_v| = 1$):** Operating under strict zero visual lookback, the model receives only the current glimpse o_t and its own accumulated history of past actions and self-generated JSON thought reflections, directly testing natural language working memory. **(iii) Metric Grid Map ($|\mathcal{H}_v| = 1$):** Under zero visual lookback, this representation extends textual working memory by prompting the agent to maintain an explicit two-dimensional relative coordinate grid of visited locations and observed stroke features, testing symbolic spatial path integration.

2. Externalized State Construction. Spatial tracking and image stitching are offloaded from the model context window to an external rendering buffer maintained by the environment. **(i) Online Visual Memory Canvas ($|\mathcal{H}_v| = 1$):** Glimpses are dynamically projected onto a global composite canvas C_t at each step under $|\mathcal{H}_v| = 1$. The model receives only this canvas and the current glimpse o_t, maintaining a fixed context of two images regardless of episode length. **(ii) Offline Visual Memory Canvas:** Trajectories recorded under Image-Only or Textual State are stitched post-hoc into a finalized canvas C_T and supplied exclusively at the final prediction step, decoupling exploratory search policy from visual state interpretation.

3. Agent Harness. To test whether autonomous agents can manage their own state and memory, we place them in a sandbox with shell access, Python execution, and three environment tools, namely `move`, `view_image`, and `submit`. Agents receive observations both in-context and as local PNG files, allowing them to write custom scripts to stitch and visually inspect images. We also test cross-episode persistence across three configurations: (i) starting each episode from scratch, (ii) carrying a persistent `NOTES.md` log across episodes, and (iii) updating the log with correctness feedback after each submission.

3 Experiments and Results

Models and Experimental Settings. We evaluate eight proprietary models across the Gemini, Claude, and GPT families, alongside two open-source models, Qwen-3.8-27B (Qwen Team, 2026) and GLM-4.6V (Team et al., 2025). All models are generally queried with the default API configurations, including temperature and thinking levels. Only GPT-5.6-Sol is set to *xhigh* reasoning effort, denoted as GPT-5.6-Sol (xhigh), in selected experiments to study the impact of scaling reasoning budgets. We evaluate these models across 100 episodes per task with a 64×64 observation window and a step size of $\delta = 32$. The budget $T_{\max}$ is set to 36 steps for Level 1 224×224 images and 78 steps for Level 2 224×448 images, as detailed in Section A. We track three core metrics, namely *control accuracy* as an upper bound defined as the classification accuracy on fully-unmasked images, *classification accuracy* under partial observability, and *average steps* as the mean exploration cost.

3.1 Native Multi-turn Baseline

We evaluate a multi-turn baseline where models rely on standard conversational history, with complete results in Section B. Across models, greater exploration does not ensure higher accuracy. While

Table 1: Model performances on MNIST-PRO across three in-context memory configurations. Each memory and model combination is evaluated in terms of accuracy and steps used for Level 1 Single Digit and Level 2 Two Digits tasks. For each model, **bold** indicates the memory with the best average accuracy.

Model	Control Acc. (%) ↑			Image Only ($\|\mathcal{H}_v\|=\infty$) Acc. (%) ↑			Image Only ($\|\mathcal{H}_v\|=\infty$) Avg. Steps ↓			Textual State ($\|\mathcal{H}_v\|=1$) Acc. (%) ↑			Textual State ($\|\mathcal{H}_v\|=1$) Avg. Steps ↓			Metric Grid Map ($\|\mathcal{H}_v\|=1$) Acc. (%) ↑			Metric Grid Map ($\|\mathcal{H}_v\|=1$) Avg. Steps ↓		
	L1	L2	Avg	L1	L2	Avg	L1	L2	Avg	L1	L2	Avg	L1	L2	Avg	L1	L2	Avg	L1	L2	Avg
Proprietary Models																					
GEMINI-3.1-PRO-PREVIEW	99.0	97.0	98.0	38.0	5.0	21.5	4.52	11.95	8.23	56.0	20.0	**38.0**	7.18	18.28	12.73	53.0	14.0	33.5	7.15	19.19	13.17
GEMINI-3.6-FLASH	96.0	96.0	96.0	53.0	18.0	**35.5**	5.62	15.11	10.37	47.0	15.0	31.0	9.35	39.29	24.32	49.0	10.0	29.5	7.92	20.44	14.18
GEMINI-3.7-FLASH	98.0	97.0	97.5	65.0	22.0	**43.5**	5.26	14.69	9.97	54.0	6.0	30.0	7.92	34.84	21.38	57.0	28.0	42.5	6.82	20.37	13.60
CLAUDE-5-SONNET	94.0	90.0	92.0	24.0	1.0	**12.5**	17.35	45.15	31.25	22.0	2.0	12.0	14.79	75.64	45.22	19.0	2.0	10.5	13.07	50.39	31.73
CLAUDE-5-OPUS	93.0	91.0	92.0	43.0	13.0	**28.0**	7.43	22.50	14.96	41.0	13.0	27.0	11.61	39.90	25.75	36.0	15.0	25.5	10.57	29.55	20.06
CLAUDE-5-FABLE	95.0	89.0	92.0	47.0	15.0	31.0	7.27	22.30	14.79	38.0	18.0	28.0	11.30	35.93	23.62	45.0	23.0	**34.0**	9.92	28.62	19.27
GPT-5.6-TERRA	83.0	62.0	72.5	35.0	1.0	**18.0**	18.19	39.67	28.93	25.0	0.0	12.5	24.09	69.67	46.88	24.0	0.0	12.0	21.92	62.43	42.17
GPT-5.6-SOL	88.0	82.0	85.0	37.0	1.0	**19.0**	6.53	16.98	11.76	29.0	0.0	14.5	7.01	22.52	14.77	32.0	1.0	16.5	7.89	21.26	14.58
GPT-5.6-SOL (xhigh)	92.0	79.0	85.5	56.0	29.0	**42.5**	9.59	35.43	22.51	37.0	3.0	20.0	9.19	28.00	18.59	39.0	5.0	22.0	9.79	36.09	22.94
Open-Source Models																					
QWEN-3.8-27B	89.0	87.0	88.0	28.0	2.0	15.0	4.76	9.19	6.97	31.0	1.0	16.0	6.97	16.94	11.96	36.0	0.0	**18.0**	6.86	13.87	10.37
GLM-4.6V	91.0	92.0	91.5	14.0	0.0	7.0	3.49	6.51	5.00	20.0	1.0	**10.5**	20.24	40.08	30.16	20.0	0.0	10.0	11.39	27.67	19.53

GEMINI-3.7-FLASH leads with 75.0% accuracy on Level 1 and 47.0% on Level 2, CLAUDE-5-SONNET over-explores across an average of 43.24 steps on Level 2 yet achieves 0.0%. Meanwhile, open-source models stop prematurely, averaging only 2.76 to 4.62 steps on Level 1. Removing conversational traces to isolate visual path integration in Table 1 reduces GEMINI-3.7-FLASH to 65.0% on Level 1 and 22.0% on Level 2 under the Image-Only condition. Because these baselines confound active search with working memory, we next benchmark structured internal state representations.

3.2 RESULTS FOR DIFFERENT INTERNAL STATE CONSTRUCTION MECHANISMS

Passive Recognition Does Not Guarantee Active State Construction. As evident in Table 1, high recognition accuracy under full observability does not ensure successful perceptual state construction. Although all models achieve 83.0% to 99.0% control accuracy on single digits and 62.0% to 97.0% on two-digit sequences in Table 1, performance drops under partial observability. Even with an unbounded visual history of $|\mathcal{H}_v| = \infty$, GEMINI-3.1-PRO-PREVIEW falls from 99.0% control accuracy to 38.0%, and CLAUDE-5-SONNET drops from 94.0% to 24.0%, demonstrating that passive recognition capability fails to translate into active evidence gathering.

Memory Representations Help and Hurt Differently. Without a visual history ($|\mathcal{H}_v| = 1$), models rely on written memory to preserve relevant information, using either a *Textual State* or a *Metric Grid Map*. Adding coordinate tracking affects models differently. For example, GEMINI-3.1-PRO-PREVIEW reaches 56.0% accuracy with textual reasoning alone, but drops to 53.0% when required to additionally maintain a grid map. Conversely, GEMINI-3.7-FLASH benefits from spatial tracking, improving from 54.0% with textual memory to 57.0% with the grid map. Thus, tracking coordinates is not always helpful. The added work of path integration can degrade performance for some models, while giving others a stable geometric state construction that improves their predictions.

Two-Digit Performance Drop and Premature Stopping. Scaling the task from single digits to two-digit sequences reveals where the models fall short. Retaining all past glimpses as $|\mathcal{H}_v| = \infty$ produces the best results on Level 1 tasks for eight proprietary models. Under this setup on Level 2, however, accuracy drops to $\leq 5.0\%$ for GEMINI-3.1-PRO-PREVIEW and GPT-5.6-SOL. Although structured text and grid maps help GEMINI-3.7-FLASH reach 28.0%, no model surpasses 30.0% using these internal representations alone. Furthermore, keeping all past glimpses causes models to stop searching before gathering enough evidence. For instance, GEMINI-3.1-PRO-PREVIEW stops after only 4.52 steps on Level 1 with full visual history, compared to 7.18 steps under zero-lookback.

3.3 DIAGNOSTIC ANALYSES

3.3.1 INTERNAL PERCEPTUAL-STATE FAILURE MODES

To systematically analyze why agents fail as task complexity scales, we decompose recognition accuracy, error distributions, and the effect of visual coverage in Section C.

Positional and Exact-Sequence Accuracy. As illustrated in Figure 4a, scaling from Level 1 single digits to Level 2 two-digit sequences causes accuracy to drop across all state representations, even though models maintain 62.0% to 97.0% control accuracy under full observability. While Level 1 accuracy averages approximately 39.0%, individual digit positional accuracy drops to between 23.5% and 29.0%, and exact sequence accuracy drops to between 9.5% and 11.5%, showing that models struggle to order and integrate multiple digits.

Three Main Reasons Models Fail on Two-Digit Tasks. As shown in Figure 4b, when models fail on Level 1, they confuse one digit for another in 60.8% of cases. Moving to Level 2 introduces new failure modes. In 31.5% of trials, models identify only one digit correctly, and in 24.0% of trials, they fail to produce a two-digit prediction altogether. An audit of 1,600 Level 2 trajectories across eight models under Textual State and Metric Grid Map settings traces these mistakes to three common failure patterns. First, models stop searching too early. After discovering the first digit, they often mistake the blank space between the two numbers for the edge of the image and give up. Over 20% of two-digit trials end with just a single-digit prediction, with 238 runs never touching the second digit and stopping after a median of 15 out of 78 allowed steps. Second, models get lost and retrace their steps. In their written reasoning, models often state that they plan to explore new areas, yet their actual actions move back to locations they already saw. For example, GEMINI-3.7-FLASH wastes 45.1% of its moves revisiting old locations when relying on written notes, missing an entire digit in 24 out of 100 episodes. While coordinate maps reduce repeated visits to 13.6%, models still miss entire areas of the image. Third, models cling to early mistakes. Once a model writes down an incorrect guess for a digit, it almost never corrects itself. In 88.2% of examined cases where a model stored a wrong digit, it kept that wrong number in its final answer, even after seeing new glimpses that directly contradicted its initial guess.

Why Seeing More Does Not Always Lead to Correct Answers. As shown in Figure 3a and Figure 4c, uncovering the digits is required to guess them correctly, but seeing them is not enough. When models observe 25% or less of either digit, accuracy is just 1.0%. Accuracy increases to 11.3% when seeing 25% to 50% of the digits, and reaches 17.6% at 50% to 75% coverage. Even when models uncover more than 75% of both digits, accuracy is still only 26.0%. This shows that seeing the strokes is not the main problem, as models struggle most with putting the pieces together into a correct answer.

3.3.2 EXTERNAL VISUAL CANVASES IMPROVE PERFORMANCE

Giving models a visual canvas that automatically stitches their glimpses together improves their performance, as shown in Table 2. Placing new glimpses directly onto a visual canvas frees models from having to integrate past glimpses in context, leading to gains for Gemini models. For models like GPT-5.6-TERRA, seeing the canvas update causes them to stop exploring too early. On Level 1, GPT-5.6-TERRA achieves 31.0% accuracy after an average of 6.91 steps with a visual canvas, compared to 56.0% accuracy across 24.09 steps when evaluating the full stitched trajectory offline, with complete step counts in Section D. This disparity reveals that seeing a partial canvas can mislead certain models into thinking they have gathered enough evidence when they have not.

Performance improves most when separating search from recognition by stitching all visited glimpses onto an offline canvas. Using the paths from the textual state experiments, this offline canvas increases accuracy for CLAUDE-5-OPUS from 41.0% to 76.0% on Level 1 and from 13.0% to 67.0% on Level 2. In the same way, CLAUDE-5-FABLE increases from 38.0% to 81.0% on Level 1 and from 18.0% to 69.0% on Level 2. As detailed in Section G, both models had already uncovered over 80% of the target strokes during active search. The problem was therefore not insufficient exploration, but an inability to combine individual glimpses into an accurate final prediction.

3.3.3 DOWNSTREAM VISUAL INTERPRETATION AS AN INDEPENDENT BOTTLENECK

Even when evidence collection and representation succeed, downstream visual interpretation remains a limiting factor. To isolate this stage, we hold the exploration trajectory fixed and swap only the predictor, replaying identical search paths with a stronger model on the same reconstructed canvases. Fixing the trajectory to GPT-5.6-SOL (xhigh) and swapping in GEMINI-3.7-FLASH improves accuracy by 17.8 percentage points on Level 1 and 15.0 percentage points on Level 2 under

Table 2: Accuracy across perceptual state construction methods, per level and averaged. **Bold** indicates each model's best average accuracy. Entries with "–" were omitted due to budget constraints.

Model	Accuracy (%) ↑														
	Image Only ($\|\mathcal{H}_v\|=\infty$)			Image Only + Canvas ($\|\mathcal{H}_v\|=\infty$, offline)			Textual State ($\|\mathcal{H}_v\|=1$)			Textual State + Canvas ($\|\mathcal{H}_v\|=1$, offline)			Memory Canvas ($\|\mathcal{H}_v\|=1$, online)		
	L1	L2	Avg	L1	L2	Avg	L1	L2	Avg	L1	L2	Avg	L1	L2	Avg
GEMINI-3.1-PRO-PREVIEW	38.0	5.0	21.5	66.0	10.0	38.0	56.0	20.0	38.0	**87.0**	35.0	**61.0**	52.0	16.0	34.0
GEMINI-3.6-FLASH	53.0	18.0	35.5	77.0	35.0	56.0	47.0	15.0	31.0	**89.0**	49.0	**69.0**	61.0	34.0	47.5
GEMINI-3.7-FLASH	65.0	22.0	43.5	74.0	41.0	57.5	54.0	6.0	30.0	**81.0**	51.0	**66.0**	68.0	38.0	53.0
CLAUDE-5-OPUS	43.0	13.0	28.0	63.0	36.0	49.5	41.0	13.0	27.0	**76.0**	67.0	**71.5**	–	–	–
CLAUDE-5-FABLE	47.0	15.0	31.0	74.0	42.0	58.0	38.0	18.0	28.0	**81.0**	69.0	**75.0**	–	–	–
GPT-5.6-TERRA	35.0	1.0	18.0	49.0	1.0	25.0	25.0	0.0	12.5	**56.0**	4.0	**30.0**	31.0	1.0	16.0
GPT-5.6-SOL	37.0	1.0	19.0	49.0	1.0	25.0	29.0	0.0	14.5	**55.0**	6.0	**30.5**	33.0	1.0	17.0
GPT-5.6-SOL (xhigh)	56.0	29.0	42.5	**68.0**	59.0	**63.5**	37.0	3.0	20.0	65.0	26.0	45.5	–	–	–

full visual history. Because the trajectories and canvases are unchanged, this gain comes from the predictor, showing that downstream visual interpretation is an independent bottleneck. In the harness setting, this bottleneck is further evident from stroke coverage data. Incorrect episodes reach 98% or more stroke coverage, matching or exceeding coverage in correct episodes, yet models still misread the assembled canvas. Complete predictor-swap results across conditions appear in Section J.

3.3.4 AUTONOMOUS HARNESSES HELP BUT PERSISTENT MEMORY DOES NOT

Harness evaluation shows that code execution helps strong models overcome memory limits, while persistent memory increases search time without improving accuracy, as detailed in Section E.

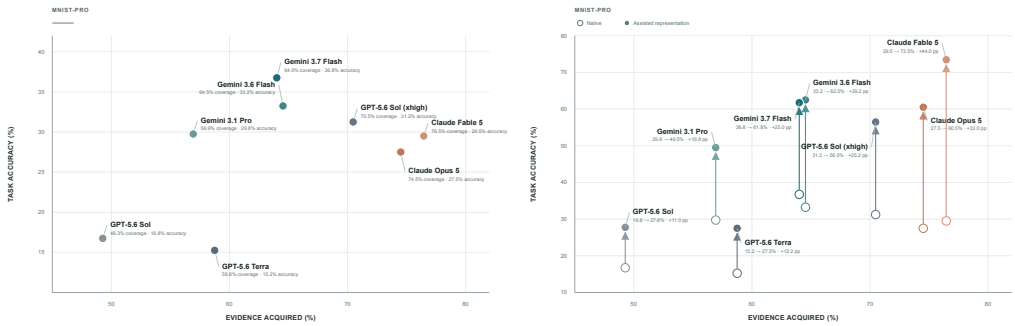
Tool Autonomy and Self-Directed Reconstruction. When equipped with coding tools, GEMINI-3.7-FLASH builds its own visual representations. Instead of relying on internal memory, the model writes and runs Python scripts such as `stitch.py` to piece together glimpses onto a visual canvas. This self-discovered strategy improves accuracy for GEMINI-3.7-FLASH to 88.0% on Level 1 and 63.0% on Level 2, outperforming both the in-context Image-Only baseline at 65.0% and 22.0%, and the online memory canvas at 68.0% and 38.0%. This gain, together with an increase in sensing steps from 5.26 to 15.67 on Level 1 and 14.69 to 41.00 on Level 2, shows that using a visual canvas prevents models from stopping too early. However, coding tools do not help weaker models like GPT-5.6-TERRA, which still struggle to track their position on the canvas.

Wasted Exploration from Persistent Memory. Allowing agents to keep notes and scripts across episodes fails to improve task performance. For example, GEMINI-3.7-FLASH averages 75.5% accuracy without memory compared to 73.5% with memory. Instead, keeping notes between runs causes models to search excessively. On Level 2, exploration increases from 41.0 steps to 52.4 steps with memory, and increases to 63.67 steps when models receive feedback on whether past answers were correct. Rather than improving search strategies, persistent memory causes models to get stuck in verification loops, re-checking strokes they have already seen without resolving their uncertainty.

Recognition Limits in Autonomous Harnesses. Harness errors do not come from insufficient search. Even on incorrect runs, models uncover over 98% of digit strokes. When facing ambiguous digit shapes, Gemini re-runs its stitching script an average of 3.75 times, compared to 2.10 times when correct. Because models gather nearly all visual evidence yet still misclassify the assembled canvas, the main bottleneck remains downstream visual interpretation.

3.3.5 MOVEMENT PATTERNS AND SPATIAL NAVIGATION

To understand how models explore, we track their Revisit Rate, measuring how often an agent returns to past coordinates, and their Stroke Coverage, measuring the percentage of the digit strokes uncovered, with complete results in Section G. Without past glimpses in memory, maintaining a coordinate grid map prevents models from wandering in circles. On Level 2, relying on textual memory alone causes GEMINI-3.7-FLASH to revisit old locations in 45.1% of its moves and miss an entire digit in 24 out of 100 episodes, but a grid map reduces these repeated visits to 13.6% while uncovering 61.7% of the strokes. Exploration strategies also differ across models. Open-source models struggle most with spatial navigation, as GLM-4.6V repeatedly cycles through old coordinates with revisit rates over 61% and uncovers only 33.1% of Level 2 strokes even with a



(a) Stroke coverage versus native task accuracy.

(b) Performance rescue via canvas reconstruction.

Figure 3: Left panel shows that high stroke coverage does not ensure high task accuracy under native in-context memory. Right panel shows that assembling visited glimpses onto an external visual canvas substantially rescues accuracy for models that explore thoroughly. Values reflect equal averages across Level 1 and Level 2 under Image Only and Textual State configurations.

grid map. Claude models prioritize broad search and uncover over 80% of the digit strokes, Gemini models prioritize speed by predicting in far fewer moves, and GPT-5.6-TERRA repeatedly retraces its steps, revisiting past locations in 46.1% of its moves. Representative trajectory plots illustrating these exploration patterns appear in Section K.

3.3.6 DIGIT SHAPES AND GEOMETRIC COMPLEXITY

The shape of each digit influences how difficult it is to explore and recognize. As detailed in Section F, accuracy declines as digits transition from simple straight strokes like 1, 4, and 7, to curves like 2, 3, and 5, and to closed loops like 0, 6, 8, and 9. For curved digits without past visual history, maintaining a coordinate grid map helps models trace curved strokes, increasing curve accuracy for GEMINI-3.7-FLASH from 36.7% to 56.7%. For digits with closed loops, coordinate tracking introduces small spatial errors that make it harder to detect where a loop closes. As a result, loop accuracy drops from 57.5% to 45.0% for GEMINI-3.7-FLASH, from 40.0% to 32.5% for CLAUDE-5-OPUS, and from 22.5% to 12.5% for CLAUDE-5-SONNET when switching to a grid map.

3.4 STRESS TESTS

To test whether our findings hold under different sensing constraints, we evaluate two dimensions, visual glimpse history $|\mathcal{H}_v|$ and glimpse window size. First, expanding visual glimpse history helps models with their search trajectories. Extending glimpse history $|\mathcal{H}_v|$ from 1 to ∞ reduces Level 2 exploration steps by more than half for GEMINI-3.7-FLASH, dropping from 34.84 steps to 15.90 steps, while improving Level 2 Textual State accuracy from 6.0% to 23.0%. Second, a larger glimpse window allows models to explore the canvas in fewer steps. Widening the window from 32×32 to 128×128 pixels increases Level 2 Image-Only accuracy for GEMINI-3.7-FLASH from 3.0% to 74.0%, while reducing sensing steps from 33.35 to 6.02. Complete model comparisons and detailed results appear in Sections H and I.

4 DISCUSSION ON BOTTLENECKS IN AGENTIC PERCEPTION

Our experiments reveal a consistent pattern of failure across the agentic perception loop. Failure concentrates at three stages, namely evidence collection, evidence representation, and downstream visual interpretation. These bottlenecks do not fail in isolation. Fixing one stage only uncovers how much the others were already limiting performance. Canvas reconstruction resolves the representation bottleneck, but only when evidence collection succeeded. For CLAUDE-5-OPUS and CLAUDE-5-FABLE, which uncovered over 80% of digit strokes during search, stitching those glimpses onto an offline canvas recovers over 50 percentage points on Level 2, as shown in Figure 3b. For models where collection failed, the canvas offers no benefit. On Level 2, GPT-5.6-TERRA reaches 1.0% accuracy with and without a canvas, because it never located the second digit. The predictor-swap

results show the same dependency at the downstream visual interpretation stage. Swapping in a stronger GEMINI-3.7-FLASH predictor on GPT-5.6-SOL (xhigh) trajectories improves accuracy by 18.4 percentage points in the 38% of episodes where both digits were observed, but achieves 0.0% in the remaining 62% where collection failed, as detailed in Section J. Persistent cross-episode memory also fails to help, and for a different reason. The bottlenecks do not come from a lack of information across episodes. They come from how models process each glimpse within a single episode. GEMINI-3.7-FLASH scores 75.5% without persistent notes and 73.5% with them, while Level 2 exploration grows from 41.0 to 52.4 steps and further to 63.7 steps with correctness feedback. Retaining notes does not change how the model integrates each new glimpse into its belief state. Instead, persistent logs leave the model aware of past uncertainty without resolving it, leading to verification loops that consume sensing budget without improving recognition, as detailed in Section E. These findings point to a broader principle for agentic perception. Gains from fixing one bottleneck are bounded by the state of the others. In richer visual domains, each bottleneck will likely be harder to address on its own. Noisier observations weaken evidence representation, longer planning horizons increase collection failures, and more complex visual structures make downstream interpretation harder. Advancing agentic perception will require addressing all three stages together.

5 RELATED WORKS

Active Perception and Visual Search. Standard vision-language benchmarks, such as MMBench (Liu et al., 2024), MMMU (Yue et al., 2024), MM-Vet (Yu et al., 2024), SEED-Bench (Li et al., 2023), BLINK (Fu et al., 2024), and MMT-Bench (Ying et al., 2024), evaluate static reasoning on fully observable images without requiring models to explore or gather evidence step by step. In contrast, active perception studies how agents move their sensors to resolve visual uncertainty (Bajcsy et al., 2016). Recent multimodal benchmarks, such as ActiView (Wang et al., 2025) and ActiveVision (Zhang et al., 2026), restrict the visible field of view to evaluate gaze shifting, while V^* (Wu & Xie, 2023) uses visual memory for high-resolution zooming. However, these works allow models to keep all past glimpses in context, bypassing the challenge of maintaining an evolving state representation. `MNIST-PRO` bridges this gap for agentic perception by enforcing controlled visual lookback constraints down to zero lookback where $|\mathcal{H}_v| = 1$, requiring agents to coordinate active visual search with state tracking under strict sensing budgets.

Working Memory and State Representation under Partial Observability. Operating in partially observable environments requires agents to maintain an internal state representation over time. Existing multimodal benchmarks evaluate memory through embodied navigation (Shridhar et al., 2020), spatial memory (Xu et al., 2026; Yang et al., 2025), or interactive GUI environments (Xie et al., 2024). However, task success in these domains is often complicated by physics simulation, complex visual rendering, and tool-use mechanics, making it difficult to isolate how models track visual state. `MNIST-PRO` removes these confounding factors through a controlled partially observable benchmark. This design allows us to systematically evaluate state representations across in-context memory, visual canvases, and autonomous agent harnesses, separating active exploration from visual state interpretation.

6 CONCLUSION

In this work, we introduce `MNIST-PRO`, a benchmark evaluating how multimodal AI agents explore and represent partially observed visual environments. By formulating digit recognition as a sequential visual search under controlled lookback and explicit state representations, we isolate agentic perception from embodied control and decouple exploration from state construction. Our experiments reveal that the performance drop under partial observability comes from three bottlenecks. Agents terminate search prematurely before exhausting their exploration budget, struggle to combine disjoint glimpses in working memory, and make downstream classification errors even when canvas coverage exceeds 98%. Furthermore, equipping agents with persistent memory across episodes in harnesses fails to improve performance, instead inducing circular re-exploration. These findings demonstrate that advancing agentic perception requires coordinating active search, structured state representations, and downstream visual interpretation, rather than simply expanding raw context windows.

AI USE STATEMENT

In this work, we used generative AI to polish writing, correct grammar, improve clarity, and debug code. AI was not used for research ideation or experimental design. The authors reviewed all edits and code changes, and they take full responsibility for the contents of this paper.

REPRODUCIBILITY STATEMENT

To ensure reproducibility, the codebase, including the environment, evaluation harnesses, prompt templates, and reproduction scripts, is provided in the supplementary material. The POMDP formulation and environment mechanics are detailed in Section 2.1 and Section A.3. Data preprocessing pipelines, canvas dimensions, and environment parameters are described in Section A. Prompt templates and JSON schemas for agents are provided in Section A.4. Experimental protocols, baseline configurations, and harness setups are documented in Section 2.4 and Sections B and E.

REFERENCES

Ruzena Bajcsy, Yiannis Aloimonos, and John K. Tsotsos. Revisiting active perception, 2016. URL https://arxiv.org/abs/1603.02729.

Anthony Brohan, Noah Brown, Justice Carbajal, Yevgen Chebotar, Xi Chen, Krzysztof Choromanski, Tianli Ding, Danny Driess, Avinava Dubey, Chelsea Finn, Pete Florence, Chuyuan Fu, Montse Gonzalez Arenas, Keerthana Gopalakrishnan, Kehang Han, Karol Hausman, Alexander Herzog, Jasmine Hsu, Brian Ichter, Alex Irpan, Nikhil Joshi, Ryan Julian, Dmitry Kalashnikov, Yuheng Kuang, Isabel Leal, Lisa Lee, Tsang-Wei Edward Lee, Sergey Levine, Yao Lu, Henryk Michalewski, Igor Mordatch, Karl Pertsch, Kanishka Rao, Krista Reymann, Michael Ryoo, Grecia Salazar, Pannag Sanketi, Pierre Sermanet, Jaspiar Singh, Anikait Singh, Radu Soricut, Huong Tran, Vincent Vanhoucke, Quan Vuong, Ayzaan Wahid, Stefan Welker, Paul Wohlhart, Jialin Wu, Fei Xia, Ted Xiao, Peng Xu, Sichun Xu, Tianhe Yu, and Brianna Zitkovich. Rt-2: Vision-language-action models transfer web knowledge to robotic control, 2023. URL https://arxiv.org/abs/2307.15818.

ARC Prize Foundation. Arc-agi-3: A new challenge for frontier agentic intelligence, 2026. URL https://arxiv.org/abs/2603.24621.

Xingyu Fu, Yushi Hu, Bangzheng Li, Yu Feng, Haoyu Wang, Xudong Lin, Dan Roth, Noah A. Smith, Wei-Chiu Ma, and Ranjay Krishna. Blink: Multimodal large language models can see but not perceive, 2024. URL https://arxiv.org/abs/2404.12390.

Y. Lecun, L. Bottou, Y. Bengio, and P. Haffner. Gradient-based learning applied to document recognition. *Proceedings of the IEEE*, 86(11):2278–2324, 1998. doi: 10.1109/5.726791.

Bohao Li, Rui Wang, Guangzhi Wang, Yuying Ge, Yixiao Ge, and Ying Shan. Seed-bench: Benchmarking multimodal llms with generative comprehension, 2023. URL https://arxiv.org/abs/2307.16125.

Yuan Liu, Haodong Duan, Yuanhan Zhang, Bo Li, Songyang Zhang, Wangbo Zhao, Yike Yuan, Jiaqi Wang, Conghui He, Ziwei Liu, Kai Chen, and Dahua Lin. Mmbench: Is your multi-modal model an all-around player?, 2024. URL https://arxiv.org/abs/2307.06281.

Qwen Team. Qwen3.8-Max: A new bar for coding and cowork, August 2026. URL https://qwen.ai/blog?id=qwen3.8.

Manolis Savva, Abhishek Kadian, Oleksandr Maksymets, Yili Zhao, Erik Wijmans, Bhavana Jain, Julian Straub, Jia Liu, Vladlen Koltun, Jitendra Malik, Devi Parikh, and Dhruv Batra. Habitat: A platform for embodied ai research, 2019. URL https://arxiv.org/abs/1904.01201.

Mohit Shridhar, Jesse Thomason, Daniel Gordon, Yonatan Bisk, Winson Han, Roozbeh Mottaghi, Luke Zettlemoyer, and Dieter Fox. Alfred: A benchmark for interpreting grounded instructions for everyday tasks, 2020. URL https://arxiv.org/abs/1912.01734.

V Team, Wenyi Hong, Wenmeng Yu, Xiaotao Gu, Guo Wang, Guobing Gan, Haomiao Tang, Jiale Cheng, Ji Qi, Junhui Ji, Lihang Pan, Shuaiqi Duan, Weihan Wang, Yan Wang, Yean Cheng, Zehai He, Zhe Su, Zhen Yang, Ziyang Pan, Aohan Zeng, Baoxu Wang, Bin Chen, Boyan Shi, Changyu Pang, Chenhui Zhang, Da Yin, Fan Yang, Guoqing Chen, Jiazheng Xu, Jiale Zhu, Jiali Chen, Jing Chen, Jinhao Chen, Jinghao Lin, Jinjiang Wang, Junjie Chen, Leqi Lei, Letian Gong, Leyi Pan, Mingdao Liu, Mingde Xu, Mingzhi Zhang, Qinkai Zheng, Sheng Yang, Shi Zhong, Shiyu Huang, Shuyuan Zhao, Siyan Xue, Shangqin Tu, Shengbiao Meng, Tianshu Zhang, Tianwei Luo, Tianxiang Hao, Tianyu Tong, Wenkai Li, Wei Jia, Xiao Liu, Xiaohan Zhang, Xin Lyu, Xinyue Fan, Xuancheng Huang, Yanling Wang, Yadong Xue, Yanfeng Wang, Yanzi Wang, Yifan An, Yifan Du, Yiming Shi, Yiheng Huang, Yilin Niu, Yuan Wang, Yuanchang Yue, Yuchen Li, Yutao Zhang, Yuting Wang, Yu Wang, Yuxuan Zhang, Zhao Xue, Zhenyu Hou, Zhengxiao Du, Zihan Wang, Peng Zhang, Debing Liu, Bin Xu, Juanzi Li, Minlie Huang, Yuxiao Dong, and Jie Tang. Glm-4.5v and glm-4.1v-thinking: Towards versatile multimodal reasoning with scalable reinforcement learning, 2025. URL https://arxiv.org/abs/2507.01006.

Ziyue Wang, Chi Chen, Fuwen Luo, Yurui Dong, Yuanchi Zhang, Yuzhuang Xu, Xiaolong Wang, Peng Li, and Yang Liu. Actiview: Evaluating active perception ability for multimodal large language models, 2025. URL https://arxiv.org/abs/2410.04659.

Penghao Wu and Saining Xie. V*: Guided visual search as a core mechanism in multimodal llms, 2023. URL https://arxiv.org/abs/2312.14135.

Tianbao Xie, Danyang Zhang, Jixuan Chen, Xiaochuan Li, Siheng Zhao, Ruisheng Cao, Toh Jing Hua, Zhoujun Cheng, Dongchan Shin, Fangyu Lei, Yitao Liu, Yiheng Xu, Shuyan Zhou, Silvio Savarese, Caiming Xiong, Victor Zhong, and Tao Yu. Osworld: Benchmarking multimodal agents for open-ended tasks in real computer environments, 2024. URL https://arxiv.org/abs/2404.07972.

Peiran Xu, Sudong Wang, Yao Zhu, Jianing Li, Gege Qi, and Yunjian Zhang. Spatialbench: Benchmarking multimodal large language models for spatial cognition, 2026. URL https://arxiv.org/abs/2511.21471.

Jihan Yang, Shusheng Yang, Anjali W. Gupta, Rilyn Han, Li Fei-Fei, and Saining Xie. Thinking in space: How multimodal large language models see, remember, and recall spaces, 2025. URL https://arxiv.org/abs/2412.14171.

Kaining Ying, Fanqing Meng, Jin Wang, Zhiqian Li, Han Lin, Yue Yang, Hao Zhang, Wenbo Zhang, Yuqi Lin, Shuo Liu, Jiayi Lei, Quanfeng Lu, Runjian Chen, Peng Xu, Renrui Zhang, Haozhe Zhang, Peng Gao, Yali Wang, Yu Qiao, Ping Luo, Kaipeng Zhang, and Wenqi Shao. Mmt-bench: A comprehensive multimodal benchmark for evaluating large vision-language models towards multitask agi, 2024. URL https://arxiv.org/abs/2404.16006.

Weihao Yu, Zhengyuan Yang, Linjie Li, Jianfeng Wang, Kevin Lin, Zicheng Liu, Xinchao Wang, and Lijuan Wang. Mm-vet: Evaluating large multimodal models for integrated capabilities, 2024. URL https://arxiv.org/abs/2308.02490.

Xiang Yue, Yuansheng Ni, Kai Zhang, Tianyu Zheng, Ruoqi Liu, Ge Zhang, Samuel Stevens, Dongfu Jiang, Weiming Ren, Yuxuan Sun, Cong Wei, Botao Yu, Ruibin Yuan, Renliang Sun, Ming Yin, Boyuan Zheng, Zhenzhu Yang, Yibo Liu, Wenhao Huang, Huan Sun, Yu Su, and Wenhu Chen. Mmmu: A massive multi-discipline multimodal understanding and reasoning benchmark for expert agi, 2024. URL https://arxiv.org/abs/2311.16502.

Jiarui Zhang, Muzi Tao, Shangshang Wang, Ollie Liu, Xuezhe Ma, and Willie Neiswanger. An exam for active observers, 2026. URL https://arxiv.org/abs/2607.16165.

Shuyan Zhou, Frank F. Xu, Hao Zhu, Xuhui Zhou, Robert Lo, Abishek Sridhar, Xianyi Cheng, Tianyue Ou, Yonatan Bisk, Daniel Fried, Uri Alon, and Graham Neubig. Webarena: A realistic web environment for building autonomous agents, 2024. URL https://arxiv.org/abs/2307.13854.

A Dataset Preprocessing and Environment Hyperparameters

A.1 Dataset Preprocessing

We preprocess MNIST images to standardize canvas dimensions and contrast. For the Level 1 task, each 28×28 image is inverted to yield a black digit on a white background, resized to 224×224 pixels using bilinear interpolation, and thresholded at an intensity of 200 to remove interpolation blur and binarize the canvas. For the Level 2 task, two raw 28×28 images are concatenated horizontally into a 28×56 composite, inverted, bilinearly resized to 224×448 pixels, and binarized using the same intensity threshold of 200.

A.2 Episode and Environment Settings

We evaluate configurations across 100 test episodes per task. For Level 1, we allocate 10 episodes to each digit class from 0 to 9. For Level 2, each episode presents a digit pair sampled from $\{0, \ldots, 9\}$.

Stroke-Anchored Initialization. To focus evaluation on stroke tracing rather than background search, the initial window position p_0 anchors on digit strokes. Let $\mathcal{K}$ denote the set of black stroke pixels in the binarized canvas I. In Level 1, $\mathcal{K}$ contains strokes from the single digit. In Level 2, $\mathcal{K}$ contains strokes from both digits across the full canvas without restriction to the left digit. Episodes can therefore start on either digit. To ensure deterministic evaluation across models, the environment derives a seed from the MD5 checksum of the image bytes.

$$\text{seed} = \text{MD5}(I) \pmod{2^{32}} \tag{1}$$

An anchor pixel with coordinates c_x and c_y in $\mathcal{K}$ is sampled using this seed to set the center of the initial glimpse window.

$$x_0 = \text{clip}_x\left(c_x - \lfloor w/2 \rfloor\right), \quad y_0 = \text{clip}_y\left(c_y - \lfloor h/2 \rfloor\right) \tag{2}$$

This initialization ensures that at $t = 0$, agents observe digit strokes rather than empty background, spending their sensing budget on stroke tracing rather than locating the digit region.

Observation Rendering and Visual Framing. At step t, the observation $o_t \in \mathbb{R}^{H \times W' \times 3}$ is a composite image rather than a cropped patch. Unobserved canvas regions are gray with RGB values $[128, 128, 128]$. The canvas region within the glimpse window contains the black digit strokes on a white background. A 2-pixel cyan border with hexadecimal value `#00E5FF` and RGB values $[0, 229, 255]$ marks the perimeter of the glimpse window. Rendering the full canvas with this border provides spatial context, allowing the model to locate the window relative to canvas boundaries without textual coordinates.

A.3 Formal POMDP Mechanics and Budget Formulation

This section details the coordinate spaces, clipping rules, and step budgets that govern the environment.

Glimpse Window and State Boundaries. For canvas dimensions $H \times W'$ and window dimensions $h \times w$, the top-left coordinate p_t is bounded by the coordinate space $\mathcal{P}$.

$$p_t \in \mathcal{P} = \{0, \ldots, W' - w\} \times \{0, \ldots, H - h\} \tag{3}$$

The state at step t comprises image I and position p_t within state space $\mathcal{S}$.

Movement Transitions and Silent Clipping. Let $u(a_t)$ denote the unit displacement vector for movement action $a_t \in \mathcal{A}_{\text{move}}$.

$$u(\text{up}) = (0, -1), \qquad u(\text{down}) = (0, 1),$$
$$u(\text{left}) = (-1, 0), \qquad u(\text{right}) = (1, 0)$$

The transition updates window coordinates using step size δ.

$$p_{t+1} = \text{clip}(p_t + \delta \cdot u(a_t)) \tag{4}$$

Here clipping clamps coordinates to canvas boundaries.

$$\text{clip}_x(x) = \max(0, \min(W' - w, x)),$$
$$\text{clip}_y(y) = \max(0, \min(H - h, y))$$

Transitions that collide with canvas boundaries are silently clipped. The environment emits no warnings or error messages. If an agent at $x_t = 0$ selects $a_t =$ left, position p_{t+1} remains p_t, and observation o_{t+1} is identical to o_t. This setup tests whether models detect boundaries from visual changes rather than textual alerts. When agents track position through text integration using state update $\mathbf{g}_t = \mathbf{g}_{t-1} + u(a_{t-1})$, boundary clipping causes coordinate drift that decouples the internal map from the actual state.

Sensing Budget and Episode Termination. The environment sets a step budget $T_{\max}$ derived from grid dimensions.

$$T_{\max} = \text{steps}_{\text{width}} \times \text{steps}_{\text{height}} \tag{5}$$

Horizontal and vertical step counts depend on canvas and window dimensions.

$$\text{steps}_{\text{width}} = \left\lceil \frac{\max(0, W' - w)}{\delta} \right\rceil + 1, \quad \text{steps}_{\text{height}} = \left\lceil \frac{\max(0, H - h)}{\delta} \right\rceil + 1 \tag{6}$$

For Level 1 with $W' = H = 224$, $w = h = 64$, and $\delta = 32$, the limits are $\text{steps}_{\text{width}} = \text{steps}_{\text{height}} = 6$, giving $T_{\max} = 36$ steps. For Level 2 with $W' = 448$ and $H = 224$, the counts are $\text{steps}_{\text{width}} = 13$ and $\text{steps}_{\text{height}} = 6$, giving $T_{\max} = 78$ steps.

When an episode reaches $t = T_{\max}$ steps without a prediction, the prompt appends a warning instructing the model to predict. The warning uses `<digit>` for Level 1 and `<sequence>` for Level 2.

```
"You have run out of steps. You must provide your final answer now. Remember, you only
    have one chance! Output JSON like {\"action\": \"answer\", \"value\": <sequence>} at
    the end."
```

If the model fails to return a valid prediction across three attempts, the environment assigns a fallback guess of -1 for Level 1 or `"-1"` for Level 2, which counts as incorrect.

A.4 Prompt Templates and Baseline JSON Schemas

This section documents system instructions, user prompt templates, and JSON response schemas for the in-loop baselines.

Level 1 System Instruction. The Level 1 system prompt defines the digit recognition task, black strokes on white canvas with dark gray masking, movement actions, and termination rules.

```
You are an active vision agent playing a game to identify an MNIST digit. Your goal is to
    figure out what the digit (0-9) is. The digit is drawn in black on a white
    background, and unseen areas are masked in dark gray. You can move the visible box
    around. When you are confident about the digit, you must provide your final answer.
    Note: You only have one chance to provide the final answer, so make sure you are
    confident before doing so! For moving, output: {"action": "move", "direction": "up"}
    (directions: 'up', 'down', 'left', 'right'). For answering, output: {"action":
    "answer", "value": <digit>}.
```

Level 2 Two-Digit System Instruction. The Level 2 system prompt specifies concatenated digits with left to right reading order.

```
You are an active vision agent playing a game to identify a sequence of horizontally
    concatenated MNIST digits. Your goal is to figure out the exact sequence of digits
    (e.g. 58) from left to right. The digits are drawn in black on a white background,
    and unseen areas are masked in dark gray. You can move the visible box around. When
    you are confident about the entire sequence of digits, you must provide your final
    answer. Note: You only have one chance to provide the final answer, so make sure you
    are confident before doing so! For moving, output: {"action": "move", "direction":
    "up"} (directions: 'up', 'down', 'left', 'right'). For answering, output the sequence
    as a raw string of digits, like: {"action": "answer", "value": "58"}.
```

Per-Turn User Prompt Templates. The Image-Only baseline with infinite history receives the base prompt at each step.

```
Based on the observation, output your next move or final answer in the requested JSON
    format.
```

The Textual State baseline prompts the model to return reasoning.

```
Based on the observation, output your next move or final answer in the requested JSON
    format. Include an extra key 'thought' in the JSON containing your reasoning.
```

Prior history in text follows a turn by turn format.

```
Previous actions and thoughts:
Turn 1:
{"thought": "I see a curved stroke. Moving down.", "action": "move", "direction": "down"}
```

Full-Image Control Prompts. Single turn control baselines receive the uncropped image and a direct question.

```
// Level 1 prompt
Look at this image of an MNIST digit. What digit (0-9) is shown?

// Level 2 prompt
Look at this image of concatenated MNIST digits. What sequence of digits is shown?
```

Action and Prediction JSON Formats. Agent responses follow action and answer schemas.

```
// Move action
{
  "action": "move",
  "direction": "right"
}

// Move with thought
{
  "thought": "I see a curved stroke. Moving down.",
  "action": "move",
  "direction": "down"
}

// Level 1 answer
{
  "action": "answer",
  "value": 3
}

// Level 2 answer
{
  "action": "answer",
  "value": "58"
}
```

For the Metric Grid Map baseline, the coordinate tracking instructions and structured spatial map schema are given in Section A.5.

A.5 Metric Grid Map Details

This section describes the prompt specification, JSON schema, and displacement vectors for the Metric Grid Map baseline with history length $|\mathcal{H}_v| = 1$.

Prompt Specification. The Metric Grid Map prompt sets the Cartesian grid and relative movement vectors.

```
Based on the observation, output your next move or final answer in the requested JSON
    format. Assume your starting position is coordinate [0, 0]. Moving up, down, left, or
    right changes your relative coordinates by [0, -1], [0, 1], [-1, 0], or [1, 0]
    respectively. Include two extra keys in your JSON response:
1. 'thought': your reasoning.
2. 'spatial_map': a list of exactly one object representing your current step as spatial
    memory, with 'coords': [x, y] and 'features': 'description of what was observed at
    those coordinates' (e.g. 'empty background', 'top curve of a digit', 'vertical
    stroke', etc.).
```

Relative Displacement Vectors. At step t, the agent tracks position on a Cartesian grid. Given previous position $\mathbf{g}_{t-1}$ and movement action $a_{t-1} \in \mathcal{A}_{\text{move}}$, the new position $\mathbf{g}_t$ updates by vector

addition.

$$\mathbf{g}_t = \mathbf{g}_{t-1} + u(a_{t-1}) \tag{7}$$

Unit displacement vectors $u(a)$ are defined for each direction.

$$u(\texttt{up}) = [0, -1], \qquad u(\texttt{down}) = [0, 1],$$
$$u(\texttt{left}) = [-1, 0], \qquad u(\texttt{right}) = [1, 0]$$

Constrained JSON Response Format. To track spatial state, the model outputs thoughts, actions, and map coordinates in JSON format.

```
{
  "thought": "I see empty space at [0,0]. Moving right.",
  "spatial_map": [{"coords": [0, 0], "features": "empty background"}],
  "action": "move", "direction": "right"
}
```

The state representation $\mathcal{M}_t$ records explored coordinates and feature descriptions accumulated up to step t.

$$\mathcal{M}_t = \{(\mathbf{g}_\tau, \mathbf{f}_\tau)\}_{\tau=0}^{t} \tag{8}$$

Here coordinate $\mathbf{g}_\tau \in \mathbb{Z}^2$ pairs with feature description $\mathbf{f}_\tau$ recorded at step τ.

A.6 Online Visual Memory Canvas Transition Formulation

This section details the canvas update rule for the Online Visual Memory Canvas baseline with history length $|\mathcal{H}_v| = 1$.

Programmatic Updates on Merged Canvas. The canvas I has dimensions $H \times W'$, and the glimpse window has dimensions $h \times w$. At $t = 0$, canvas C_0 is initialized with gray value 128 across all pixels.

When the agent receives glimpse o_t at position p_t, canvas C_t updates by a piecewise rule.

$$C_t(u, v) = \begin{cases} I(u, v) & \text{if } (u, v) \in [x_t, x_t + w) \times [y_t, y_t + h) \\ C_{t-1}(u, v) & \text{otherwise} \end{cases} \tag{9}$$

This update overlays observed pixels onto the canvas, creating an aligned record of explored regions. Because this projection extracts pixel values directly from ground-truth image I, the canvas contains no border lines.

B Baseline Performance under Native Multi-Turn Setup

This section reports the native multi-turn baseline where models use conversational history to choose moves and answers without tools or external memory. Table 3 shows performance across proprietary and open-source models. On unmasked control canvases, proprietary models reach 92.0% to 98.0% accuracy and open-source models reach 88.0% to 91.5% accuracy. This shows that models recognize full digits when they are visible. Under active sensing, accuracy is lower across models. On Level 2, models such as Claude-5-Sonnet, GPT-5.6-Terra, GPT-5.6-Sol, Qwen-3.8-27B, and GLM-4.6V score 0.0%. In comparison, Gemini-3.7-Flash achieves 75.0% on Level 1 and 47.0% on Level 2. Exploration steps also vary across models. Claude series models take more steps, with Claude-5-Sonnet averaging 43.24 steps on Level 2. Open-source models stop earlier, taking 2.76 to 4.62 steps on average.

C Level 2 Two-Digit Failure Mode Decomposition

This section describes errors on the Level 2 two-digit task discussed in Section 3.3.1. Figure 4 shows the error distributions. In Figure 4a, models often predict one digit correctly, but exact sequence accuracy is lower because models need to identify both digits. In Figure 4b, error types change across tasks. Level 1 errors are mostly wrong digit guesses. Level 2 adds partial answers and answers with the wrong number of digits when models stop before reaching the second digit. In Figure 4c, higher stroke coverage does not always lead to correct answers. Even when models see more than 75% of both digits, sequence accuracy stays at 26.0%, indicating that combining glimpses into a full number remains challenging.

Table 3: Performance on MNIST-PRO with native multi-turn conversation history.

Model	Control Accuracy (%) ↑			Multi-turn Accuracy (%) ↑			Average Steps		
	L1	L2	Avg.	L1	L2	Avg.	L1	L2	Avg.
Proprietary Models									
GEMINI-3.1-PRO-PREVIEW	99.0	97.0	98.0	53.0	18.0	35.5	7.33	23.53	15.43
GEMINI-3.6-FLASH	96.0	96.0	96.0	64.0	28.0	46.0	9.48	38.69	24.09
GEMINI-3.7-FLASH	98.0	97.0	97.5	75.0	47.0	61.0	8.23	28.45	18.34
CLAUDE-5-SONNET	94.0	90.0	92.0	23.0	0.0	11.5	15.96	43.24	29.60
CLAUDE-5-OPUS	93.0	91.0	92.0	49.0	17.0	33.0	12.26	36.86	24.56
CLAUDE-5-FABLE	95.0	89.0	92.0	50.0	16.0	33.0	10.45	25.01	17.73
GPT-5.6-TERRA	83.0	62.0	72.5	30.0	0.0	15.0	10.29	20.79	15.54
GPT-5.6-SOL	88.0	82.0	85.0	37.0	0.0	18.5	5.74	13.99	9.87
Open-Source Models									
QWEN-3.8-27B	89.0	87.0	88.0	29.0	0.0	14.5	4.62	9.23	6.93
GLM-4.6V	91.0	92.0	91.5	13.0	0.0	6.5	2.76	5.25	4.01

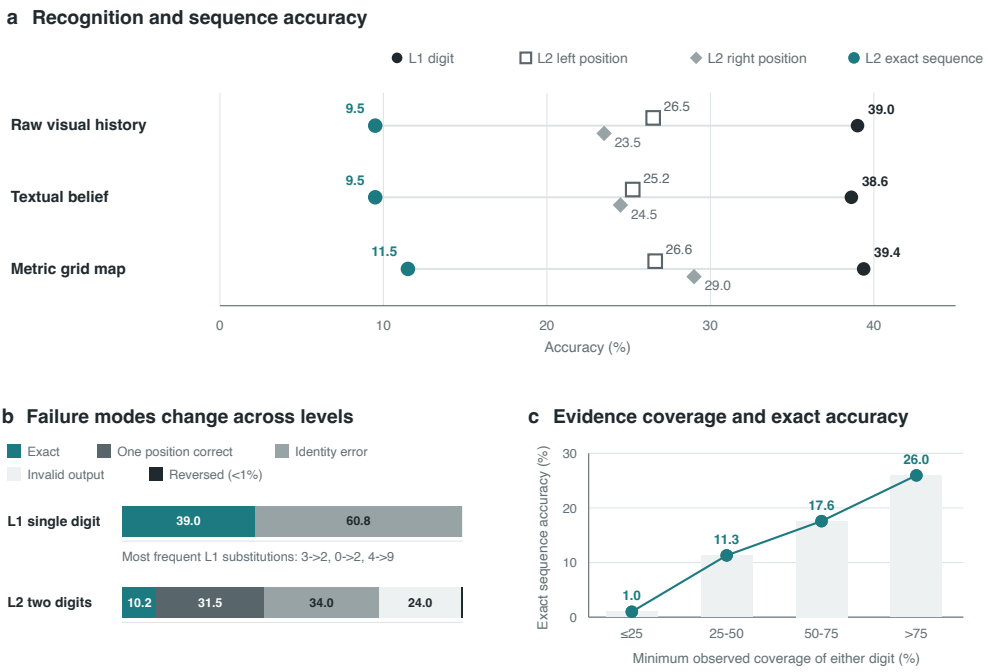


Figure 4: **Level 2 introduces search and sequence-construction failures beyond single-digit recognition.** (a) Level 1 digit accuracy compared with Level 2 positional and exact-sequence accuracy. (b) Level 1 errors are dominated by valid digit substitutions, whereas Level 2 adds partial sequences and invalid digit counts, while reversing the two digits is rare. Identity error denotes an incorrect digit in Level 1 and both positions incorrect in Level 2. (c) Exact-sequence accuracy increases with the minimum observed coverage of either digit. Values pool the eight models with complete logs across the three internal state conditions (Image Only, Textual State, Metric Grid Map), with 100 matched evaluation episodes per model-condition cell. Coverage is computed from the union of all observed glimpse windows, where all statistics are descriptive.

D AVERAGE EXPLORATION STEPS ACROSS STATE CONSTRUCTION METHODS

This section reports exploration step counts for the state construction experiments in Section 3.3.2. Table 4 lists the average steps across tested settings. Offline canvas runs replay moves from the live runs, so their step counts match. With textual state and history length $|\mathcal{H}_v| = 1$, models take more steps on Level 2. For example, GPT-5.6-TERRA averages 69.67 steps, CLAUDE-5-OPUS takes

Table 4: Average steps across state construction methods, per level and averaged. Offline conditions replay the trajectory of the live condition to their left, so their step counts are identical by construction. Entries marked with "–" were omitted due to budget constraints.

Model	Image Only ($\|\mathcal{H}_v\|=\infty$)			Image Only + Canvas ($\|\mathcal{H}_v\|=\infty$, offline)			Textual State ($\|\mathcal{H}_v\|=1$)			Textual State + Canvas ($\|\mathcal{H}_v\|=1$, offline)			Memory Canvas ($\|\mathcal{H}_v\|=1$, online)		
	L1	L2	**Avg**	L1	L2	**Avg**	L1	L2	**Avg**	L1	L2	**Avg**	L1	L2	**Avg**
GEMINI-3.1-PRO-PREVIEW	4.52	11.95	8.23	4.52	11.95	8.23	7.18	18.28	12.73	7.18	18.28	12.73	4.12	16.78	10.45
GEMINI-3.6-FLASH	5.62	15.11	10.37	5.62	15.11	10.37	9.35	39.29	24.32	9.35	39.29	24.32	5.03	17.57	11.30
GEMINI-3.7-FLASH	5.26	14.69	9.97	5.26	14.69	9.97	7.92	34.84	21.38	7.92	34.84	21.38	5.56	20.40	12.98
CLAUDE-5-OPUS	7.43	22.50	14.97	7.43	22.50	14.97	11.61	39.90	25.76	11.61	39.90	25.76	–	–	–
CLAUDE-5-FABLE	7.27	22.30	14.79	7.27	22.30	14.79	11.30	35.93	23.62	11.30	35.93	23.62	–	–	–
GPT-5.6-TERRA	18.19	39.67	28.93	18.19	39.67	28.93	24.09	69.67	46.88	24.09	69.67	46.88	6.91	44.56	25.73
GPT-5.6-SOL	6.53	16.98	11.76	6.53	16.98	11.76	7.01	22.52	14.77	7.01	22.52	14.77	4.09	14.52	9.30
GPT-5.6-SOL (xhigh)	9.59	35.43	22.51	9.59	35.43	22.51	9.19	28.00	18.59	9.19	28.00	18.59	–	–	–

Table 5: **Autonomous agent harness performance on `MNIST-PRO`.** Accuracy and exploration steps for GPT-5.6-TERRA and GEMINI-3.7-FLASH across non-harness baselines, offline canvas reconstruction, agentic shell, and native MCP harness configurations (with persistent memory and correctness feedback). Entries marked with "–" were omitted due to budget constraints.

Condition	GPT-5.6-TERRA						GEMINI-3.7-FLASH					
	Acc (%) ↑			**Avg. Steps ↓**			**Acc (%) ↑**			**Avg. Steps ↓**		
	L1	L2	**Avg**	L1	L2	**Avg**	L1	L2	**Avg**	L1	L2	**Avg**
Non-harness												
Image Only ($\|\mathcal{H}_v\|=\infty$)	35.0	1.0	18.0	18.19	39.67	28.93	65.0	22.0	43.5	5.26	14.69	9.97
Textual State ($\|\mathcal{H}_v\|=1$)	25.0	0.0	12.5	24.09	69.67	46.88	54.0	6.0	30.0	7.92	34.84	21.38
Metric Grid Map ($\|\mathcal{H}_v\|=1$)	24.0	0.0	12.0	21.92	62.43	42.17	57.0	28.0	42.5	6.82	20.37	13.60
Memory Canvas ($\|\mathcal{H}_v\|=1$, online)	31.0	1.0	16.0	6.91	44.56	25.73	68.0	38.0	53.0	5.56	20.40	12.98
Non-harness, offline canvas												
Image Only + Canvas	49.0	1.0	25.0	18.19	39.67	28.93	74.0	41.0	57.5	5.26	14.69	9.97
Textual State + Canvas	**56.0**	4.0	**30.0**	24.09	69.67	46.88	**81.0**	51.0	**66.0**	7.92	34.84	21.38
Agentic harness, shell interface												
Agentic shell	23.0	**6.0**	14.6	14.42	40.46	27.44	–	–	–	–	–	–
Agentic harness, native MCP												
Agentic MCP	31.0	–	31.0	10.24	–	10.24	88.0	63.0	75.5	15.67	41.00	28.33
+ persistent memory	32.0	–	32.0	9.91	–	9.91	85.0	62.0	73.5	15.30	52.42	33.86
+ memory + feedback	28.0	–	28.0	11.16	–	11.16	86.0	63.0	74.5	16.02	63.67	39.84

39.90 steps, and GEMINI-3.6-FLASH takes 39.29 steps. These runs often show models revisiting locations when they cannot see past images. In contrast, the online visual canvas shows fewer steps on Level 1. GPT-5.6-TERRA stops after an average of 6.91 steps compared to 24.09 steps with textual state, while GEMINI-3.1-PRO-PREVIEW stops after 4.12 steps compared to 7.18 steps. A partial canvas view may prompt models to stop searching early.

E AUTONOMOUS HARNESS EVALUATION

This section evaluates agents in tool-using harnesses discussed in Section 3.3.4. Table 5 reports accuracy and average step counts for GPT-5.6-TERRA and GEMINI-3.7-FLASH across standard baselines, offline canvas reconstruction, a shell interface executing bash actions, and a Model Context Protocol harness under different memory and feedback settings. In Table 5, writing and running image stitching scripts helps GEMINI-3.7-FLASH, with accuracy increasing from 68.0% to 88.0% on Level 1 and from 38.0% to 63.0% on Level 2. In contrast, persistent notes across runs and outcome feedback show little change in accuracy. For GEMINI-3.7-FLASH, average accuracy is 73.5% with notes and 74.5% with feedback, compared to 75.5% without notes. Steps on Level 2 also increase from 41.00 without notes to 52.42 with notes and 63.67 with feedback, as models spend more steps checking earlier areas.

E.1 HARNESS ENVIRONMENT ARCHITECTURE AND TOOL INTERFACE

To test whether models can invent and maintain state representations, we evaluate them within tool harnesses. Unlike in-process baselines described in Section A.4, where the framework manages state, the harness places the agent in a workspace with code execution and tool calling.

Tool Interface Primitives. The agent interacts with the environment through three tools. First, `move` shifts the glimpse window by $\delta = 32$ pixels in directions `up`, `down`, `left`, or `right`. Each move returns the glimpse as image content in context and saves a PNG file to the workspace. Second, `view_image` opens any PNG file in the workspace as image content, allowing the agent to inspect generated composite canvases. Third, `submit` records the predicted digit and ends the episode, taking values 0 to 9 for Level 1 and 00 to 99 for Level 2.

Security Isolation and Mailbox Architecture. To prevent data leakage, the environment separates the tool server from the episode controller. The tool server holds no labels, coordinates, or sensing schedules, relaying requests to the controller through a mailbox authenticated with HMAC-SHA256 tokens. Even if the agent inspects server processes or tool definitions, ground-truth state cannot be recovered. In addition, observation files on disk use random 128-bit hex names rather than sequential counters. This prevents models from inferring step counts or temporal order from filenames.

Cross-Episode Memory Configurations. To evaluate retention across tasks, we test three memory configurations. Under the stateless setting, no information carries over, and the workspace, context, and observation files are cleared between tasks. Under the persistent notes setting, a file named `NOTES.md` persists in the workspace across episodes. While conversation context and glimpse files are cleared between tasks, the agent can update `NOTES.md` to record procedures and coordinate strategies. Under the feedback setting, the controller returns a correct or incorrect result upon submission without revealing the true label, allowing the agent to refine notes across episodes.

E.2 Analysis of Autonomous Harness Behaviors

Procedural Discovery versus Execution in GPT-5.6-Terra. With persistent memory, agents can write information that survives across episodes. In our traces, GPT-5.6-Terra discovered a method to build an ASCII canvas and stored it in memory across subsequent episodes. However, GPT-5.6-Terra executed this method in only 2% of episodes, revealing a gap between discovering procedures and invoking them.

Procedural Standardization in Gemini-3.7-Flash. Traces show that Gemini-3.7-Flash reconstructs perceptual state rather than predicting from individual glimpses. The model writes Python scripts such as `stitch.py` to assemble observations into downsampled canvas images. Persistent memory standardizes this approach across episodes. Instructions, filenames, and decoding heuristics remain in memory and are reused over time. Without memory, Gemini-3.7-Flash stitched glimpses in 75% of Level 1 episodes. With memory, it stitched glimpses in 100% of episodes across both levels. Because memory retains instructions rather than executable files, the model re-implements scripts when needed.

Over-Exploration with Persistent Notes. Persistent memory does not improve accuracy, yielding 73.5% with memory compared to 75.5% without memory for Gemini-3.7-Flash. Instead, memory increases search steps. On Level 2, exploration rises from 41.00 steps without memory to 52.42 steps with memory, and to 63.67 steps with correctness feedback. Retaining notes leads agents to revisit confirmed strokes repeatedly rather than pruning redundant moves.

Level 2 Traversal Mechanics under Tool Autonomy. On Level 2, Gemini-3.7-Flash reconstructs the digit around the initial glimpse, crosses the canvas center to the second digit, and returns periodically for verification. The model encounters the second digit after a median of 17 observations, when the first digit has reached a median coverage of 94.1%. The agent crosses the canvas midpoint once in 57% of episodes and twice in 21% of episodes, with immediate reversals in only 2.8% of moves. The resulting Level 2 accuracy of 63.0% reflects multi-pass search rather than random exploration.

Visual State Interpretation as the Primary Bottleneck. Harness failures do not stem from insufficient coverage. For Gemini-3.7-Flash, incorrect episodes reach 98.01% stroke coverage and 3.75 stitching passes compared to 94.88% coverage and 2.10 passes for correct episodes. Under multi-turn interaction without tools, incorrect episodes exhibit lower coverage than correct episodes,

Table 6: Task success rates by topological complexity across the three configurations. Digits are grouped into simple strokes (1, 4, 7), high-curvature curves (2, 3, 5), and closed loops (0, 6, 8, 9). **Bold** values show the best performance for each model within each complexity category across the three configurations.

Model	Image Only Baseline ($\lvert\mathcal{H}_v\rvert = \infty$)			Textual State ($\lvert\mathcal{H}_v\rvert = 1$)			Metric Grid Map ($\lvert\mathcal{H}_v\rvert = 1$)		
	Strokes (%) ↑	Curves (%) ↑	Loops (%) ↑	Strokes (%) ↑	Curves (%) ↑	Loops (%) ↑	Strokes (%) ↑	Curves (%) ↑	Loops (%) ↑
Proprietary Models									
GEMINI-3.1-PRO-PREVIEW	43.3	**56.7**	20.0	**66.7**	53.3	**50.0**	56.7	**56.7**	47.5
GEMINI-3.6-FLASH	60.0	**50.0**	**50.0**	**63.3**	30.0	47.5	**63.3**	36.7	47.5
GEMINI-3.7-FLASH	**76.7**	53.3	**65.0**	66.7	36.7	57.5	73.3	**56.7**	45.0
CLAUDE-5-SONNET	**50.0**	13.3	12.5	23.3	20.0	**22.5**	23.3	**23.3**	12.5
CLAUDE-5-OPUS	56.7	**43.3**	32.5	**60.0**	23.3	**40.0**	43.3	33.3	32.5
CLAUDE-5-FABLE	**66.7**	**26.7**	47.5	60.0	23.3	32.5	56.7	**26.7**	**50.0**
GPT-5.6-TERRA	**50.0**	**40.0**	**20.0**	30.0	26.7	**20.0**	26.7	26.7	**20.0**
GPT-5.6-SOL	**50.0**	**30.0**	**32.5**	46.7	26.7	17.5	**50.0**	20.0	27.5
GPT-5.6-SOL (xhigh)	**76.7**	**40.0**	**52.5**	50.0	30.0	32.5	53.3	33.3	32.5
Open-Source Models									
QWEN-3.8-27B	36.7	**33.3**	17.5	43.3	26.7	**25.0**	**56.7**	**33.3**	22.5
GLM-4.6V	33.3	6.7	5.0	**36.7**	**20.0**	7.5	30.0	**20.0**	**12.5**

confirming that evidence acquisition remains a bottleneck in standard chat setups. Tool harnesses thus resolve the coverage bottleneck, leaving visual interpretation as the main error source. These recognition failures do not come from script rendering errors or border occlusions. Self-generated canvases improve accuracy for GEMINI-3.7-FLASH by 20.0 to 25.0 percentage points over online memory canvases. In addition, the 2-pixel observation border covers less than 2.2% of stroke pixels, and recognition errors persist on programmatic canvases in Section 4 and Table 10.

F TASK SUCCESS BY TOPOLOGICAL COMPLEXITY

This section evaluates performance across digit shapes and observation settings. Table 6 presents results for digits grouped into straight strokes for 1, 4, and 7, curves for 2, 3, and 5, and closed loops for 0, 6, 8, and 9. Across models, straight stroke digits show the highest accuracy, reaching 76.7% for GEMINI-3.7-FLASH and GPT-5.6-SOL (xhigh) in the Image Only baseline. Accuracy tends to be lower on curves and closed loops, especially under textual state with history length $|\mathcal{H}_v| = 1$. For example, loop accuracy in Image Only is 20.0% for GPT-5.6-TERRA and 12.5% for CLAUDE-5-SONNET. Metric grid maps show higher loop accuracy for some models, reaching 45.0% for GEMINI-3.7-FLASH and 50.0% for CLAUDE-5-FABLE.

G EXPLORATION AND REVISIT RATE METRICS

This section defines movement and coverage metrics across Image Only, Textual State, and Metric Grid Map configurations. Results are shown in Table 7.

Mathematical Formulations. We represent the canvas as a binarized grid I of size $W' \times H$ where stroke pixels have value 0. The set of stroke pixels S is defined below.

$$S = \{(u, v) \in \mathbb{N}^2 \mid I(u, v) = 0,\ 0 \le u < W',\ 0 \le v < H\} \tag{10}$$

At step t, window B_t of size $h \times w$ covers pixels starting at coordinate p_t.

$$B(x_t, y_t) = \{(u, v) \in \mathbb{N}^2 \mid x_t \le u < x_t + w,\ y_t \le v < y_t + h\} \tag{11}$$

The explored region V_T combines all glimpse windows visited up to step T.

$$V_T = \bigcup_{t=0}^{T} B(x_t, y_t) \tag{12}$$

Stroke coverage C_T is the fraction of stroke pixels S within the explored region V_T.

$$C_T = \frac{|S \cap V_T|}{|S|} = \frac{\left|S \cap \left(\bigcup_{t=0}^{T} B(x_t, y_t)\right)\right|}{|S|} \tag{13}$$

Table 7: Exploration and revisit rate metrics, reporting Average Moves (M_T), Revisit Rate ($RR_T \downarrow$), and Stroke Coverage ($C_T \uparrow$) for Level 1 (Single-Digit) and Level 2 (Two-Digit) tasks across three observation configurations. Bold values indicate the lowest revisit rate or highest stroke coverage within the corresponding model and task configuration.

Model	Image Only ($\lvert\mathcal{H}_v\rvert = \infty$)						Textual State ($\lvert\mathcal{H}_v\rvert = 1$)						Metric Grid Map ($\lvert\mathcal{H}_v\rvert = 1$)					
	Level 1			Level 2			Level 1			Level 2			Level 1			Level 2		
	M_T	RR_T (%) $\downarrow$	C_T (%) $\uparrow$	M_T	RR_T (%) $\downarrow$	C_T (%) $\uparrow$	M_T	RR_T (%) $\downarrow$	C_T (%) $\uparrow$	M_T	RR_T (%) $\downarrow$	C_T (%) $\uparrow$	M_T	RR_T (%) $\downarrow$	C_T (%) $\uparrow$	M_T	RR_T (%) $\downarrow$	C_T (%) $\uparrow$
Proprietary Models																		
GEMINI-3.1-PRO-PREVIEW	3.5	3.7	57.7	10.9	**10.0**	40.4	6.2	5.3	**73.8**	17.3	13.6	**55.9**	6.2	**3.6**	71.7	18.2	20.2	52.9
GEMINI-3.6-FLASH	4.6	**5.6**	65.9	14.1	**10.6**	49.0	8.3	16.3	**77.9**	38.3	45.0	**65.4**	6.9	6.1	71.0	19.4	13.3	56.4
GEMINI-3.7-FLASH	4.3	**4.2**	66.8	13.7	**6.3**	53.9	6.9	15.8	**74.4**	33.8	45.1	61.0	5.8	8.1	69.3	19.4	13.6	**61.7**
CLAUDE-5-SONNET	16.4	46.7	71.1	44.1	54.7	45.1	13.8	34.9	72.2	74.6	64.2	50.6	12.1	**20.2**	**72.9**	49.4	**46.6**	**56.4**
CLAUDE-5-OPUS	6.4	9.6	72.9	21.5	20.2	61.7	10.6	18.8	**83.0**	38.9	25.4	**80.5**	9.6	**8.6**	81.5	28.6	**13.6**	75.4
CLAUDE-5-FABLE	6.3	7.3	74.7	21.3	12.6	68.7	10.3	17.0	**80.5**	34.9	19.7	**82.0**	8.9	**6.5**	76.8	27.6	**9.7**	76.6
GPT-5.6-TERRA	17.2	35.8	71.0	38.7	49.3	41.5	23.1	44.4	78.1	68.7	71.6	44.4	20.9	**26.2**	**81.3**	61.4	**46.1**	**55.6**
GPT-5.6-SOL	5.5	12.5	58.5	16.0	26.3	35.4	6.0	20.3	60.7	21.5	33.0	**42.5**	6.9	**8.1**	**64.0**	20.3	**22.2**	41.1
GPT-5.6-SOL (xhigh)	8.6	**9.9**	75.3	34.4	**20.7**	**75.8**	8.2	13.9	**75.3**	27.0	25.8	55.5	8.8	8.6	71.8	35.1	24.9	58.4
Open-Source Models																		
QWEN-3.8-27B	3.8	22.1	48.1	8.2	39.7	25.4	6.0	34.2	53.0	15.9	53.6	32.1	5.9	**16.4**	**56.2**	12.9	**29.7**	**35.6**
GLM-4.6V	2.5	**20.1**	36.5	5.5	**17.2**	20.7	19.2	61.8	**49.4**	39.1	62.3	**36.3**	10.4	39.0	46.7	26.7	55.2	33.1

The revisit rate RR_T measures movement redundancy, where R_T counts steps returning to earlier window coordinates.

$$R_T = \sum_{t=1}^{T} \mathbb{I}\Big((x_t, y_t) \in \{(x_k, y_k)\}_{k=0}^{t-1}\Big) \tag{14}$$

Here $\mathbb{I}$ denotes the indicator function, and the rate RR_T divides revisits R_T by total movement actions M_T.

$$RR_T = \frac{R_T}{M_T} \tag{15}$$

Exploration and Coverage Observations. Table 7 reports results across models and settings. Metric grid maps show lower revisit rates than textual state on Level 2. The revisit rate decreases from 45.1% to 13.6% for GEMINI-3.7-FLASH, from 71.6% to 46.1% for GPT-5.6-TERRA, and from 64.2% to 46.6% for CLAUDE-5-SONNET. In addition, high stroke coverage does not necessarily translate to high accuracy. On Level 2 with textual state, CLAUDE-5-OPUS covers 80.5% of strokes and CLAUDE-5-FABLE covers 82.0% of strokes, while both score low accuracy. In Image Only, models with lower revisit rates also show lower Level 2 coverage. GEMINI-3.1-PRO-PREVIEW covers 40.4% of strokes, while QWEN-3.8-27B covers 25.4%.

H VISUAL-HISTORY HORIZON ROBUSTNESS STUDY

We evaluate performance across glimpse history lengths $|\mathcal{H}_v| \in \{1, 2, 4, \infty\}$, summarized in Table 8. At $|\mathcal{H}_v| = 1$, the agent operates without visual lookback and relies on current observations and memory. Settings with $|\mathcal{H}_v| = 2$ and $|\mathcal{H}_v| = 4$ preserve recent glimpses in context, while $|\mathcal{H}_v| = \infty$ allows lookback across conversation history.

Expanding Glimpse History Enhances Exploration Efficiency. Retaining past glimpses reduces exploration steps across models and tasks. On Level 1 with textual state representations, increasing history length from $|\mathcal{H}_v| = 1$ to $|\mathcal{H}_v| = \infty$ reduces step counts, falling from 7.18 to 5.48 for GEMINI-3.1-PRO-PREVIEW, from 9.35 to 5.45 for GEMINI-3.6-FLASH, and from 7.92 to 5.44 for GEMINI-3.7-FLASH. On Level 2, steps decrease further under textual states, dropping from 39.29 to 15.89 for GEMINI-3.6-FLASH and from 34.84 to 15.90 for GEMINI-3.7-FLASH.

Extended Visual History Improves Accuracy. Longer visual history increases accuracy on tracking tasks. On Level 1, extending lookback from $|\mathcal{H}_v| = 1$ to $|\mathcal{H}_v| = \infty$ under textual states raises accuracy from 54.0% to 64.0% for GEMINI-3.7-FLASH and from 47.0% to 58.0% for GEMINI-3.6-FLASH. On Level 2, GEMINI-3.7-FLASH accuracy increases from 6.0% at $|\mathcal{H}_v| = 1$ to 23.0% at $|\mathcal{H}_v| = \infty$ with textual states, and from 28.0% to 35.0% with metric grid maps. Visual lookback thus limits coordinate drift and improves final decisions.

Table 8: Ablation results for varying glimpse history $|\mathcal{H}_v| \in \{1, 2, 4, \infty\}$ on Level 1 (Single-Digit) and Level 2 (Two-Digit) tasks under both the Textual State and Metric Grid Map configurations. Bold values highlight the best performance (highest accuracy ↑ or lowest average steps ↓) within each model block across the different history lengths.

Model	History ($\lvert\mathcal{H}_v\rvert$)	Level 1 (Single-Digit)				Level 2 (Two-Digit)			
		Textual State		Metric Grid Map		Textual State		Metric Grid Map	
		Acc (%) ↑	Avg. Steps ↓	Acc (%) ↑	Avg. Steps ↓	Acc (%) ↑	Avg. Steps ↓	Acc (%) ↑	Avg. Steps ↓
GEMINI-3.1-PRO-PREVIEW	1	56.0	7.18	53.0	7.15	**20.0**	18.28	14.0	19.19
	2	48.0	5.99	53.0	6.75	16.0	19.29	14.0	17.91
	4	50.0	5.89	49.0	**6.50**	15.0	16.65	16.0	**16.83**
	∞	**57.0**	**5.48**	**57.0**	6.56	16.0	**16.40**	**22.0**	17.06
GEMINI-3.6-FLASH	1	47.0	9.35	49.0	7.92	**15.0**	39.29	10.0	20.44
	2	52.0	7.33	45.0	7.09	12.0	32.39	**18.0**	18.83
	4	53.0	5.97	**54.0**	6.06	**15.0**	23.80	14.0	17.07
	∞	**58.0**	**5.45**	51.0	**5.82**	13.0	**15.89**	13.0	**16.53**
GEMINI-3.7-FLASH	1	54.0	7.92	57.0	6.82	6.0	34.84	28.0	20.37
	2	60.0	6.80	63.0	6.17	10.0	30.68	23.0	19.96
	4	62.0	5.85	60.0	5.63	16.0	26.65	20.0	19.56
	∞	**64.0**	**5.44**	**64.0**	**5.49**	**23.0**	**15.90**	**35.0**	**17.68**

Table 9: Glimpse window dimension ablation study results under varying glimpse window sizes ($w \times w$) on Single-Digit (Level 1) and Two-Digit (Level 2) tasks. Results are evaluated under both zero-visual-lookback ($|\mathcal{H}_v| = 1$) and unbounded history ($|\mathcal{H}_v| = \infty$).

Model	Window	Image Only Baseline ($\lvert\mathcal{H}_v\rvert=\infty$)						Textual State ($\lvert\mathcal{H}_v\rvert=1$)						Metric Grid Map ($\lvert\mathcal{H}_v\rvert=1$)					
		Acc. (%) ↑			Steps ↓			Acc. (%) ↑			Steps ↓			Acc. (%) ↑			Steps ↓		
		L1	L2	Avg	L1	L2	Avg	L1	L2	Avg	L1	L2	Avg	L1	L2	Avg	L1	L2	Avg
GEMINI-3.1-PRO-PREVIEW	128 × 128	88.0	61.0	74.5	2.48	6.53	4.50	79.0	62.0	70.5	3.41	8.35	5.88	82.0	51.0	66.5	3.19	7.43	5.31
	64 × 64	38.0	5.0	21.5	4.52	11.95	8.23	56.0	20.0	38.0	7.18	18.28	12.73	53.0	14.0	33.5	7.15	19.19	13.17
	32 × 32	30.0	0.0	15.0	8.41	19.75	14.08	22.0	1.0	11.5	12.92	35.76	24.34	21.0	4.0	12.5	12.65	35.85	24.25
GEMINI-3.6-FLASH	128 × 128	85.0	61.0	73.0	2.48	6.39	4.43	78.0	61.0	69.5	3.26	12.06	7.66	75.0	41.0	58.0	3.05	9.99	6.52
	64 × 64	53.0	18.0	35.5	5.62	15.11	10.37	47.0	15.0	31.0	9.35	39.29	24.32	49.0	10.0	29.5	7.92	20.44	14.18
	32 × 32	24.0	2.0	13.0	11.50	30.41	20.95	19.0	3.0	11.0	17.70	55.75	36.73	21.0	3.0	12.0	14.25	30.73	22.49
GEMINI-3.7-FLASH	128 × 128	89.0	74.0	81.5	2.48	6.02	4.25	83.0	60.0	71.5	2.99	11.14	7.07	84.0	67.0	75.5	2.64	7.84	5.24
	64 × 64	65.0	22.0	43.5	5.26	14.69	9.97	54.0	6.0	30.0	7.92	34.84	21.38	57.0	28.0	42.5	6.82	20.37	13.60
	32 × 32	30.0	3.0	16.5	12.80	33.35	23.08	28.0	2.0	15.0	16.16	49.98	33.07	30.0	10.0	20.0	13.52	40.40	26.96

I GLIMPSE WINDOW DIMENSIONS ROBUSTNESS STUDY

Larger glimpse windows reduce exploration steps and improve accuracy, as reported in Table 9. Expanding window size from 32×32 to 128×128 pixels increases accuracy across models. For example, Level 2 accuracy for GEMINI-3.7-FLASH rises from 3.0% to 74.0% under the Image Only baseline. Larger windows provide more visual evidence per step and reduce the moves needed to cover the canvas, limiting tracking errors.

J DECOUPLING EXPLORATION POLICY FROM STATE INTERPRETATION

This section presents predictor-swap experiments discussed in Section 4. To determine whether recognition errors stem from exploration failures or state interpretation bottlenecks, we replay identical trajectories and evaluate different model heads on reconstructed canvases.

Table 10 reports results across Level 1 and Level 2 tasks. Fixing the exploration trajectory to GPT-5.6-SOL (xhigh), we compare the native model against an upgraded predictor using GEMINI-3.7-FLASH. Upgrading the predictor improves accuracy by 17.8 percentage points on Level 1 and by 15.0 points on Level 2 Image Only, confirming that state interpretation is an independent bottleneck.

Table 10 also breaks down Level 2 Textual State performance by visual coverage. In this condition, GPT-5.6-SOL (xhigh) observed both digits in only 38% of episodes. When both digits appear on the canvas, GEMINI-3.7-FLASH reaches 86.8% accuracy compared to 68.4% for the native predictor. When exploration misses either digit in the remaining 62% of episodes, accuracy drops to 0.0% for both models. Downstream interpretation is therefore bounded by upstream coverage, since models cannot classify unobserved digits.

Finally, Table 10 reports cross-trajectory exploration asymmetry using GEMINI-3.7-FLASH as the fixed predictor across canvases generated by different exploration policies. Under full visual history with Image Only, GPT-5.6-SOL (xhigh) explores more effectively, enabling GEMINI-3.7-FLASH to reach 74.0% on GPT traces compared to 41.0% on Gemini traces, a gain of 33.0 percentage

Table 10: **Controlled predictor-swap and cross-trajectory evaluation.** (Top) Holding the exploration policy fixed to GPT-5.6-SOL (xhigh) while swapping the predictor head to GEMINI-3.7-FLASH isolates state interpretation from exploration dynamics. When both digits are captured on canvas, the swapped predictor achieves 86.8%, but drops to 0.0% under incomplete coverage. (Bottom) Holding the predictor fixed to GEMINI-3.7-FLASH while evaluating across different exploration trajectories reveals a capability asymmetry between visual and textual memory configurations.

Task Condition	Exploration Trajectory	Predictor Head	Native (%)	Swapped (%)	Δ
Predictor Swap with Fixed Exploration (GPT-5.6-SOL (xhigh)) and Upgraded Predictor (GEMINI-3.7-FLASH)					
Level 1 (Single-Digit)	GPT-5.6-SOL (xhigh)	GPT-5.6-SOL (xhigh) → GEMINI-3.7-FLASH	66.5	**84.3**	+17.8
Level 2 Image Only ($\|\mathcal{H}_v\|=\infty$)	GPT-5.6-SOL (xhigh)	GPT-5.6-SOL (xhigh) → GEMINI-3.7-FLASH	59.0	**74.0**	+15.0
Level 2 Textual State ($\|\mathcal{H}_v\|=1$)	GPT-5.6-SOL (xhigh)	GPT-5.6-SOL (xhigh) → GEMINI-3.7-FLASH	26.0	**33.0**	+7.0
Both digits visible (38% of episodes)	GPT-5.6-SOL (xhigh)	GPT-5.6-SOL (xhigh) → GEMINI-3.7-FLASH	68.4	**86.8**	+18.4
Single digit visible (62% of episodes)	GPT-5.6-SOL (xhigh)	GPT-5.6-SOL (xhigh) → GEMINI-3.7-FLASH	0.0	0.0	0.0
Cross-Trajectory Exploration Asymmetry with Fixed Predictor (GEMINI-3.7-FLASH) and Swapped Exploration Policy					
Level 2 Image Only ($\|\mathcal{H}_v\|=\infty$)	GEMINI-3.7-FLASH → GPT-5.6-SOL (xhigh)	GEMINI-3.7-FLASH	41.0	**74.0**	+33.0
Level 2 Textual State ($\|\mathcal{H}_v\|=1$)	GPT-5.6-SOL (xhigh) → GEMINI-3.7-FLASH	GEMINI-3.7-FLASH	33.0	**51.0**	+18.0

points. Under zero visual lookback with Textual State, GEMINI-3.7-FLASH plans trajectories more effectively, reaching 51.0% on Gemini traces compared to 33.0% on GPT traces, a gain of 18.0 percentage points.

K QUALITATIVE EXPLORATION TRAJECTORIES ACROSS STATE CONSTRUCTION METHODS

This section presents exploration trajectories across five state construction methods using GEMINI-3.7-FLASH. Figure 5 shows single-digit Level 1 episodes, and Figure 6 shows two-digit Level 2 sequences. In each trajectory panel, circle marker S indicates the start coordinate and diamond marker F indicates the final position. Directed arrows transition from dark blue early steps to gold late steps. Below each panel, we list total steps T, predicted label pred, and whether the outcome was correct or incorrect.

K.1 LEVEL 1 SINGLE-DIGIT EXPLORATION DYNAMICS

Premature Termination versus Tracking Drift. In-loop configurations exhibit contrasting failure modes on single digits, as shown in Figure 5. With unbounded visual history $|\mathcal{H}_v| = \infty$, GEMINI-3.7-FLASH stops after brief trajectories of $T = 4$ to 7 steps and misclassifies digits, such as predicting 8 for digit 2 at $T = 6$, 2 for digit 5 at $T = 6$, 4 for digit 6 at $T = 4$, and 5 for digit 7 at $T = 6$. Because glimpses lack coordinate alignment, the model predicts before resolving stroke contours. With zero lookback $|\mathcal{H}_v| = 1$, textual memory degrades tracking, causing loops and coordinate drift such as $T = 12$ on digit 5 and $T = 10$ on digit 9 without reaching correct answers.

Coordinate Anchoring and Autonomous Tracing. External state representations replace local sampling with structured search. A metric grid map anchors coordinates, reducing revisits and constraining movement within digit boundaries when tracing loops in digits 0, 8, and 9. The visual canvas improves recognition, correctly identifying digits 2, 5, 7, 8, and 9 within $T = 2$ to 10 steps. The autonomous harness drives structured exploration by running scripts such as `stitch.py` to project glimpses onto a canvas. Here GEMINI-3.7-FLASH traces stroke contours with trajectories from $T = 12$ to 34 steps, including $T = 34$ on digit 5 and $T = 27$ on digit 7, achieving 100% accuracy across all ten classes.

K.2 LEVEL 2 TWO-DIGIT TRAJECTORY ANALYSIS AND BOUNDARY BLINDNESS

Inter-Digit Boundary Blindness in In-Loop Agents. Scaling the active vision task to two digits on a 224×448 canvas exposes inter-digit boundary blindness, as shown in Figure 6. When navigating empty space between digits, in-loop methods mistake the central gap for an outer canvas edge and remain confined within a single digit. For sequence `49` in Row 4, where the episode begins on the right digit, Textual State at $T = 42$ and Metric Grid Map at $T = 19$ search only the right digit `9`

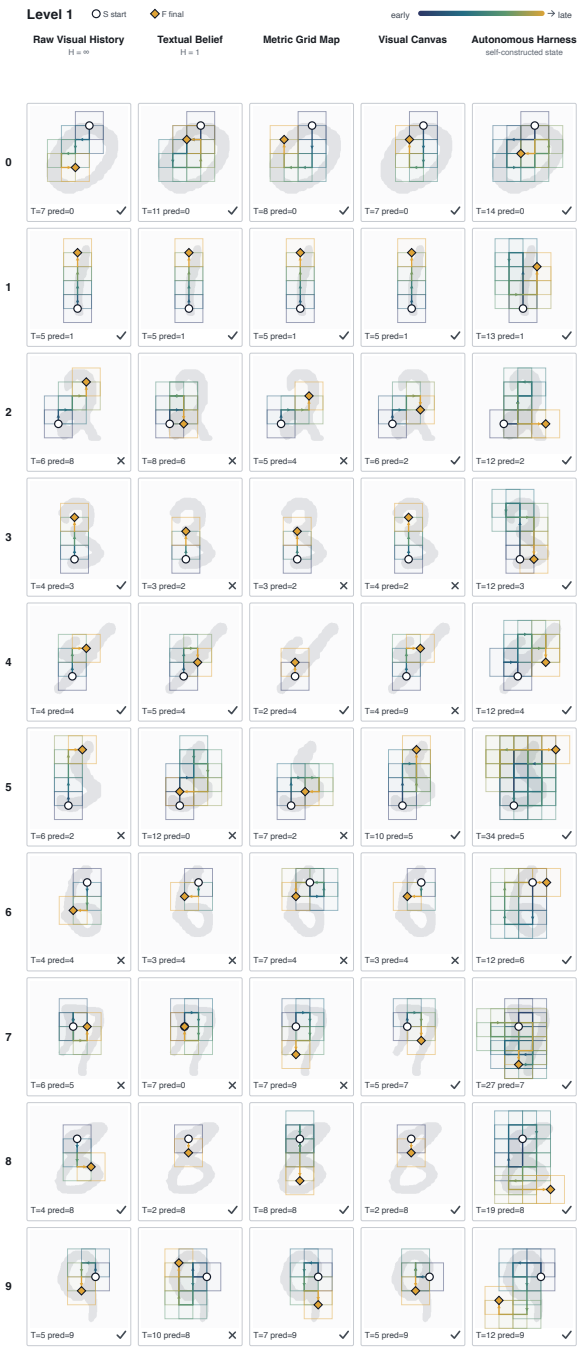


Figure 5: **Qualitative exploration trajectories across five state construction methods with GEMINI-3.7-FLASH on Level 1 (Single-Digit).** Columns represent Image Only ($|\mathcal{H}_v| = \infty$), Textual State ($|\mathcal{H}_v| = 1$), Metric Grid Map ($|\mathcal{H}_v| = 1$), Visual Canvas, and Autonomous Harness (code execution). Circles (◯) denote initial glimpse positions (S), and diamonds (◊) mark final termination positions (F). Directed arrows transition from early (navy) to late (gold) steps. Sub-labels indicate total sensing steps (T), model prediction (pred), and classification outcome (✓ / ×). In-loop baselines suffer from premature stopping or tracking drift, whereas the autonomous harness systematically traces stroke contours ($T = 12$–34), achieving 10/10 accuracy.

without crossing into the left digit `4`. Similarly, for sequence `19` in Row 1, Textual State spends 68 steps cycling around digit `9` without reaching digit `1`.

Autonomous Multi-Pass Traversal Overcomes Spatial Boundaries. The autonomous harness overcomes boundary blindness by using programmatic tools to coordinate search. Guided by programmatic state tracking, GEMINI-3.7-FLASH spans both canvas halves with trajectories between $T = 30$ and 61 steps across all ten sequences. As shown in the right column of Figure 6, the agent moves across the central boundary between digits and correctly classifies 5 of 10 sequences.

Qualitative Examples of the Evidence-to-Decision Bottleneck. Figure 6 also illustrates the evidence-to-decision bottleneck identified in Section 3.3.2. For sequence `05` at $T = 57$, sequence `51` at $T = 30$, and sequence `87` at $T = 40$, the harness agent achieves high stroke coverage across both digits, yet predicts `03`, `01`, and `43`. Because visual evidence was gathered and stitched onto the canvas, these errors do not stem from tracking loss or missing glimpses. Instead, downstream visual state interpretation remains the primary bottleneck when converting assembled features into final predictions.

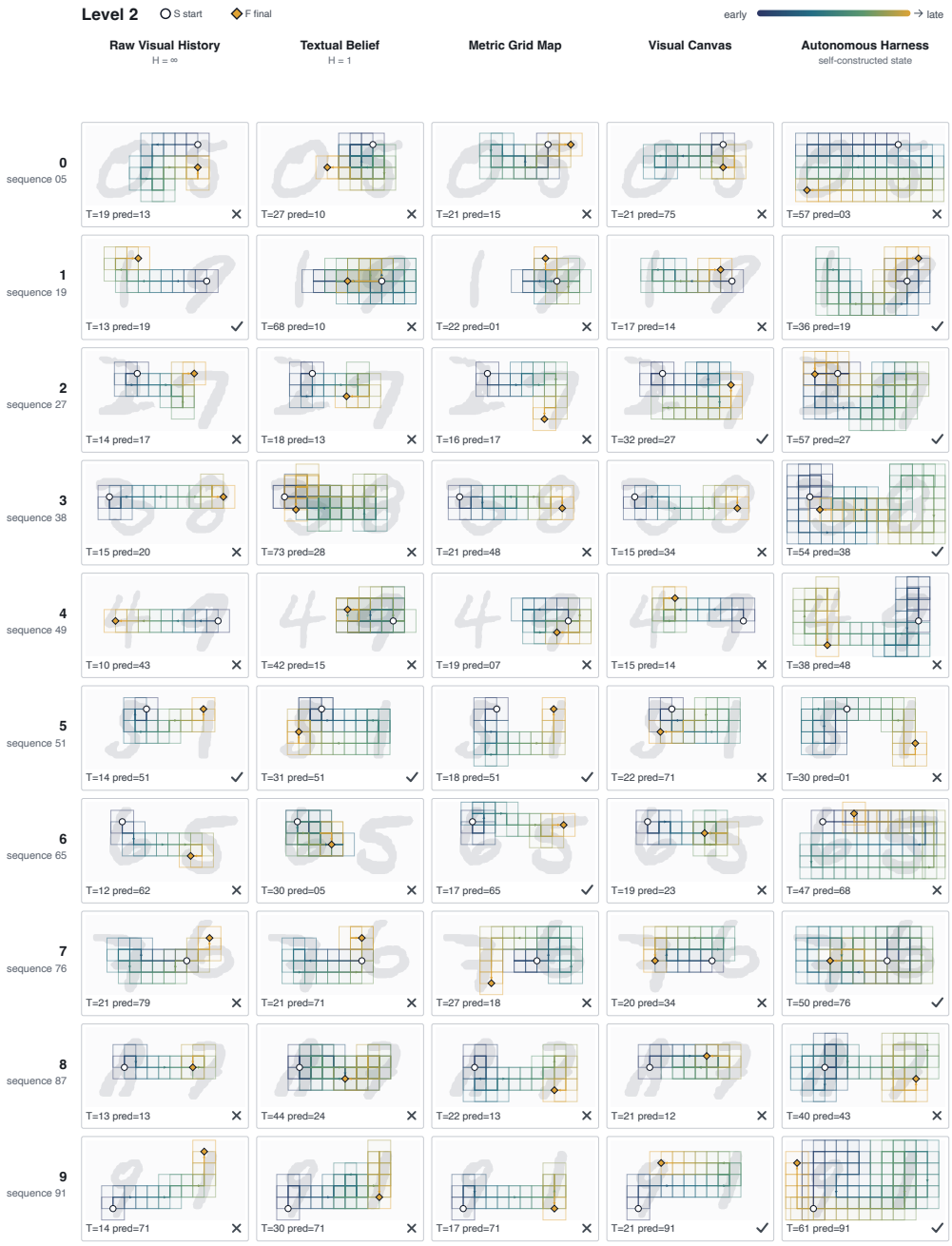


Figure 6: **Qualitative exploration trajectories across five state construction methods with GEMINI-3.7-FLASH on Level 2 (Two-Digit).** Rows depict ten representative two-digit sequences on a 224×448 canvas. In-loop agents at times focus on only one digit, missing the other entirely (such as for sequence `49`, where Textual State and Metric Grid Map miss the left digit). Conversely, the autonomous harness agent reliably spans both canvas halves ($T = 30$–61). However, sequences `05`, `51`, and `87` produce misclassifications despite high two-digit stroke coverage, demonstrating the evidence-to-decision bottleneck.