BEYOND NEXT-SCREEN PREDICTION: THE DECISION VALUE OF GUI WORLD MODELS

Anonymous authors
Paper under double-blind review

ABSTRACT

GUI world models are increasingly used to improve action selection by predicting the consequences of candidate actions, but it is unclear how much of the resulting gain comes from prediction itself rather than from the evaluator that interprets those predictions. We study this attribution problem through controlled interventions on GUI action selection. With candidates and predictions held fixed, lookahead gains can change with how the prediction is presented to the evaluator, so end-to-end gains cannot be attributed to prediction alone. We then introduce Direct Action Ranking (DAR), which learns from reference actions to score candidates without generating future screens, and compare it with a prediction-augmented variant under matched supervision. Predicted successors add no detectable ranking signal once evaluation is learned directly. Without retraining, DAR improves AndroidWorld task success by 5.8–16.4 points across three proposers, outperforms world-model lookahead, and transfers zero-shot to unseen apps, where every prediction-conditioned configuration falls below proposer confidence. These results suggest that GUI world models should be evaluated by their decision value, the incremental action-selection value of their predictions beyond a strong direct evaluator, rather than by prediction fidelity or end-to-end lookahead gain alone.

1 INTRODUCTION

GUI agents increasingly rely on learned models of interface dynamics to reason before acting. Rather than committing immediately to the proposer's highest-ranked action, a lookahead agent can predict how the interface will change under several candidate actions and use those predicted outcomes to guide its choice. This idea has become a prominent direction in GUI agents, spanning textual outcome prediction, graphical generation, and renderable next-state simulation (Gu et al., 2025; Chae et al., 2025a; Zheng et al., 2026). The appeal is straightforward: anticipating action consequences can prevent mistakes before execution.

A lookahead pipeline has two components. A world model generates possible future observations; an evaluator turns them into a ranking over candidate actions. End-to-end success depends on both the information in the prediction and the evaluator's ability to use it. The same prediction can support a better choice under one evaluator and a worse choice under another. A gain from "world-model lookahead" is therefore a gain of the prediction–evaluator system, and it does not reveal how much came from the predicted observation.

This distinction changes what should be measured. Prediction fidelity measures whether a predicted screen resembles the true future interface; end-to-end lookahead gain measures the whole pipeline. Neither measures what the prediction contributes to the choice. We call this *decision value of GUI world models*: the improvement in action selection attributable to predicted observations beyond direct evaluation of the same candidates. Decision value is defined by intervention, not by fidelity. A visually accurate prediction can be decision-irrelevant when the current screen and action already determine the right choice, while a coarse prediction can still be useful if it exposes a consequence that changes which candidate is best. Measuring it therefore requires comparing direct and prediction-conditioned evaluation on the same candidates under matched supervision (Fig. 1). We study this question for one-step prediction used in inference-time candidate selection.

We investigate this quantity by progressively separating prediction from evaluation. First, on AndroidControl trajectories (Li et al., 2024), we keep candidate actions and cached world-model predictions

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.

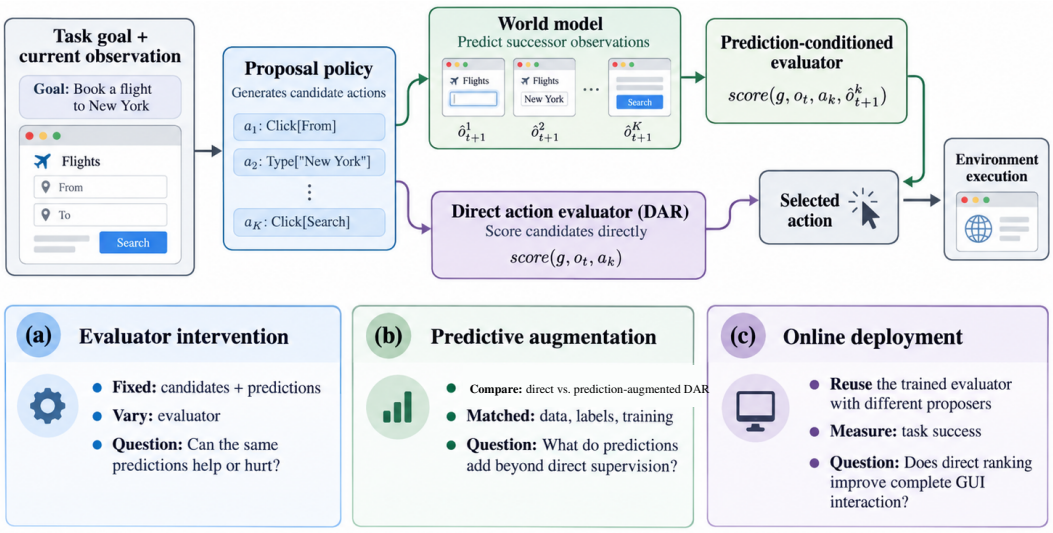


Figure 1: **Action selection and controlled comparisons.** Lookahead and DAR use alternative paths to score the same candidate actions. We vary the evaluator with predictions fixed, compare DAR with prediction-augmented DAR under matched training, and evaluate online task completion. Interface thumbnails are schematic.

fixed and change the evaluator. Identical predictions gain changes with how the prediction is presented to the evaluator, establishing evaluator choice as a first-order component of lookahead performance. We then introduce *Direct Action Ranking* (DAR), which learns to rank candidate actions directly from the goal and current interface using reference-action supervision, without generating a future screen. Under matched training, adding predicted successors to DAR changes reference-match AUROC by less than 0.001 across two candidate pools, isolating the incremental offline ranking signal supplied by those predictions. We also compare with proposer SFT trained on the same reference actions.

We then test DAR in complete interaction. Applied without retraining, DAR improves AndroidWorld task success by 5.8–16.4 points across three proposers, outperforms world-model lookahead. On 14 unseen Korean-language apps, DAR ranks candidates above all lookahead configurations, each of which falls below proposer confidence. Together, these results support a simple principle: *a GUI world model used for action selection should be credited for the improvement its predictions add beyond a strong direct evaluator, not for the gain of the whole lookahead pipeline.*

Our contributions.

- **A new evaluation target for GUI world models.** We formalize *decision value*, the improvement in action selection that predicted observations add beyond direct evaluation of the same candidates, and introduce a protocol that measures it by separating prediction from evaluation.
- **Predictions add little once evaluation is learned.** Under matched supervision, predicted screens add no detectable ranking signal to a direct evaluator; with predictions fixed, lookahead gains change with how the prediction is evaluated.
- **We introduce DAR, a world-model-free selector that outperforms world-model lookahead.** Without generating future screens or retraining, DAR improves task success across proposers and world models, and transfers to unseen apps and interface.

2 Background and Related Work

GUI agents and grounding. SeeClick and UGround learn visual grounding (Cheng et al., 2024; Gou et al., 2025); SeeAct coordinates planning and grounding (Zheng et al., 2024). WebVoyager and UI-TARS pursue end-to-end multimodal interaction (He et al., 2024; Qin et al., 2025). These

agents must both propose useful actions and identify their targets. We study the subsequent selection problem, keeping the proposer fixed while comparing ways to choose among its candidates.

World models for GUI agents. GUI lookahead uses textual predictions in WMA and WebDreamer (Chae et al., 2025a; Gu et al., 2025), separate graphical and textual generation in ViMo (Luo et al., 2026), and renderable code in Code2World and gWorld (Zheng et al., 2026; Koh et al., 2026). WebEvolver and WebWorld also use simulation for policy training and inference-time search (Fang et al., 2025; Xiao et al., 2026). Xu et al. (2026) examine feedback representation and confidence in mobile guidance. End-to-end gains evaluate the complete system; our fixed-prediction and matched-training comparisons separate the effects of changing an evaluator from those of adding predicted observations.

Learned action evaluation. Solution verifiers (Cobbe et al., 2021) and process supervision (Lightman et al., 2024) learn to assess reasoning. Web-Shepherd learns step-level web rewards (Chae et al., 2025b); WebArbiter generates reasoning-based preference verdicts (Zhang et al., 2026). DAR learns from reference-action matches to score GUI candidates directly. Training a prediction-augmented DAR model on the same labels then tests whether predicted successors improve this learned comparison.

3 Action Selection and Study Design

3.1 Proposal, prediction, and evaluation

At step t, a proposal policy π receives a goal g and observation o_t (screenshot and available interface structure) and produces candidates $\mathcal{A}_t = \{a_{t,1}, \ldots, a_{t,K}\}$ with confidence scores $c_{t,k}$. Policy Top-1 executes the highest-confidence candidate; every other selector reranks this fixed set. A world model predicts $\hat{o}^{(k)}_{t+1} = W_\phi(o_t, a_{t,k})$, and an evaluator scores each candidate with or without that prediction:

$$e^{\text{pred}}_{t,k} = E_\psi(g, o_t, a_{t,k}, \hat{o}^{(k)}_{t+1}), \qquad e^{\text{dir}}_{t,k} = E_\theta(g, o_t, a_{t,k}). \tag{1}$$

The selector executes $a_{t,\hat{k}_t}$ with $\hat{k}_t = \arg\max_k e_{t,k}$. We use three evaluators. *Lookahead* uses a prompted E_ψ and falls back to Policy Top-1. DAR is a trained E_θ. prediction-augmented DAR is a trained E_ψ that matches DAR in candidates, labels, backbone, and objective, and additionally receives $\hat{o}^{(k)}_{t+1}$.

3.2 Reference labels and metrics

For a recorded reference action a^{ref}_t, a domain-specific matcher defines

$$y_{t,k} = \mathbf{1}[a_{t,k} \simeq a^{\text{ref}}_t], \tag{2}$$

and several candidates can match. Each metric answers a different question. *Reference Coverage@K*: does any candidate match? *Reference-Match AUROC*: pooled over decisions, do matching candidates score higher than non-matching ones? *Reference-Match Accuracy*: does the selected action match, $y_{t,\hat{k}_t} = 1$? *Task Success Rate*: does the agent complete the task, as verified by the benchmark? Together they trace a gain from available candidates to ranking quality to task completion.

3.3 Decision value and comparisons

Decision value. For a metric M, the decision value of W_ϕ relative to a direct evaluator E_θ is

$$\text{DV}_M(W_\phi \mid E_\theta) = M(E_\psi; W_\phi) - M(E_\theta), \tag{3}$$

where E_ψ is trained identically to E_θ and differs only in receiving $\hat{o}^{(k)}_{t+1}$. DV is defined relative to a direct evaluator, not for the world model alone: the same predictions may add value to a weak direct evaluator and none to a strong one.

Comparisons. We use three comparisons (Figure 1). *Evaluator intervention* (Section 4) holds candidates and predictions fixed and varies a prompted E_ψ, testing whether the evaluator alone

Table 1: **Diagnostic signals for useful overrides.** Override-Benefit AUROC on each evaluator's informative subset. Each evaluator has a selected-minus-Top-1 contrast column and a per-candidate scalar column; set- and step-level scores span both columns. The first four rows require the reference successor; the remaining signals are available before execution. Darker/lighter lavender marks the highest/second-highest scores per evaluator and availability block. Full statistics: Appendix A.3.

	Sonnet 4.6 (330)		Opus 4.8 (366)	
Signal	Contrast	Scalar	Contrast	Scalar
Visual agreement (SigLIP)	**0.725**	0.608	**0.746**	0.590
DOM agreement	**0.711**	0.530	**0.671**	0.547
Element grounding	0.552	0.552	0.548	0.546
Text embedding	0.517	0.528	0.495	0.514
WM likelihood	0.562	**0.597**	**0.541**	**0.587**
WM confidence	**0.579**	0.539	0.533	0.492
Likelihood spread	0.503		0.502	
Confidence spread	0.438		0.475	
Policy confidence	0.428		0.430	
Negated selector score	0.430		0.458	

changes the effect of lookahead. *Prediction augmentation* (Section 5) estimates DV with the matched DAR/prediction-augmented DAR pair, using Reference-Match AUROC as M. *Online deployment* (Section 6) compares Policy Top-1, Lookahead, and DAR by task success; trajectories may diverge as different actions are selected. Lookahead does not meet the matching condition, so this comparison does not measure DV; it shows how a prompted prediction pipeline compares with a trained direct evaluator. Section 5.4 also compares DAR with proposer SFT trained on the same reference actions. All comparisons study one-step prediction for inference-time candidate selection.

Data. AndroidControl (Li et al., 2024) provides the offline trajectories: a 3,000-step episode-level evaluation split and separate selector and SFT training pools. AndroidWorld (Rawles et al., 2025) evaluates complete tasks.

4 Separating Prediction from Action Evaluation

4.1 Fixed predictions, different decisions

We first fix candidates and predictions and vary only their evaluation, using 3,000 AndroidControl evaluation steps, Qwen2.5-VL-7B candidates (Bai et al., 2025b) ($K \leq 3$), and cached Code2World-8B predictions (Zheng et al., 2026). Selection alone has headroom: 56.1% of candidate sets contain a reference match, whereas Policy Top-1 matches on 43.8% (Table 3(a)).

With predictions fixed, lookahead raises Reference-Match Accuracy over Policy Top-1 by 1.7 points with Claude Sonnet 4.6 and 2.5 with Claude Opus 4.8 (both 95% episode-clustered intervals exclude zero; Appendix A.1). The gain also depends on how predictions reach the evaluator: re-encoding the same successors as an accessibility tree or plain text lowers the Opus 4.8 point estimate from $+2.5$ to $+1.2$ (Appendix D). End-to-end lookahead performance thus characterizes a prediction–evaluator pair, not the prediction alone; this comparison does not measure decision value, which requires matched evaluators (Section 3.3). In Figure 2, Top-1 and lookahead choose different clicks, while the direct evaluator of Section 5 selects the reference scroll (more in Appendix E).

4.2 Which signals identify useful overrides?

We next ask whether any signal separates useful overrides from harmful ones. For $\Delta_t = y_{t,\hat{k}_t} - y_{t,k_t^{\text{top}}}$, *Override-Benefit AUROC* distinguishes overrides that recover a reference match ($\Delta_t = +1$) from those that lose one ($\Delta_t = -1$) on each evaluator's decisions with $\Delta_t \neq 0$ (Table 1).

The strongest signals all require the reference successor, unavailable at decision time: visual (SigLIP; Zhai et al., 2023) and DOM agreement with it separate useful overrides best, and contrasting the selected candidate with Top-1 is more informative than scoring it alone, for all evaluators. Pre-

execution signals are weaker and inconsistent: likelihood spread stays at chance, and confidence spread falls below 0.5 for both evaluators. No pre-execution signal is consistently informative, which motivates learning the comparison directly from reference actions (Section 5); Appendix D also varies lookahead depth.

Takeaway 1: Lookahead gains belong to the prediction–evaluator pair.
With predictions fixed, the gain changes with how the prediction is presented to the evaluator.

5 Learning to Select Actions Directly

5.1 DAR: reference-supervised action ranking

DAR instantiates the direct evaluator E_θ in Eq. 1: it is both the control for measuring decision value and a practical selector. It scores each candidate from the goal, current interface, and action description, using reference actions only to construct training labels. The mobile input includes the screenshot and available accessibility elements with bounding boxes. We adapt Qwen3-VL-8B-Instruct (Bai et al., 2025a) with frozen base weights, LoRA adapters (Hu et al., 2022), and a linear scoring head on the final-token representation. Each candidate is scored independently; candidates from the same observation are grouped when computing the ranking loss.

The listwise term (Cao et al., 2007) trains the within-set ordering needed for selection, while the pointwise term supervises whether each candidate matches the reference. For $\mathcal{P}_t = \{k : y_{t,k} = 1\} \neq \varnothing$,

$$\mathcal{L}_{\text{DAR}} = \mathcal{L}_{\text{rank}} + \beta \mathcal{L}_{\text{match}},$$
$$\mathcal{L}_{\text{rank}} = -\frac{1}{|\mathcal{P}_t|} \sum_{k \in \mathcal{P}_t} \log \frac{\exp(e_{t,k})}{\sum_j \exp(e_{t,j})}, \quad \mathcal{L}_{\text{match}} = \frac{1}{K} \sum_k \text{BCE}(\sigma(e_{t,k}), y_{t,k}). \tag{4}$$

For all-negative groups, $\mathcal{L}_{\text{rank}} = 0$ and pointwise supervision remains. The listwise term targets within-state competition, while the pointwise term anchors reference compatibility and still trains all-negative groups; Section 6.4 tests the online effect of this combination. We average over intact candidate groups, use $\beta = 0.5$, train for two epochs, and select the highest-scoring candidate. Appendix B.1 gives filtering, input construction, domain-specific matching, and optimization settings.

5.2 What direct supervision achieves

Reference supervision improves candidate discrimination well beyond proposer confidence (Table 2): on the shared AndroidControl test pool, DAR trained on either candidate source raises AUROC by 0.17–0.18 over policy confidence, and a separate Mind2Web-trained evaluator (Deng et al., 2023) also exceeds confidence on all three splits, including the cross-website and cross-domain splits.

Table 2: **Offline reference-match ranking (AUROC).** Both AndroidControl columns use one Qwen2.5-VL-7B test-candidate pool; column labels identify DAR's training-candidate source. Mind2Web uses a separate web evaluator. Darker/lighter lavender marks the highest and second-highest scores within each column.

	AndroidControl		Mind2Web		
Evaluator / score	Qwen2.5 train	External train	Cross-Task	Cross-Website	Cross-Domain
Policy confidence	**0.775**	**0.775**	**0.870**	**0.874**	**0.867**
DAR	**0.957**	**0.946**	**0.949**	**0.939**	**0.943**
DAR − confidence	+0.182	+0.171	+0.079	+0.065	+0.076

Better discrimination also produces better choices: on the original $K \leq 3$ pool, DAR raises selection accuracy from 43.8% to 50.1% (Table 3(a)). A second proposer decode raises both reference coverage and DAR accuracy but widens the gap between them (Policy Top-1 uses the first decode throughout): more matching actions are available, yet the evaluator still misses some. Since a matching action must be both proposed and selected, proposal coverage and candidate evaluation are complementary.

Table 3: **Candidate budget, selector supervision, and an SFT baseline.** 3,000 AndroidControl decisions. (a) Candidate budget for a fixed proposer. (b) Nested DAR training sets; conditional match accuracy is computed on decisions whose candidate set contains a reference match. (c) Qwen3-VL-32B proposer SFT, evaluated on the same evaluation fold; $|\mathcal{A}|$ is the deduplicated candidate count. The two-decode setting has mean candidate count 3.64. Full settings: Appendices B.2–B.4.

(a) Candidate budget

Candidates	Cov.%	DAR%	Δ pp
$K = 1$	43.8	43.8	0.0
$K \leq 2$	52.8	48.9	+5.1
$K \leq 3$	**56.1**	**50.1**	+6.3
Two decodes	**60.8**	**52.5**	+8.6

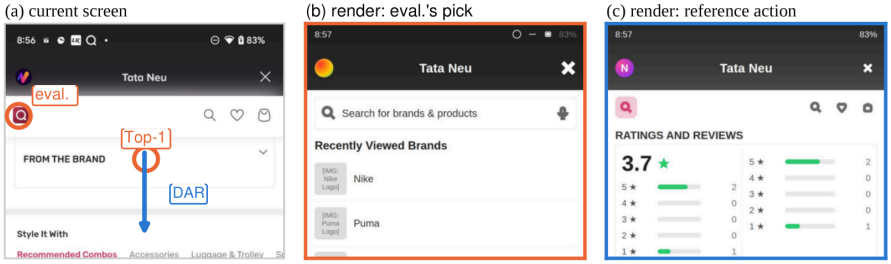
(b) DAR training examples

Examples	AUROC	Cond. match acc.
2,000	.900	.849
5,000	.932	.890
10,000	**.947**	**.901**
26,365	**.957**	**.911**

(c) Proposer-SFT baseline

Qwen3-VL-32B proposer	Labels	Greedy	Mean $\lvert\mathcal{A}\rvert$	Pass@3
Zero-shot	0	0.6137	2.7259	0.7430
SFT	2,000	0.4797	1.6780	0.6187
SFT	26,365	0.6543	1.3367	0.7347

goal: To check the price, I would like to view similar products like Nike Men's RUN SWIFT 2 Black Running Shoes. Use the ...



candidate action	policy conf.	eval. score	DAR score	reference?
click (546,524) [Top-1]	0.8	-0.70	-5.16	no
scroll down [DAR]	0.6	-0.75	+0.46	yes
click (54,357) [eval.]	0.5	+0.85	-3.18	no

Figure 2: **Different choices on a recorded AndroidControl step.** To find product prices, Top-1 and lookahead choose different clicks; DAR selects the reference scroll. Panels show the current screen and predictions for the lookahead and reference actions. The table gives candidate scores; "eval." denotes the lookahead evaluator. Case selection: Appendix E.

More supervision also improves the learned comparison itself: expanding a nested training pool, with the same two-epoch schedule, raises AUROC from 0.900 to 0.957 and improves selection conditional on a matching candidate (Table 3(b); settings in Appendices B.3–B.4).

5.3 What do predicted observations add?

We now measure decision value directly: do predictions improve an evaluator already trained on reference actions? On an external-proposer pool and a Qwen2.5-VL-7B-proposed pool, we train DAR and prediction-augmented DAR independently with the same backbone, candidates, labels, objective, and optimization; the predicted successor is the only added input. Predictions change AUROC by -0.0003 and $+0.0005$, with 95% intervals within ± 0.004 (Appendix B.1), whereas direct supervision raises AUROC by 0.17–0.18 over policy confidence: once evaluation is learned directly, predicted screens add no detectable ranking signal. Section 4.1 showed that lookahead gains depend on how predictions are evaluated; for a learned evaluator, the predictions themselves add

little. Because pooled AUROC and online success can diverge (Section 6.4), this bounds the offline ranking signal of predictions but does not by itself rule out online decision value; Section 6 tests online choices.

Takeaway 2: Predictions add no detectable ranking signal once evaluation is learned. Under matched supervision, predicted successors change reference-match AUROC by at most 0.004 (95% CI), against a 0.17–0.18 gain from direct supervision.

5.4 Comparison with proposer SFT

A natural alternative spends the same reference actions on the proposer. Full-budget Qwen3-VL-32B SFT raises greedy single-step matching from 61.4% to 65.4% but shrinks the mean deduplicated candidate set from 2.73 to 1.34; DAR still adds 2.7 points on the SFT candidates, so proposer training and candidate evaluation are complementary. With 2,000 labels, SFT falls below its zero-shot baseline, whereas DAR trained on 2,000 labels improves its own candidate pool by 3.8 points; the settings differ in proposer and candidate pool, so this contrast is only qualitative (Appendix B.2).

6 From Action Ranking to Task Success

6.1 Online selection on AndroidWorld

Offline matching evaluates single decisions; an online agent must complete a sequence of them. We train a mobile DAR model once, on Qwen2.5-VL-7B-proposed AndroidControl candidates, and apply it without retraining to three proposers on 116 AndroidWorld tasks (Rawles et al., 2025) (three seeds, $K \leq 3$). Lookahead scores Code2World-8B predictions with an independent evaluator (External) or the proposer itself (Self) and retains the confidence-based fallback; task and candidate-generation protocols are shared within each proposer (Appendix C.1). This compares a trained direct evaluator with prompted lookahead and does not measure decision value (Section 3.3).

Table 4: **AndroidWorld application-family results (%).** 116 tasks, three seeds. All: mean ± s.e.m. External/Self use Code2World. Counts appear below family names; shading ranks selectors per proposer and column. Full paired statistics: Appendix C.1.

Proposer	Selector	SMS 6	Tasks 6	Sport 6	Exp. 9	Recipe 13	Cal. 17	Markor 14	Sys. 13	Other 32	**All** 116
Maverick	Top-1	11	22	**11**	7	**28**	4	21	13	27	18.1 ± 2.0
	External	**22**	**28**	**22**	**15**	**41**	4	21	**44**	28	**25.3 ± 1.9**
	Self	11	22	**11**	7	26	**8**	**24**	**44**	**30**	23.0 ± 2.4
	DAR	**33**	**39**	6	**19**	**41**	**12**	**33**	**64**	**42**	**34.5 ± 2.5**
Qwen3 8B	Top-1	44	22	0	**48**	31	8	**19**	15	29	23.9 ± 1.6
	External	**67**	17	0	37	33	**10**	17	**26**	37	27.6 ± 4.1
	Self	50	**28**	**11**	**41**	**36**	**12**	14	**26**	**38**	**28.4 ± 2.5**
	DAR	**72**	**39**	**22**	**41**	**38**	**12**	**29**	**31**	**45**	**35.3 ± 2.0**
Qwen3 235B	Top-1	**89**	**39**	**22**	**52**	**44**	**20**	29	72	35	40.8 ± 2.4
	External	**83**	28	**28**	**52**	**44**	12	**33**	**77**	**43**	**42.2 ± 1.7**
	Self	**89**	**33**	17	**44**	**44**	14	31	**74**	41	40.8 ± 3.5
	DAR	**89**	**39**	**28**	**44**	**49**	**22**	**36**	**77**	**49**	**46.6 ± 1.5**

DAR has the highest mean task success for every proposer (Table 4), improving over Top-1 by 16.4, 11.5, and 5.8 points and over both lookahead configurations by 4.3–11.5 points; all nine paired contrasts have $p < 0.05$ (exact McNemar, uncorrected; Table 14), the smallest being +4.3 over External with Qwen3-VL-235B ($p = 0.049$). Gains hold under a candidate-generation shift: trained only on Qwen2.5-VL-7B proposals, DAR ranks actions from Llama-4 Maverick and two Qwen3-VL models without next-screen generation. As Top-1 success rises from 18.1% to 40.8%, DAR's gain falls from 16.4 to 5.8 points, consistent with less selection headroom for stronger proposers.

6.2 CHANGING THE WORLD MODEL

We replace Code2World-8B with gWorld-8B (Koh et al., 2026), keeping proposer and selector protocols fixed; each world model uses its native input format (Code2World: goal conditioning and an action annotation; gWorld: a raw screenshot without a goal slot).

Table 5: **AndroidWorld with gWorld-8B (%).** Same 116 tasks and families as Table 4; entries are three-seed means. Δ is the difference between the displayed three-seed mean success rates (gWorld–Code2World). Top-1/DAR do not use a world model.

Proposer	Selector	SMS	Tasks	Sport	Exp.	Recipe	Cal.	Markor	Sys.	Other	**All**	Δ
Maverick	Top-1	11	**22**	**11**	7	28	4	**21**	13	27	18.1±2.0	—
	External	**17**	17	**17**	**22**	**38**	**18**	**21**	**31**	**31**	**25.9**	+0.6
	Self	**17**	17	**17**	**33**	**38**	6	**21**	**31**	**31**	25.0	+2.0
	DAR	**33**	**39**	6	**19**	**41**	**12**	**33**	**64**	**42**	**34.5±2.5**	—
Qwen3 8B	Top-1	44	22	0	**48**	31	8	19	15	29	23.9±1.6	—
	External	61	**33**	**22**	**41**	**36**	**12**	**24**	**26**	**41**	**31.9±0.5**	+4.3
	Self	**67**	**33**	**17**	**41**	**38**	**10**	**24**	18	34	29.3±3.6	+0.9
	DAR	**72**	**39**	**22**	**41**	**38**	**12**	**29**	**31**	**45**	**35.3±2.0**	—
Qwen3 235B	Top-1	**89**	**39**	**22**	**52**	44	**20**	29	72	35	40.8±2.4	—
	External	**94**	**33**	**28**	**44**	**46**	12	**31**	72	**42**	**41.7±3.6**	−0.5
	Self	83	28	11	**52**	44	16	**31**	**74**	40	40.5±1.7	−0.3
	DAR	**89**	**39**	**28**	**44**	**49**	**22**	**36**	**77**	**49**	**46.6±1.5**	—

The swap changes lookahead success by -0.5 to $+4.3$ points depending on the proposer, while DAR, which uses no world model, keeps the highest mean for every proposer (Table 5): the predictor changes the evidence lookahead receives, whereas direct evaluation is a fixed reference point.

6.3 TRANSFER TO KOREAN APPS AND WEB TASKS

New applications and language. On MWMBench-KApps (Koh et al., 2026), we apply the same mobile DAR model without retraining to 14 unseen Korean-language apps, changing the application inventory and interface language; two of three proposers were also unseen in training. DAR has the highest AUROC for every proposer (Table 6; Appendix C.2), whereas every prediction-conditioned configuration ranks candidates worse than the proposer's own confidence under this shift.

Table 6: **KApps transfer.** Reference-Match AUROC (mean ± SD over three seeds) across 14 Korean apps. The same DAR evaluator is used without retraining across proposers; shading ranks each column.

	Candidate proposer		
Evaluator / score	Qwen2.5-7B	Qwen3-8B	Qwen3-235B
Policy confidence	**.739±.008**	**.836±.029**	**.791±.004**
Code2World–External	.629±.011	.667±.004	.633±.009
Code2World–Self	.580±.029	.631±.016	.606±.016
gWorld–External	.624±.010	.687±.040	.675±.026
gWorld–Self	.578±.043	.616±.016	.622±.024
DAR	**.907±.033**	**.911±.028**	**.877±.001**
DAR − confidence	+.169	+.075	+.085

Web interaction. MiniWoB changes both the interaction substrate and the task distribution, so we deploy a separate Mind2Web-trained DAR evaluator on 98 MiniWoB tasks with three seeds (Deng et al., 2023; Liu et al., 2018). The ordering now depends on the proposer (Table 7): lookahead (External) outperforms DAR with Qwen3-VL-8B (38.9% vs. 32.0%), whereas DAR outperforms it with Qwen3-VL-235B (44.4% vs. 40.8%), so direct evaluation does not win everywhere. As the two selectors differ in both evaluator and input, this does not isolate the prediction's contribution.

Table 7: **MiniWoB task-family results (%).** 98 tasks × 3 seeds. All: mean ± s.e.m.; parentheses: paired DAR–Top-1 gain ± s.e.m. Shading ranks each proposer and column.

Proposer	Selector	Click	Email	Text	Date	Reason	Form	Other	**All**
Qwen3 8B	Top-1	39	6	26	**85**	20	10	38	28.5±0.7
	External	**53**	**15**	**48**	0	**38**	**32**	**44**	**38.9±0.8**
	Self	**49**	6	**48**	0	32	**22**	41	**34.5±0.8**
	DAR	48	**8**	**27**	**18**	**35**	18	**42**	32.0±0.7 (+3.6±1.0)
Qwen3 235B	Top-1	51	**9**	**45**	**90**	**52**	32	**48**	**43.5±0.8**
	External	53	5	**51**	0	48	**40**	46	40.8±0.6
	Self	**54**	4	44	0	42	**36**	**48**	39.1±1.0
	DAR	**55**	**11**	42	**62**	**55**	34	**52**	**44.4±0.8 (+0.9±0.8)**

6.4 When offline ranking and task success diverge

Removing the pointwise loss lowers AUROC from 0.957 to 0.934, but reduces conditional match accuracy from 0.911 to 0.890 and task success by 6.0 points, with worse calibration (Table 8(b); Appendix C.4). Backbones show the same pattern on a smaller scale: GLM has slightly higher AUROC than MiMo but lower mean task success, and both improve on Top-1 (Table 8(a)). Pooled AUROC scores orderings across candidates, whereas execution depends on each set's winner, so similar AUROC can hide different action choices; the matched comparison of Section 5.3 thus bounds only the offline ranking signal of predictions.

Table 8: **Evaluator and objective sensitivity.** AndroidControl offline; AndroidWorld online (116 tasks, Qwen3-VL-8B, three-seed mean ± s.e.m.). Conditional match accuracy is computed on decisions whose candidate set contains a reference match; full statistics are in Appendix C.4.

(a) Configuration, ranking, and task success

Configuration	Evaluator backbone	β	AUROC ↑	Online success (%) ↑
Full DAR	Qwen3-VL-8B	0.5	**.957**	**35.3 ± 2.0**
Backbone variant	MiMo-VL-7B-RL	0.5	.879	**30.5 ± 1.4**
Backbone variant	GLM-4.1V-9B-Thinking	0.5	.884	28.7 ± 2.3
No pointwise loss	Qwen3-VL-8B	0	**.934**	29.3 ± 2.0

(b) Conditional selection accuracy and score quality

Objective	Conditional match acc. ↑	ECE ↓	Brier ↓	NLL ↓
Full DAR ($\beta = 0.5$)	**.911**	**.022**	**.099**	**.324**
No pointwise loss ($\beta = 0$)	**.890**	**.101**	**.131**	**.408**

Takeaway 3: A trained direct evaluator sets a high bar for lookahead.
Without retraining, DAR has the highest mean success on AndroidWorld and the highest AUROC on KApps for every proposer; DAR also improves Top-1 on both MiniWoB proposers.

7 Conclusion

GUI lookahead couples prediction with evaluation, so end-to-end gains cannot be attributed to prediction alone. We propose crediting a GUI world model with its decision value: what its predictions add to action selection beyond a direct evaluator of the same candidates. Three findings support this view. First, with predictions fixed, lookahead gains change with how predictions are evaluated and presented, so the evaluator is a first-order component of any lookahead result. Second, under matched supervision, predicted successors add no detectable ranking signal once evaluation is learned directly. Third, without retraining or generating future screens, DAR outperforms lookahead on AndroidWorld with either world model and on unseen Korean-language apps, although lookahead still wins on MiniWoB for the weaker proposer. Our study covers one-step prediction for inference-time selection with reference-action labels; world models used for training, longer horizons, or consequence-aware objectives may show larger decision value. Future GUI world models should report decision value alongside prediction fidelity and end-to-end gain.

AI USE STATEMENT

Generative AI tools were used to provide feedback on research framing, methodology, experimental comparisons, and result interpretation; to assist with manuscript drafting, editing, organization, and reviewer-style critique; to revise figure and table presentation; to check internal consistency of reported numbers and cross-references; and to prepare and debug the LaTeX source. All AI-assisted material included in the submission was reviewed and edited by the authors. The authors take responsibility for the final content of this work, including text, claims, analyses, and artifacts produced with the aid of generative AI. Datasets used to train DAR are also partially generated by AI models.

REPRODUCIBILITY STATEMENT

The paper and appendix document the candidate construction and reference-matching conventions, model and optimization settings, evaluation splits, seeds, statistical procedures, and online evaluation protocols used in the principal experiments. Appendix A gives the fixed-prediction evaluation details, Appendix B reports DAR implementation and sensitivity settings, Appendix C specifies online and transfer evaluation procedures and paired tests, and Appendix D reports the additional prediction diagnostics.

REFERENCES

Anthropic. Introducing Claude Opus 4.8, 2026a. URL https://www.anthropic.com/news/claude-opus-4-8.

Anthropic. Introducing Claude Sonnet 4.6, 2026b. URL https://www.anthropic.com/news/claude-sonnet-4-6.

Shuai Bai, Yuxuan Cai, Ruizhe Chen, Keqin Chen, Xionghui Chen, Zesen Cheng, Lianghao Deng, Wei Ding, Chang Gao, Chunjiang Ge, et al. Qwen3-VL technical report. *arXiv preprint arXiv:2511.21631*, 2025a. URL https://arxiv.org/abs/2511.21631.

Shuai Bai, Keqin Chen, Xuejing Liu, Jialin Wang, Wenbin Ge, Sibo Song, Kai Dang, Peng Wang, Shijie Wang, Jun Tang, et al. Qwen2.5-VL technical report. *arXiv preprint arXiv:2502.13923*, 2025b. URL https://arxiv.org/abs/2502.13923.

Yoav Benjamini and Yosef Hochberg. Controlling the false discovery rate: A practical and powerful approach to multiple testing. *Journal of the Royal Statistical Society: Series B (Methodological)*, 57(1):289–300, 1995. doi: 10.1111/j.2517-6161.1995.tb02031.x.

Zhe Cao, Tao Qin, Tie-Yan Liu, Ming-Feng Tsai, and Hang Li. Learning to rank: From pairwise approach to listwise approach. In *Proceedings of the 24th International Conference on Machine Learning*, pp. 129–136, 2007. doi: 10.1145/1273496.1273513.

Hyungjoo Chae, Namyoung Kim, Kai Tzu-iunn Ong, Minju Gwak, Gwanwoo Song, Jihoon Kim, Sunghwan Kim, Dongha Lee, and Jinyoung Yeo. Web agents with world models: Learning and leveraging environment dynamics in web navigation. In *International Conference on Learning Representations*, 2025a. URL https://openreview.net/forum?id=moWiYJuSGF.

Hyungjoo Chae, Seonghwan Kim, Junhee Cho, Seungone Kim, Seungjun Moon, Gyeom Hwangbo, Dongha Lim, Minjin Kim, Yeonjun Hwang, Minju Gwak, Dongwook Choi, Minseok Kang, Gwanhoon Im, ByeongUng Cho, Hyojun Kim, Jun Han, Taeyoon Kwon, Minju Kim, Beong-woo Kwak, Dongjin Kang, and Jinyoung Yeo. Web-Shepherd: Advancing PRMs for reinforcing web agents. In *Advances in Neural Information Processing Systems*, volume 38, pp. 70623–70665, 2025b. doi: 10.52202/085713-2127.

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 (Volume 1: Long Papers)*, pp. 9313–9332, 2024. doi: 10.18653/v1/2024.acl-long.505.

Karl Cobbe, Vineet Kosaraju, Mohammad Bavarian, Mark Chen, Heewoo Jun, Lukasz Kaiser, Matthias Plappert, Jerry Tworek, Jacob Hilton, Reiichiro Nakano, Christopher Hesse, and John Schulman. Training verifiers to solve math word problems. *arXiv preprint arXiv:2110.14168*, 2021. URL https://arxiv.org/abs/2110.14168.

Xiang Deng, Yu Gu, Boyuan Zheng, Shijie Chen, Samuel Stevens, Boshi Wang, Huan Sun, and Yu Su. Mind2Web: Towards a generalist agent for the web. In *Advances in Neural Information Processing Systems (Datasets and Benchmarks Track)*, volume 36, pp. 28091–28114, 2023. doi: 10.52202/075280-1220.

Bradley Efron. Bootstrap methods: Another look at the jackknife. *The Annals of Statistics*, 7(1): 1–26, 1979. doi: 10.1214/aos/1176344552.

Tianqing Fang, Hongming Zhang, Zhisong Zhang, Kaixin Ma, Wenhao Yu, Haitao Mi, and Dong Yu. WebEvolver: Enhancing web agent self-improvement with co-evolving world model. In *Proceedings of the 2025 Conference on Empirical Methods in Natural Language Processing*, pp. 8959–8975, 2025. doi: 10.18653/v1/2025.emnlp-main.454.

GLM-V Team, Wenyi Hong, Wenmeng Yu, Xiaotao Gu, Guo Wang, Guobing Gan, Haomiao Tang, Jiale Cheng, Ji Qi, Junhui Ji, et al. GLM-4.5V and GLM-4.1V-Thinking: Towards versatile multimodal reasoning with scalable reinforcement learning. *arXiv preprint arXiv:2507.01006*, 2025. URL https://arxiv.org/abs/2507.01006.

Tilmann Gneiting and Adrian E. Raftery. Strictly proper scoring rules, prediction, and estimation. *Journal of the American Statistical Association*, 102(477):359–378, 2007. doi: 10.1198/016214506000001437.

Boyu Gou, Ruohan Wang, Boyuan Zheng, Yanan Xie, Cheng Chang, Yiheng Shu, Huan Sun, and Yu Su. Navigating the digital world as humans do: Universal visual grounding for GUI agents. In *International Conference on Learning Representations*, 2025. URL https://openreview.net/forum?id=kxnoqaisCT.

Yu Gu, Kai Zhang, Yuting Ning, Boyuan Zheng, Boyu Gou, Tianci Xue, Cheng Chang, Sanjari Srivastava, Yanan Xie, Peng Qi, Huan Sun, and Yu Su. Is your LLM secretly a world model of the Internet? Model-based planning for web agents. *Transactions on Machine Learning Research*, 2025. URL https://openreview.net/forum?id=c6l7yA0HSq.

Chuan Guo, Geoff Pleiss, Yu Sun, and Kilian Q. Weinberger. On calibration of modern neural networks. In *Proceedings of the 34th International Conference on Machine Learning*, volume 70 of *Proceedings of Machine Learning Research*, pp. 1321–1330, 2017. URL https://proceedings.mlr.press/v70/guo17a.html.

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 (Volume 1: Long Papers)*, pp. 6864–6890, 2024. doi: 10.18653/v1/2024.acl-long.371.

Edward J. Hu, Yelong Shen, Phillip Wallis, Zeyuan Allen-Zhu, Yuanzhi Li, Shean Wang, Lu Wang, and Weizhu Chen. LoRA: Low-rank adaptation of large language models. In *International Conference on Learning Representations*, 2022. URL https://openreview.net/forum?id=nZeVKeeFYf9.

Woosung Koh, Sungjun Han, Segyu Lee, Se-Young Yun, and Jamin Shin. Generative visual code mobile world models. In *International Conference on Machine Learning*, 2026. URL https://openreview.net/forum?id=ZI6dgwPpoE.

Wei Li, William Bishop, Alice Li, Chris Rawles, Folawiyo Campbell-Ajala, Divya Tyamagundlu, and Oriana Riva. On the effects of data scale on UI control agents. In *Advances in Neural Information Processing Systems (Datasets and Benchmarks Track)*, volume 37, pp. 92130–92154, 2024. doi: 10.52202/079017-2925.

Hunter Lightman, Vineet Kosaraju, Yura Burda, Harri Edwards, Bowen Baker, Teddy Lee, Jan Leike, John Schulman, Ilya Sutskever, and Karl Cobbe. Let's verify step by step. In *International Conference on Learning Representations*, 2024. URL https://openreview.net/forum?id=v8L0pN6EOi.

Evan Zheran Liu, Kelvin Guu, Panupong Pasupat, Tianlin Shi, and Percy Liang. Reinforcement learning on web interfaces using workflow-guided exploration. In *International Conference on Learning Representations*, 2018. URL https://openreview.net/forum?id=ryTp3f-0-.

Ilya Loshchilov and Frank Hutter. Decoupled weight decay regularization. In *International Conference on Learning Representations*, 2019. URL https://openreview.net/forum?id=Bkg6RiCqY7.

Dezhao Luo, Bohan Tang, Kang Li, Georgios Papoudakis, Jifei Song, Shaogang Gong, Jianye Hao, Jun Wang, and Kun Shao. ViMo: A generative visual GUI world model for app agents. In *International Conference on Learning Representations*, 2026. URL https://openreview.net/forum?id=mWoMyDEfbM.

Quinn McNemar. Note on the sampling error of the difference between correlated proportions or percentages. *Psychometrika*, 12(2):153–157, 1947. doi: 10.1007/BF02295996.

Meta AI. The Llama 4 herd: The beginning of a new era of natively multimodal AI innovation, 2025. URL https://ai.meta.com/blog/llama-4-multimodal-intelligence/.

Yujia Qin, Yining Ye, Junjie Fang, Haoming Wang, Shihao Liang, Shizuo Tian, Junda Zhang, Jiahao Li, Yunxin Li, Shijue Huang, et al. UI-TARS: Pioneering automated GUI interaction with native agents. *arXiv preprint arXiv:2501.12326*, 2025. URL https://arxiv.org/abs/2501.12326.

Christopher Rawles, Sarah Clinckemaillie, Yifan Chang, Jonathan Waltz, Gabrielle Lau, Marybeth Fair, Alice Li, William E. Bishop, Wei Li, Folawiyo Campbell-Ajala, Daniel Kenji Toyama, Robert James Berry, Divya Tyamagundlu, Timothy P. Lillicrap, and Oriana Riva. AndroidWorld: A dynamic benchmarking environment for autonomous agents. In *International Conference on Learning Representations*, 2025. URL https://openreview.net/forum?id=il5yUQsrjC.

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*, volume 70 of *Proceedings of Machine Learning Research*, pp. 3135–3144, 2017. URL https://proceedings.mlr.press/v70/shi17a.html.

Zikai Xiao, Jianhong Tu, Chuhang Zou, Yuxin Zuo, Zhi Li, Peng Wang, Bowen Yu, Fei Huang, Junyang Lin, and Zuozhu Liu. WebWorld: A large-scale world model for web agent training. In *International Conference on Machine Learning*, 2026. URL https://openreview.net/forum?id=ES7Vy9rvRl.

Xiaomi LLM-Core Team, Zihao Yue, Zhenru Lin, Yifan Song, Weikun Wang, Shuhuai Ren, Shuhao Gu, Shicheng Li, Peidian Li, Liang Zhao, et al. MiMo-VL technical report. *arXiv preprint arXiv:2506.03569*, 2025. URL https://arxiv.org/abs/2506.03569.

Weikai Xu, Kun Huang, Yunren Feng, Jiaxing Li, Yuhan Chen, Yuxuan Liu, Zhizheng Jiang, Heng Qu, Pengzhi Gao, Wei Liu, Jian Luan, Xiaolin Hu, and Bo An. How mobile world model guides GUI agents? *arXiv preprint arXiv:2605.10347*, 2026. URL https://arxiv.org/abs/2605.10347.

Xiaohua Zhai, Basil Mustafa, Alexander Kolesnikov, and Lucas Beyer. Sigmoid loss for language image pre-training. In *Proceedings of the IEEE/CVF International Conference on Computer Vision*, pp. 11975–11986, 2023. URL https://openaccess.thecvf.com/content/ICCV2023/html/Zhai_Sigmoid_Loss_for_Language_Image_Pre-Training_ICCV_2023_paper.html.

Yao Zhang, Shijie Tang, Zeyu Li, Zhen Han, and Volker Tresp. WebArbiter: A generative reasoning process reward model for web agents. In *International Conference on Learning Representations*, 2026. URL https://openreview.net/forum?id=canA6Ef0RP.

Boyuan Zheng, Boyu Gou, Jihyung Kil, Huan Sun, and Yu Su. GPT-4V(ision) is a generalist web agent, if grounded. In *Proceedings of the 41st International Conference on Machine Learning*, volume 235 of *Proceedings of Machine Learning Research*, pp. 61349–61385, 2024. URL https://proceedings.mlr.press/v235/zheng24e.html.

Yuhao Zheng, Li'an Zhong, Yi Wang, Rui Dai, Kaikui Liu, Xiangxiang Chu, Linyuan Lv, Philip Torr, and Kevin Qinghong Lin. Code2World: A GUI world model via renderable code generation. *arXiv preprint arXiv:2602.09856*, 2026. URL https://arxiv.org/abs/2602.09856.

A Supporting Details for the Fixed-Prediction Study

A.1 Candidate pool and evaluator configurations

All evaluator comparisons use the same $3{,}000$ AndroidControl evaluation steps, Qwen2.5-VL-7B candidates ($K \leq 3$), and cached Code2World-8B predictions. Reference matches follow the benchmark's deterministic element–bounding-box matcher. The self-evaluator is Qwen2.5-VL-7B. The external evaluators are Claude Sonnet 4.6 (Anthropic, 2026b) and Claude Opus 4.8 (Anthropic, 2026a).

The paired changes in Reference-Match Accuracy relative to Policy Top-1 are -1.4 percentage points for self-evaluation (95% interval: $[-2.3, -0.6]$), $+1.7$ for Claude Sonnet 4.6 ($[+0.6, +2.9]$), and $+2.5$ for Claude Opus 4.8 ($[+1.3, +3.8]$). Intervals use episode-clustered bootstrap resampling (Efron, 1979).

A.2 Override filtering and signal interpretation

Using the reference-match labels from Eq. (2), define

$$\Delta_t = y_{t,\hat{k}_t} - y_{t,k_t^{\mathrm{top}}}, \tag{5}$$

where $\hat{k}_t$ and k_t^{top} index the lookahead and Policy Top-1 choices. Override-Benefit AUROC distinguishes $\Delta_t = +1$ from $\Delta_t = -1$ on decisions with $\Delta_t \neq 0$; its labels therefore measure whether an override improves or harms reference matching.

For each evaluator, Override-Benefit AUROC is computed on decisions where that evaluator's override changes reference-match status. Claude Opus 4.8 changes the selected action on 797 of the $3{,}000$ decisions, of which 366 have $\Delta_t \neq 0$; Qwen self-evaluation and Claude Sonnet 4.6 yield 172 and 330 informative decisions, respectively. The confidence-based short circuit remains part of the lookahead protocol throughout.

Policy confidences are verbalized scores with a few coarse levels. We preserve each signal's direction when reporting AUROC, so values below 0.5 indicate association in the opposite direction. Reference-grounded signals compare a predicted observation with the successor recorded for the reference action. Contrast scores subtract agreement for the Policy Top-1 candidate from agreement for the lookahead-selected candidate; per-candidate scores omit that subtraction, and spread signals summarize variation across the candidate set.

A.3 DIAGNOSTIC SIGNALS

Table 9 reports the full signal sweep with episode-clustered confidence intervals. Benjamini–Hochberg correction is applied within each evaluator panel (Benjamini & Hochberg, 1995).

Table 9: **Complete diagnostic signal sweep.** Directional Override-Benefit AUROC with 95% episode-clustered bootstrap intervals. Each column uses its evaluator's own informative overrides: Sonnet 4.6 $n = 330$ and Opus 4.8 $n = 366$. AUROC values with within-panel Benjamini–Hochberg adjusted $q < 0.05$ are bold. "Contrast" denotes selected-minus-Policy Top-1; "scalar" denotes a per-candidate or step-level score.

Signal	Reduction	Sonnet 4.6	Opus 4.8
Requires the reference successor			
Visual agreement (SigLIP)	Contrast	**0.725** [0.663, 0.787]	**0.746** [0.689, 0.802]
Visual agreement (SigLIP)	Scalar	**0.608** [0.537, 0.675]	**0.590** [0.521, 0.660]
DOM agreement	Contrast	**0.711** [0.646, 0.772]	**0.671** [0.604, 0.733]
DOM agreement	Scalar	0.530 [0.460, 0.598]	0.547 [0.476, 0.616]
Element grounding	Contrast	**0.552** [0.529, 0.575]	**0.548** [0.529, 0.569]
Element grounding	Scalar	**0.552** [0.524, 0.581]	**0.546** [0.521, 0.572]
Text embedding	Contrast	0.517 [0.444, 0.584]	0.495 [0.423, 0.565]
Text embedding	Scalar	0.528 [0.459, 0.600]	0.514 [0.444, 0.583]
Available before execution			
WM likelihood	Contrast	0.562 [0.501, 0.627]	0.541 [0.482, 0.600]
WM likelihood	Scalar	**0.597** [0.538, 0.656]	**0.587** [0.529, 0.643]
WM confidence	Contrast	**0.579** [0.518, 0.641]	0.533 [0.473, 0.591]
WM confidence	Scalar	0.539 [0.476, 0.604]	0.492 [0.430, 0.550]
Likelihood spread	Set spread	0.503 [0.440, 0.567]	0.502 [0.444, 0.564]
Confidence spread	Set spread	0.438 [0.378, 0.497]	0.475 [0.418, 0.537]
Policy confidence	Scalar	**0.428** [0.372, 0.483]	**0.430** [0.376, 0.480]
Negated selector score	Scalar	0.430 [0.368, 0.494]	0.458 [0.399, 0.518]

B DIRECT ACTION RANKING: IMPLEMENTATION AND SENSITIVITY

B.1 IMPLEMENTATION AND EVALUATION CONVENTIONS

DAR evaluates one candidate at a time. The reference action and the other candidates are not model inputs; candidates from the same observation are grouped only to compute the listwise training loss. The mobile input contains the current screenshot, goal, visible accessibility elements with bounding boxes, and the proposed action with its resolved target element. The web instantiation uses the goal and interactable-element representation with the proposed element and operation. Mobile labels follow the deterministic AndroidControl element–bounding-box matcher (Li et al., 2024); web labels use element, operation, and value equivalence.

Table 10 gives the main mobile DAR training configuration. Groups in which every candidate matches are dropped during training. For retained groups with no match, the listwise term is zero and the pointwise loss supplies negative supervision. The same mobile scoring prompt is used for training and evaluation.

The two AndroidControl columns of Table 2 use the same Qwen2.5-VL-7B-proposed evaluation candidate pool; the two DAR evaluators are trained on Qwen2.5-VL-7B and external-proposer candidates, respectively. The Mind2Web columns use a web DAR evaluator trained on Mind2Web and evaluated on the Cross-Task, Cross-Website, and Cross-Domain test splits (Deng et al., 2023).

Matched predictive augmentation. Section 5.3 evaluates predictive augmentation on an external-proposer pool and a Qwen2.5-VL-7B-proposed candidate pool. Within each pool, DAR and prediction-augmented DAR use matched candidates, labels, backbone, objective, and optimization settings; the latter additionally receives the predicted successor. The paired AUROC changes are -0.0003 (95% CI: $[-0.003, +0.003]$) and $+0.0005$ (95% CI: $[-0.003, +0.004]$), respectively, using 2,000 episode-clustered bootstrap resamples.

Table 10: **Main mobile DAR configuration.**

Component	Setting
Backbone	Qwen3-VL-8B-Instruct; base weights frozen; language-model head removed
Trainable components	LoRA adapters and scalar head; 87.3M trainable parameters
LoRA configuration	Rank 32, scaling 64, dropout 0; decoder text projections only; vision tower frozen
Scoring head	FP32 linear map from the final-token hidden state to a scalar
Candidate generation	Up to 3 candidates from one proposer decode
Objective	Listwise softmax and $\beta = 0.5$ pointwise BCE
Batching	4 observation groups per device; candidates in a group remain together
Optimizer	AdamW (Loshchilov & Hutter, 2019); learning rates 10^{-4} (head) and 5×10^{-6} (LoRA)
Schedule	2 epochs; 20 warmup steps; gradient clipping 1.0; three seeds
Precision / hardware	BF16 mixed precision; 8 H100 GPUs with distributed data parallelism
Mobile image handling	Screenshot long side capped at 1,024 pixels
Basic inference	One scoring forward pass per candidate; highest-score selection; no generated successor

B.2 POLICY-SFT COMPARISON BASELINE

The policy-SFT baseline uses Qwen3-VL-32B-Instruct with LoRA rank 64 and scaling 128 on decoder text projections; the vision tower is frozen. The full-budget SFT setting uses the same 26,365 reference-action pool, prompt skeleton, and action vocabulary as the full-budget DAR comparison. All outputs are evaluated in the same pixel-coordinate action space: the zero-shot checkpoint natively emits Qwen3-VL's 0–1000 coordinates, which are deterministically mapped to pixels before action deduplication, while the SFT checkpoints emit pixel coordinates directly.

Table 11: **Proposer-SFT baseline.** Greedy reference matching, candidate-set size, and Pass@3 on the shared evaluation fold.

Arm	Greedy	Mean $\lvert\mathcal{A}\rvert$	Pass@3
Qwen3-VL-32B zero-shot	0.6137	2.7259	0.7430
SFT epoch 1	0.6447	1.3513	0.7373
SFT epoch 2	0.6543	1.3367	0.7347
SFT epoch 2, beam-3	0.6533	2.1057	0.7727
SFT on 2,000 labels	0.4797	1.6780	0.6187

On the SFT epoch-2 candidate sets, DAR adds 0.0267 selected-action matching. With 2,000 labels, the DAR selector improves selected-action matching by 0.038 on its fixed candidate pool.

B.3 CANDIDATE-SET SIZE

Table 12 evaluates all 3,000 AndroidControl evaluation decisions. The $K \leq 3$ settings are nested subsets of the original pool: the Policy Top-1 candidate is retained, and lower-confidence alternatives are removed first. The two-decode setting adds an independent second decode of the same prompt and deduplicates actions by equivalence. Policy Top-1 remains the highest-confidence candidate from the first decode throughout.

Table 12: **Candidate-budget sensitivity.** Reference coverage is the proportion of decisions containing at least one matching candidate. Selected-action columns report Reference-Match Accuracy; gains are absolute fractions. The two-decode setting has mean candidate count 3.64.

Candidate set	Ref. coverage	Policy Top-1	DAR	Gain
$K = 1$	0.438	0.438	0.438	+0.000
$K \leq 2$	0.528	0.438	0.489	+0.051
$K \leq 3$	0.561	0.438	0.501	+0.063
Two decodes	0.608	0.438	0.525	+0.086

DAR improves selected-action matching over Policy Top-1 whenever alternative candidates are available. The second decode increases both reference coverage and selection accuracy.

B.4 Training-data sensitivity

Table 13 uses nested subsets of one Qwen2.5-VL-7B-proposed training pool with the configuration in Table 10. All models are evaluated on the same 3,000-decision pool. Training examples count sampled observations before the all-matching filter; groups count the observations retained after that filter.

Table 13: **Training-data sensitivity.** AUROC includes 95% intervals. Conditional match accuracy measures selected-action matching on observations whose candidate set contains a reference match.

Training examples	Groups	Ref.-Match AUROC	Cond. match acc.
2,000	1,859	0.900 [0.890, 0.910]	0.849
5,000	4,664	0.932 [0.923, 0.940]	0.890
10,000	9,345	0.947 [0.939, 0.954]	0.901
26,365	24,636	0.957 [0.951, 0.964]	0.911

Both candidate discrimination and conditional selection accuracy improve as the training set grows.

C Online Evaluation, Transfer, and Sensitivity

C.1 AndroidWorld deployment and paired comparisons

The mobile DAR model is trained on Qwen2.5-VL-7B-proposed AndroidControl candidates and reused for all three AndroidWorld proposers, Llama-4 Maverick (Meta AI, 2025), Qwen3-VL-8B, and Qwen3-VL-235B (Bai et al., 2025a), without parameter updates. Each task is evaluated under seeds $\{30, 31, 32\}$, with up to three candidates from a single proposer decode at each decision. Policy Top-1 selects by proposer confidence; DAR selects by its learned score. Lookahead–External and Lookahead–Self score Code2World-8B successors with fixed evaluator prompts across all online runs; both retain the confidence-based short circuit to Policy Top-1.

All 116 tasks remain in the denominator, and execution errors count as failures for the affected configuration. Table 4 gives the application-family breakdown. Table 14 reports paired contrasts using 2,000 task-clustered bootstrap resamples (Efron, 1979) and exact McNemar tests (McNemar, 1947). Online trajectories may diverge after different selected actions, so pairing is defined by task and seed rather than by later candidate sets.

Table 14: **Paired AndroidWorld task-success contrasts.** DAR minus each baseline, pooled over seeds 30, 31, and 32. Gains are within-seed paired differences in percentage points. Intervals use task-clustered bootstrap; p-values use exact McNemar tests. Paired differences are computed before rounding marginal means.

Proposer	Baseline	Gain	95% CI	p
Llama-4 Maverick	Top-1	+16.4	[+10.1, +22.7]	8×10^{-12}
	External	+9.2	[+4.3, +14.4]	10^{-4}
	Self	+11.5	[+7.2, +16.1]	2×10^{-7}
Qwen3-VL-8B	Top-1	+11.5	[+6.9, +16.4]	2×10^{-7}
	External	+7.8	[+3.4, +12.1]	5×10^{-4}
	Self	+6.9	[+3.4, +10.3]	9×10^{-4}
Qwen3-VL-235B	Top-1	+5.8	[+1.7, +9.5]	0.008
	External	+4.3	[+0.0, +8.6]	0.049
	Self	+5.8	[+1.7, +9.8]	0.005

C.2 MWMBENCH-KAPPS TRANSFER

KApps contains 488 steps from 230 episodes across 14 Korean applications. The same mobile DAR evaluator is applied without retraining to candidates from all three proposers. Table 6 reports Reference-Match AUROC as mean ± standard deviation over three candidate-generation seeds. Policy confidence does not use world-model output; the lookahead conditions use Code2World and gWorld predictions.

C.3 MINIWOB WEB EVALUATION

A web DAR evaluator is trained on Mind2Web and reused with Qwen3-VL-8B and Qwen3-VL-235B on MiniWoB (Shi et al., 2017; Liu et al., 2018). We evaluate the 98 tasks expressible in the action vocabulary {`CLICK`, `TYPE`, `SELECT`}. Each task uses three environment seeds with one trajectory per seed, and all configurations share the same task–seed pairs. Table 7 reports task-family results with s.e.m. over the three per-seed success rates; paired-gain s.e.m. is computed from the per-seed differences.

C.4 EVALUATOR-BACKBONE AND OBJECTIVE SENSITIVITY

Table 15 adds paired online significance tests for the backbone variants, MiMo-VL-7B-RL (Xiaomi LLM-Core Team et al., 2025) and GLM-4.1V-9B-Thinking (GLM-V Team et al., 2025), in main-text Table 8(a), using the same candidate data and training configuration. Figure 3 complements Table 8(b) by visualizing the score calibration and distributions of the full and no-pointwise objectives (Guo et al., 2017; Gneiting & Raftery, 2007).

Table 15: **Backbone sensitivity with paired online tests.** AndroidControl AUROC and AndroidWorld success for the Qwen3-VL-8B proposer; online values are three-seed mean ± s.e.m. The final column reports exact paired tests against Top-1.

Backbone	AUROC	Δ AUROC	Online %	Gain pp	p vs. Top-1
Qwen3-VL-8B	.957	0 (reference)	35.3 ± 2.0	+11.5	2×10^{-7}
MiMo-VL-7B-RL	.879	−.078	30.5 ± 1.4	+6.6	.003
GLM-4.1V-9B-Thinking	.884	−.073	28.7 ± 2.3	+4.9	.036

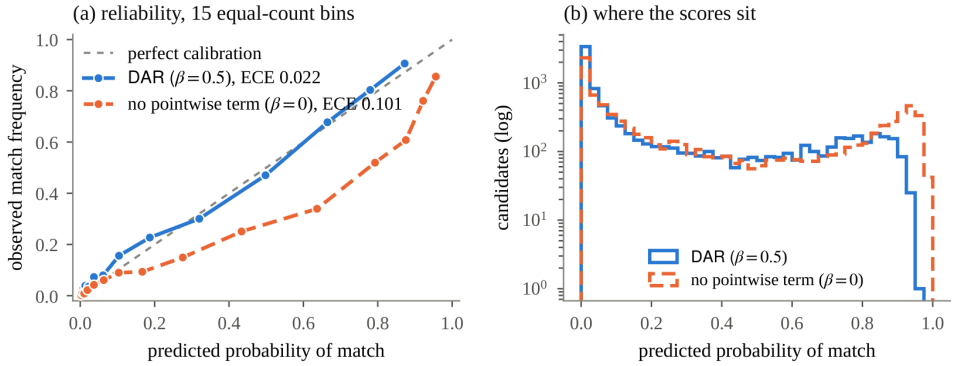


Figure 3: **Score behavior under pointwise supervision.** Reliability curves (15 equal-count bins; 8,839 candidates) and score distributions for the full and no-pointwise Qwen3-VL-8B objectives from Table 8(b).

D PREDICTION DIAGNOSTICS

Table 16 tests two extensions of the fixed-prediction study: longer lookahead and alternative representations of the predicted successor. All settings use Claude Opus 4.8; the one-step native-code result is the reference from Section 4.

Table 16: **Lookahead depth and representation.** Reference-match change relative to Top-1, in absolute fractions, under Claude Opus 4.8. Depth results include 95% confidence intervals; representation results report point estimates.

(a) Lookahead depth (95% CI)			
Depth	One step	+.025	[+.013,+.038]
	Two steps	+.018	[+.001,+.035]
	Three steps	+.019	[+.002,+.036]
(b) Predicted-successor representation (point estimates)			
Representation	Native code	+.025	
	Accessibility tree	+.012	
	Plain text	+.012	

One-step lookahead gives the largest gain in this comparison. Re-encoding the same predicted successor as an accessibility tree or plain text also reduces the gain relative to the native representation.

E Qualitative Action-Selection Cases

Among the 3,000 evaluation decisions, DAR recovers the reference action on 75 steps where Top-1 and lookahead do not. These recoveries comprise 51 clicks, 19 scrolls, 4 back actions, and one text entry. Table 17 reports all 12 cases in which Top-1, lookahead, and DAR select three distinct actions; in the other 63 recoveries, Top-1 and lookahead choose equivalent actions.

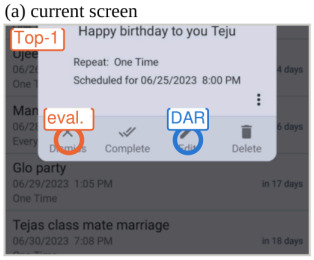
Table 17: **Three-way reference recoveries.** All 12 decisions where Top-1, lookahead, and DAR choose distinct actions and DAR matches the reference. Parentheses show each selector's score for its selected action; DAR margin is the DAR-score difference between the reference action and Top-1.

Step	Choice pattern	Top-1	Lookahead	DAR = reference	Margin
ep18765/1	commit vs. reveal	`click (546,524)`	`click (54,357)` (+0.85)	`scroll down` (+0.46)	+5.62
ep11190/3	back navigation	`click (546,1372)`	`navigate_back` (+0.70)	`click (127,2126)` (+2.03)	+5.37
ep12520/2	app restart	`click (952,2246)`	`open_app Google Play Books` (+0.95)	`scroll down` (−0.03)	+4.65
ep17950/1	adjacent controls	`wait`	`click (228,1553)` (+0.90)	`click (649,1553)` (−1.87)	+4.25
ep160/0	app restart	`navigate_home`	`open_app Music app` (+0.90)	`navigate_back` (−1.42)	+3.97
ep15495/2	app restart	`navigate_back`	`open_app LeafSnap` (+0.90)	`scroll down` (+1.79)	+3.31
ep3570/1	adjacent controls	`click (560,365)`	`click (332,2239)` (+0.95)	`click (201,2239)` (+0.28)	+3.30
ep19950/1	action-modality mismatch	`click (607,1824)`	`input_text "Eyes Closed Nets"` (+0.95)	`scroll down` (+0.99)	+3.21
ep9305/1	back navigation	`click (540,2118)`	`navigate_back` (+0.80)	`scroll down` (−1.67)	+1.75
ep3210/6	adjacent controls	`click (546,1040)`	`click (1005,2176)` (+0.80)	`click (546,1201)` (−0.81)	+1.70
ep19190/2	app restart	`click (169,390)`	`open_app PowerPoint` (+0.90)	`click (969,2202)` (+0.02)	+1.39
ep16635/11	app restart	`navigate_back`	`open_app OYO` (+0.95)	`scroll down` (−0.13)	+0.34

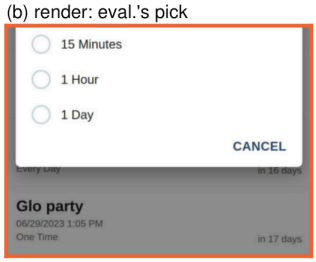
Figure 2 provides one example in the main text. The three cases below highlight complementary mechanisms: ambiguity between nearby controls, variation in generated evidence for equivalent actions, and navigation away from the reference trajectory.

goal: Snooze Teja´s birthday reminder for 5 minutes

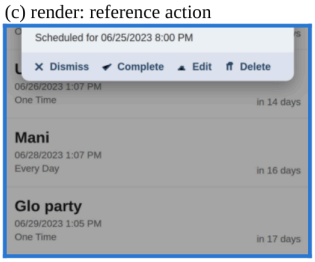
(a) current screen



(b) render: eval.'s pick



(c) render: reference action

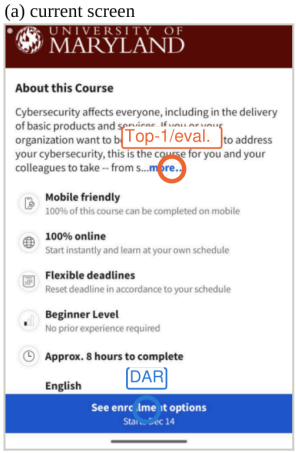


candidate action	policy conf.	eval. score	DAR score	reference?
wait [Top-1]	0.6	-0.85	-6.13	no
click (228,1553) [eval.]	0.6	+0.90	-3.82	no
click (649,1553) [DAR]	0.6	-0.85	-1.87	yes

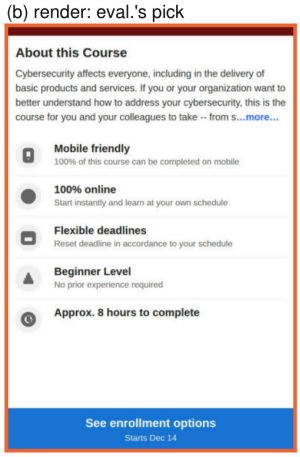
Figure 4: **Neighboring controls.** To snooze a reminder, Top-1 waits, lookahead selects the adjacent control, and DAR selects the reference control.

goal: Enroll for the course "CyberSecurity for Everyone" without a certificate on the Coursera app.

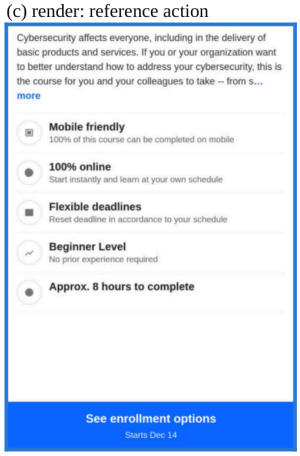
(a) current screen



(b) render: eval.'s pick



(c) render: reference action

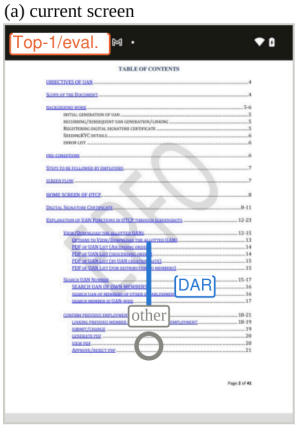


candidate action	policy conf.	eval. score	DAR score	reference?
click (626,1349) [Top-1]	0.8	-0.80	-4.37	no
click (532,2250) [DAR]	0.8	-0.60	+1.79	yes
click (626,1349) [eval.]	0.8	+0.85	-4.37	no

Figure 5: **Equivalent actions, different predictions.** Equivalent non-reference clicks receive different lookahead scores; DAR scores them identically and selects the reference control.

goal: I would like to see the EPFO file in the Dropbox app.

(a) current screen



(b) render: eval.'s pick



(c) render: reference action



candidate action	policy conf.	eval. score	DAR score	reference?
navigate_home [Top-1, eval.]	0.9	+0.60	-5.02	no
scroll down [DAR]	0.7	-0.80	+0.57	yes
click (540,1200)	0.6	-0.80	-3.04	no

Figure 6: **Navigation choice.** Lookahead prefers navigating home, while DAR ranks the reference scroll first.