WHEN SHOULD GUI AGENTS LOOK AGAIN? SEPARATING OBSERVATION AND EXECUTION TIMING

Anonymous authors
Paper under double-blind review

ABSTRACT

GUI agents must decide not only what action to take, but when a post-action screenshot is informative enough to request the next action. Existing fixed delays and global pixel-stability rules conflate two events: observation maturity and execution readiness. We introduce a causal timing protocol based on identical-seed counterfactual replay that independently varies the screenshot used by a frozen vision-language actor and the time at which its action is executed. Across two MiniWoB task families, premature observations remain wrong even when execution is delayed, whereas mature observations can fail when executed during a transition. Controlled static-pending and moving-ready interfaces expose complementary failures of global stability. We then train an instruction-conditioned visual-readiness controller from counterfactual actor outcomes. On 80 unseen frozen episodes, the controller matches a validation-locked 0.5-second wait at 83.75% success, reduces observation time by 73.6% (0.500 to 0.132 seconds), and reduces counterfactual next-step completion time by 26.4% (1.448 to 1.066 seconds). These results make post-action observation timing a measurable control problem and show how visual readiness enables earlier task-relevant observation at matched success.

1 INTRODUCTION

Vision-language agents interact with graphical interfaces through a repeated loop of observation, inference, and action. Most work optimizes the action itself, yet every action also creates a temporal decision: when should the agent look again? A premature screenshot may omit the target or capture it at a transient location. A conservative delay slows every interaction, including interfaces that are ready immediately. The problem is particularly acute in graphical interfaces, where updates are asynchronous, localized, and often unrelated to the current task.

Practical agents typically wait for a fixed interval or for global image differences to vanish. Both strategies treat two events as one. First, the screenshot must contain enough task-relevant evidence for the actor to select the next action. Second, the interface must be ready to receive that action when it arrives. These events need not coincide. A target may already be visible while an unrelated region keeps moving, or the screen may be static while the target remains unavailable.

We study this overlooked decision as *post-action observation timing*. We call a screenshot *mature* when it contains sufficient task-relevant evidence for a frozen actor. We call the interface *execution-ready* when it can accept the resulting action at arrival. Our identical-seed counterfactual protocol varies these factors independently. It collects actions from early and late screenshots, then replays each frozen action at early and late execution times in fresh episodes. The design separates errors caused by what the actor saw from errors caused by when its action landed.

The resulting 2×2 timing matrix reveals a consistent pattern across two MiniWoB task families. Early observations produce incorrect coordinates, and delaying execution does not repair them. Mature observations improve aggregate outcomes, but can still fail when the action lands during a transition. Matched no-delay controls show that the effect is not explained by general grounding difficulty. We then construct two controlled interfaces that expose complementary failures of global pixel stability: a static-but-pending interface, where the screen is unchanged before the target mounts, and a moving-but-ready interface, where irrelevant motion continues after the target is actionable.

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.

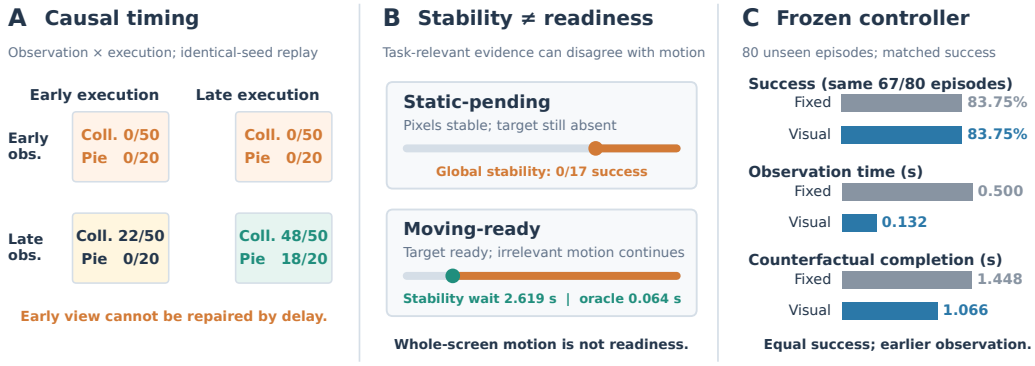

Figure 1: **Separating timing failures and selecting informative observations.** (A) Identical-seed replay separates observation maturity from execution readiness. (B) Static-pending and moving-ready interfaces expose opposite failures of global pixel stability. (C) On 80 unseen frozen episodes, visual readiness matches the fixed wait on every episode-level success outcome while selecting earlier observations and reducing counterfactual next-step completion time.

These findings motivate an instruction-conditioned visual-readiness controller. At each candidate time, it chooses whether to observe again or commit the current screenshot to the frozen actor. Counterfactual rollouts supervise this decision with the realized outcome of querying the actor at that moment. On 80 unseen frozen episodes, the controller preserves the success rate of a validation-locked fixed wait while selecting substantially earlier observations.

Our contributions are:

- We formulate post-action observation timing as a control problem distinct from action execution timing and provide an identical-seed counterfactual replay protocol for measuring both factors separately.
- We show across two MiniWoB task families and controlled dynamic interfaces that premature observation and insufficient execution readiness create different failure modes, while global pixel stability is not a reliable readiness signal.
- We introduce an instruction-conditioned visual-readiness controller that preserves the aggregate and episode-level success of a validation-locked fixed wait while reducing mean observation time by 73.6% and counterfactual next-step completion time by 26.4% on unseen rollouts.

2 Related Work

GUI agents and multimodal interaction. GUI-agent research has progressed from controlled web interaction in World of Bits to realistic web and desktop benchmarks such as WebArena, Mind2Web, VisualWebArena, and OSWorld (Shi et al., 2017; Zhou et al., 2023; Deng et al., 2023; Koh et al., 2024; Xie et al., 2024). Multimodal agents such as WebVoyager further operate directly on live websites from screenshots (He et al., 2024). These benchmarks primarily evaluate action prediction and task completion across websites or desktop applications. We isolate a complementary decision that occurs between actions: when the next screenshot should be acquired and sent to the actor.

Vision-language grounding for interface actions. Vision-language models increasingly translate screenshots and instructions into coordinates or structured GUI actions. CogAgent, SeeClick, and ShowUI specialize visual representations and action prediction for GUI interaction, while ScreenAI and OmniParser improve UI understanding and screen parsing (Hong et al., 2024; Cheng et al., 2024; Lin et al., 2025; Baechler et al., 2024; Lu et al., 2024). These methods improve what an actor can infer from a screenshot. Our question is orthogonal: whether the screenshot was acquired at an informative moment. We freeze the actor so timing effects are not conflated with representation learning or action-parser changes.

Table 1: Counterfactual timing design.

Condition	Observation	Execution
A	early	early
B	early	late
C	late	early
D	late	late

Temporal control and adaptive observation. Event-triggered control initiates computation or communication only when a state-dependent condition is met (Heemels et al., 2012). GUI automation commonly offers fixed waits or externally defined readiness events, while vision-only agents can instead monitor elapsed time or frame change. Our setting differs because the relevant event is semantic: the current image must support the next actor decision. The proposed replay protocol further separates this observation condition from the later readiness of the interface to receive the action.

3 Problem Formulation

3.1 Post-action timing

Let an opening action define time zero. In a source episode with seed s, a frozen actor produces $a_o = f(I_s(o), u_s)$ from screenshot $I_s(o)$, instruction u_s, and observation time o. In a fresh identically seeded episode, the collected action is replayed at absolute time e. We define

$$Y_s(o, e) = \mathbb{1}[\text{replaying } a_o \text{ at time } e \text{ succeeds}]. \tag{1}$$

Observation time o controls the evidence used to select the action, while execution time e controls the state receiving it. Replay permits $e < o$, as in condition C of Table 1. This cell is a mechanism intervention across separate episodes, not a deployable policy that observes the future.

3.2 Counterfactual frozen-actor replay

For each seed, we collect actions from an early screenshot $I_s(t_e)$ and a late screenshot $I_s(t_l)$. We replay each action in fresh episodes at execution delays d_e and d_l. Because replay performs no new semantic actor query, differences between cells are attributable to timing and the frozen action itself. We use strict success, requiring reward 1 and a terminal environment state.

3.3 Readiness decision

In online control, the agent observes a sequence of candidate frames and chooses $z_t \in \{\text{OBSERVE}, \text{COMMIT}\}$. OBSERVE acquires another frame. COMMIT queries the frozen actor on the current frame and executes the returned action. The objective is to preserve task success while minimizing time before the actor query.

4 Visual Readiness Controller

4.1 Observable representation

The controller receives the task instruction, current and preceding screenshot representations, elapsed time, and whole-screen pixel change. A frozen Qwen2.5-VL-3B-Instruct actor (Bai et al., 2025) produces instruction-conditioned visual states for both frames. No DOM state, environment reward, target coordinates, or future frames are provided. For hidden states h_t and h_{t-1}, the full feature vector is

$$x_t = [h_t; h_{t-1}; h_t - h_{t-1}; \Delta t; \Delta_{\text{pix}}(I_t, I_{t-1})]. \tag{2}$$

4.2 BINARY CONTROLLER

We use a multilayer perceptron $g_\theta(x_t)$ with one hidden layer and scalar output

$$p_t = \sigma(g_\theta(x_t)), \tag{3}$$

where p_t is a readiness score trained to predict whether querying the frozen actor on the current observation will produce a successful action. The controller commits at the first candidate time where p_t exceeds a validation-selected threshold. If no candidate crosses the threshold, it commits at the final candidate time.

4.3 COUNTERFACTUAL SUPERVISION

Each training label is the realized terminal outcome obtained when the frozen actor is queried at a candidate time and its action is replayed under the prescribed timing protocol. We fit the controller with class-weighted binary cross-entropy. Feature normalization statistics are estimated on the training split only. The operating threshold is selected on a disjoint validation split using a utility that rewards success and penalizes observation time. The frozen test split is not used for training, normalization, or threshold selection.

4.4 RUNTIME ACCOUNTING

The frozen evaluation selects outcomes and execution times from counterfactual trajectories using precomputed visual features. Observation time is the capture time of the selected screenshot. It isolates observation-selection quality under a fixed actor and replay protocol; online implementation measurements are reported in Appendix A.4. We use *next-step completion time* for the interval from dispatch of the opening action until the environment returns from the next action. Appendix A.4 reports measurements with feature extraction on the execution path.

5 EXPERIMENTAL SETUP

Tasks and actor. We evaluate `click-collapsible` and `click-pie`, each with animated and matched no-delay conditions. MiniWoB supplies reproducible seeds and programmatic success signals (Shi et al., 2017). The frozen actor is Qwen2.5-VL-3B-Instruct. Candidate screenshots occur at 0, 0.1, 0.25, 0.5, 0.75, 1, 1.25, 1.5, and 2 seconds after the opening action. Training, validation, and frozen test seeds are disjoint.

Baselines. We compare the full visual controller with the validation-locked fixed wait and global stability. The primary 0.5-second fixed wait is selected on validation data.

Causal protocol. The causal matrix uses 50 unseen collapsible seeds and 20 unseen pie seeds, with matched no-delay controls. Early and late observations are collected immediately and at 1.5 seconds. Early and late execution delays are 0.25 and 1.5 seconds. Each frozen action is replayed in a fresh identically seeded episode.

Statistics. Success comparisons use paired per-seed outcomes. Latency differences between full visual and fixed wait use paired bootstrap 95% confidence intervals over the 80 frozen episodes. We report means for direct comparison with the fixed operating point and preserve all episode-level rows.

6 RESULTS

6.1 OBSERVATION MATURITY AND EXECUTION READINESS ARE SEPARABLE

On 50 unseen collapsible seeds, the early-observation conditions A and B both achieve 0/50 success. Delaying execution cannot repair an action selected from an immature observation. With a mature observation, early execution succeeds on 22/50 seeds, while late execution succeeds on 48/50. The same protocol on pie gives 0/20, 0/20, 0/20, and 18/20 (Table 2).

Table 2: Strict success under independent timing interventions.

Task	A	B	C	D
Collapsible	0/50	0/50	22/50	**48/50**
Pie	0/20	0/20	0/20	**18/20**

Table 3: Frozen evaluation on 80 unseen episodes. Lower latency is better.

Policy	Success (%)	Observation (s)	Actor query (s)	Completion (s)
Fixed wait, 0.5 s	83.75	0.500	0.886	1.448
Full visual	**83.75**	**0.132**	**0.873**	**1.066**

Removing animation eliminates the timing effect while preserving seed-specific grounding failures. On the 15 pie seeds solved in every no-delay cell, the animated matrix is 0/15, 0/15, 0/15, and 15/15. Together, both task families show that observation maturity and execution readiness are distinct factors.

6.2 Global stability fails in opposite directions

The static-pending interface remains pixel-identical until the target mounts. A global stability rule releases the actor on the empty plateau and succeeds on 0/17 seeds, whereas an oracle target-presence probe succeeds on 17/17. In the moving-ready interface, the target is actionable immediately while an unrelated element continues moving. Global stability waits a median 2.619 seconds, whereas the oracle decides at 64 milliseconds and succeeds on 16/17 seeds. These mechanism probes isolate why whole-screen motion is an unreliable proxy for task readiness.

Premature observation also creates a qualitative grounding failure. On an empty static canvas, the actor returns the same coordinate, $[294, 301]$, on three seeds instead of reporting that the target is absent. An immature screenshot can therefore induce a grounded-looking but unsupported action.

6.3 Visual readiness selects earlier observations at matched success

On 80 unseen frozen episodes, the validation-locked fixed wait and full visual controller both achieve 83.75% aggregate success. They also agree on every episode-level success outcome (67/80), so the latency comparison does not trade one set of failures for another. The controller reduces mean observation time from 0.500 to 0.132 seconds and mean next-step completion time from 1.448 to 1.066 seconds. The paired reductions are 0.368 seconds (95% CI $[0.336, 0.399]$) and 0.381 seconds (95% CI $[0.346, 0.416]$), respectively. Actor generation times are similar, 0.886 and 0.873 seconds, so the replay gain comes from earlier readiness decisions.

The per-condition comparisons have the same direction. On animated collapsible, full visual reduces next-step completion time from 1.409 to 1.130 seconds at equal 85% aggregate success. On animated pie, it reduces next-step completion time from 1.502 to 1.266 seconds at equal 90% aggregate success. The no-delay conditions retain equal success and show larger reductions because the controller commits immediately when no transition is present.

7 Discussion

The timing matrix identifies two distinct intervention points in the GUI-agent loop: acquiring evidence and delivering the resulting action. Correcting execution timing alone cannot recover evidence absent from the screenshot used to choose that action. The controlled interfaces further show why whole-screen stability cannot determine whether task-relevant evidence is available.

Counterfactual actor outcomes provide supervision for learning when to acquire that evidence. At a validation-selected operating point, instruction-conditioned visual states support earlier observations with identical episode-level success on the frozen test split. This connects timing diagnosis to an explicit observation-selection policy and motivates studying readiness in longer interaction sequences.

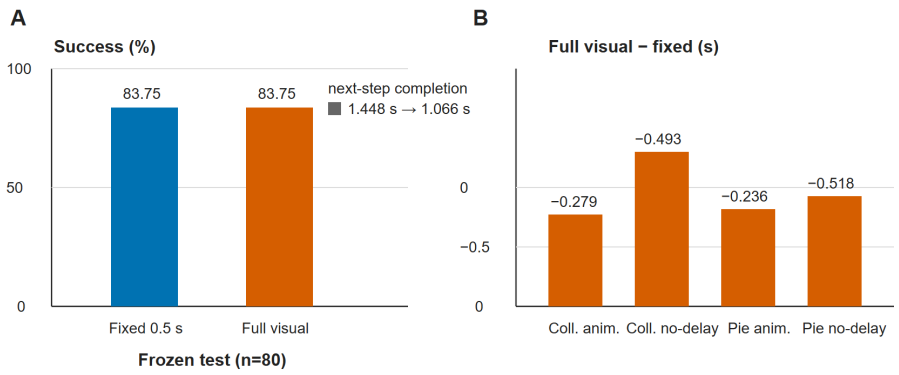

Figure 2: **Frozen visual-readiness evaluation.** The full visual policy preserves aggregate success while reducing counterfactual next-step completion time. Condition-wise differences are full visual minus fixed wait; all four conditions move in the same direction.

8 Conclusion

We introduced post-action observation timing as a distinct control problem for GUI agents. Counterfactual replay separated observation maturity from execution readiness, while controlled interfaces exposed the failure of global pixel stability. An instruction-conditioned visual-readiness controller preserved episode-level success while selecting substantially earlier observations in frozen counterfactual evaluation. Deciding when to look again is therefore a measurable component of GUI-agent control, distinct from both action prediction and action execution.

AI use statement

Generative AI tools assisted hypothesis exploration, experiment and software development, data analysis, literature search, figure drafting, and manuscript editing. The authors take full responsibility for the methods, results, claims, code, and text.

Reproducibility statement

The method, timing protocol, candidate observation schedule, baselines, strict success criterion, and statistical procedure are specified in Sections 3–5. Appendix A provides the exact candidate schedule, data splits, controller hyperparameters, bootstrap procedure, and full online policy curves.

References

Gilles Baechler, Srinivas Sunkara, Maria Wang, Fedir Zubach, Hassan Mansoor, Vincent Etter, Victor Carbune, Jason Lin, Jindong Chen, and Abhanshu Sharma. ScreenAI: A vision-language model for UI and infographics understanding. In *Proceedings of the Thirty-Third International Joint Conference on Artificial Intelligence*, pp. 3058–3068, 2024. doi: 10.24963/ijcai.2024/339.

Shuai Bai, Keqin Chen, Xuejing Liu, Jialin Wang, Wenbin Ge, et al. Qwen2.5-VL technical report. *arXiv preprint arXiv:2502.13923*, 2025.

Kanzhi Cheng, Qiushi Sun, Yougang Chu, Fangzhi Xu, Yantao Li, Jianbing Zhang, and Zhiyong Wu. Seeclick: Harnessing gui grounding for advanced visual gui agents. In *Proceedings of the 62nd Annual Meeting of the Association for Computational Linguistics*, pp. 9313–9332, 2024. doi: 10.18653/v1/2024.acl-long.505.

Xiang Deng, Yu Gu, Boyuan Zheng, Shijie Chen, Sam Stevens, Boshi Wang, Huan Sun, and Yu Su. Mind2web: Towards a generalist agent for the web. In *Advances in Neural Information Processing Systems 36*, pp. 28091–28114, 2023. doi: 10.52202/075280-1220.

Hongliang He, Wenlin Yao, Kaixin Ma, Wenhao Yu, Yong Dai, Hongming Zhang, Zhenzhong Lan, and Dong Yu. WebVoyager: Building an end-to-end web agent with large multimodal models. In *Proceedings of the 62nd Annual Meeting of the Association for Computational Linguistics*, pp. 6864–6890, 2024. doi: 10.18653/v1/2024.acl-long.371.

W. P. M. H. Heemels, K. H. Johansson, and P. Tabuada. An introduction to event-triggered and self-triggered control. In *2012 IEEE 51st IEEE Conference on Decision and Control*, pp. 3270–3285, 2012. doi: 10.1109/CDC.2012.6425820.

Wenyi Hong, Weihan Wang, Qingsong Lv, Jiazheng Xu, Wenmeng Yu, Junhui Ji, Yan Wang, Zihan Wang, Yuxiao Dong, Ming Ding, and Jie Tang. CogAgent: A visual language model for GUI agents. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, pp. 14281–14290, 2024. doi: 10.1109/CVPR52733.2024.01354.

Jing Yu Koh, Robert Lo, Lawrence Jang, Vikram Duvvur, Ming Lim, Po-Yu Huang, Graham Neubig, Shuyan Zhou, Russ Salakhutdinov, and Daniel Fried. Visualwebarena: Evaluating multimodal agents on realistic visual web tasks. In *Proceedings of the 62nd Annual Meeting of the Association for Computational Linguistics*, pp. 881–905, 2024. doi: 10.18653/v1/2024.acl-long.50.

Kevin Qinghong Lin, Linjie Li, Difei Gao, Zhengyuan Yang, Shiwei Wu, Zechen Bai, Stan Weixian Lei, Lijuan Wang, and Mike Zheng Shou. Showui: One vision-language-action model for gui visual agent. In *2025 IEEE/CVF Conference on Computer Vision and Pattern Recognition*, pp. 19498–19508, 2025. doi: 10.1109/CVPR52734.2025.01816.

Yadong Lu, Jianwei Yang, Yelong Shen, and Ahmed Hassan Awadallah. OmniParser for pure vision based GUI agent. *arXiv preprint arXiv:2408.00203*, 2024. doi: 10.48550/arXiv.2408.00203.

Tianlin Shi, Andrej Karpathy, Linxi Fan, Jonathan Hernandez, and Percy Liang. World of bits: An open-domain platform for web-based agents. In *Proceedings of the 34th International Conference on Machine Learning*, 2017.

Tianbao Xie, Danyang Zhang, Jixuan Chen, Xiaochuan Li, Siheng Zhao, Ruisheng Cao, Toh 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. In *Advances in Neural Information Processing Systems 37*, pp. 52040–52094, 2024. doi: 10.52202/079017-1650.

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. *arXiv preprint arXiv:2307.13854*, 2023. doi: 10.48550/arXiv.2307.13854.

A Reproducibility Details

A.1 Candidate schedule and counterfactual replay

Candidate screenshots are collected at $\{0, 0.1, 0.25, 0.5, 0.75, 1.0, 1.25, 1.5, 2.0\}$ seconds after the opening action. For the causal matrix, the early and late screenshots are collected immediately and at 1.5 seconds, respectively; frozen actions are replayed at 0.25 and 1.5 seconds. Every replay resets a fresh environment with the identical seed. No semantic actor query occurs during replay, and strict success requires reward 1 together with a terminal state.

The frozen comparison uses paired bootstrap confidence intervals with 100,000 resamples. Bootstrap seeds are fixed separately for success, observation time, and next-step completion time. Every failed episode remains in the denominator.

A.2 Controller architecture and optimization

The 6,146-dimensional full feature is $[h_t; h_{t-1}; h_t - h_{t-1}; \Delta t; \Delta_{\text{pix}}]$, where each frozen Qwen hidden state has 2,048 dimensions. Training-split means and scales standardize every feature. The controller is a two-layer MLP, $6146 \to 32 \to 1$, with a GELU hidden activation. We train for 100 full-batch epochs with class-weighted binary cross-entropy and AdamW (learning rate 10^{-3}, weight decay 10^{-3}, training seed 2026). Thresholds $\{0.05, 0.10, \ldots, 0.95\}$ are evaluated only on validation data using success minus 0.05 times mean observation time; the selected threshold is $\tau = 0.05$.

A.3 Exact data splits

The four groups below are disjoint:

- animated training: $\{301, 303, \ldots, 339\}$; no-delay training: $\{361, 363, \ldots, 379\}$;
- animated validation: $\{341, 343, \ldots, 359\}$; no-delay validation: $\{381, 383, 385, 387, 389\}$;
- frozen test in every task/condition: $\{101, 103, 107, 109, 113, 127, 131, 137, 139, 149, 151, 157, 163, 167, 173, 179, 181, 191, 193, 197\}$;
- online confirmation in every task/condition: $\{401, 403, 405, 407, 409, 411, 413, 415, 417, 419\}$.

A.4 Scope and online diagnostic

The study covers two click-based MiniWoB task families and one frozen Qwen2.5-VL actor. The controlled interfaces isolate timing mechanisms; transfer to longer workflows, richer action spaces, and additional actor families remains to be evaluated.

The main text reports the frozen comparison because it isolates observation-selection quality. For completeness, we also ran a 40-episode online confirmation with screenshot capture, feature extraction, controller inference, actor generation, action dispatch, and environment response on the critical path. A 0.5-second fixed wait reaches 95.0% success, while a 0.25-second wait reaches 65.0%. The locked visual policy also reaches 95.0% success, but sequential feature extraction adds 0.361 seconds and raises completion time from 1.560 to 1.732 seconds. These measurements identify representation cost as an implementation bottleneck without changing the frozen causal conclusion.

Table 4: Online confirmation on 40 new episodes. Completion includes controller, actor, and environment costs.

Policy	Success	Observation	Feature cost	Completion
Fixed wait, 0.25 s	26/40 (65.0%)	0.391	0	1.374
Fixed wait, 0.5 s	38/40 (95.0%)	0.635	0	**1.560**
Full visual, $\tau = 0.05$	38/40 (95.0%)	**0.632**	0.361	1.732
Global stability	38/40 (95.0%)	1.096	0	2.052

A.5 FULL NEW-SEED ONLINE CURVES

Table 5 reports the predeclared fixed-wait curve, global-stability baseline, and all predeclared visual thresholds. All policies have 40 episodes over the same four task/condition cells.

Table 5: Full new-seed online confirmation curves. Feature cost is included in completion time.

Policy	Success	Observation (s)	Feature (s)	Completion (s)
Fixed 0.00	20/40	0.299	0	1.298
Fixed 0.10	19/40	0.295	0	1.286
Fixed 0.25	26/40	0.391	0	1.374
Fixed 0.50	38/40	0.635	0	1.560
Fixed 0.75	38/40	0.881	0	1.805
Fixed 1.00	38/40	1.136	0	2.067
Fixed 1.50	38/40	1.627	0	2.546
Global stability	38/40	1.096	0	2.052
Visual 0.01	38/40	0.653	0.387	1.752
Visual 0.05	**38/40**	**0.632**	**0.361**	**1.732**
Visual 0.10	37/40	0.705	0.377	1.810
Visual 0.25	38/40	0.686	0.377	1.803
Visual 0.50	38/40	0.808	0.425	1.898
Visual 0.75	38/40	0.890	0.465	2.011