DEMONSTRATION CONDITIONING GENERALLY IMPROVES JAGGED FRONTIER MODEL PERFORMANCE ON DIGITAL AND PHYSICAL AGENTIC TASKS

Anonymous authors
Paper under double-blind review

ABSTRACT

We measure how much an in-context demonstration improves frontier models, and how much of that improvement comes from the demonstration containing the answer. Benchmark averages hide both: they show neither where inside a benchmark an agent is weak nor whether a demonstration taught a procedure or supplied the item's answer. Holding the prompt fixed, we vary only the demonstration for seven models, including GPT Astra and Claude Opus 5.5, on tasks across web automation, manipulation, assembly, industrial logistics, and self-driving. A demonstration from the same situation raises the score in all five domains and for all seven web models. With the answer removed, the gain survives on driving and manipulation for all three models and disappears on assembly. Across the seven models the gain falls as the baseline rises, narrowing the gap between the best and worst model while the unevenness inside each benchmark remains.

1 INTRODUCTION

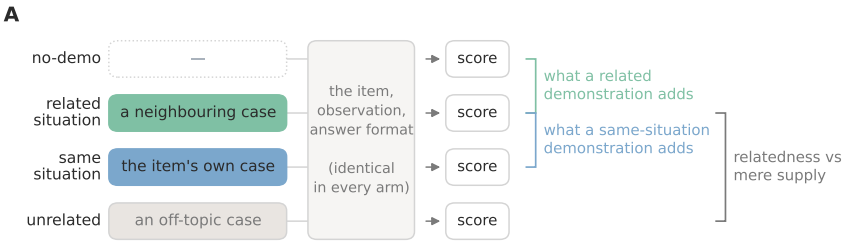
Vision-language agents are deployed in computer use (Koh et al., 2024), tabletop manipulation (Zhang et al., 2024), driving (Tian et al., 2024; Jia et al., 2024), toy assembly (Sener et al., 2022), and industrial procedural tasks (Chavan et al., 2025), and each benchmark reports their competence as a single average score. That average hides how unevenly competence is distributed across the benchmark's own categories. Prior studies of this unevenness are either theoretical or evaluate text-centric agents (Gans, 2026; Rabanser et al., 2026; Srinivasan, 2026; Kapoor et al., 2025), and few measure it across multimodal agentic domains. The cheapest training-free remedy is a demonstration placed in context at inference time. Its forms range from few-shot prompting (Brown et al., 2020; Alayrac et al., 2022) to procedural-memory systems that retrieve a previous episode and insert it into the prompt as text, as frames, or as both interleaved (Ding et al., 2026; Luo et al., 2026; Jiang et al., 2026; Kohar & Krishnan, 2025). We call this intervention *demonstration conditioning*, and we call the change it produces in a model's score that model's *demonstration-following*.

How much a demonstration adds in these domains has received little study. Work on demonstration conditioning either studies multimodal question answering or reports an aggregate lift one benchmark at a time, which cannot show where procedural knowledge transfers to an unseen situation (Wang et al., 2026; Zhang et al., 2026).

We run a controlled pass that holds the prompt fixed and varies only the demonstration block (Figure 1). Every item is scored under a no-demonstration floor, a same-situation ceiling, a related-situation arm, and an unrelated control arm where the domain admits one. Measuring answer availability per item, rather than assuming it from the arm label, splits the related and unrelated arms into answer-present and answer-absent subsets. Seven models run on the closed-loop web domain, where a retriever selects the demonstration, and three models run on four offline physical domains, where the arms are built from dataset metadata. Two models, GPT-6 Astra and Opus 5.5, run on all five domains.

The no-demonstration arm is jagged across every benchmark's own categories, confirming the unevenness reported for real-world deployments and language-only agents (Rabanser et al., 2026; Kapoor et al., 2025). A same-situation demonstration then helps in all five domains. Each offline physical domain gains on its own metric, and on VisualWebArena all seven models gain, by $+1.7$ to $+14.1$pp (one-sided sign test over the seven paired lifts, $p = 0.008$; two separate passes of Opus 4.7

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.



B

	VisualWebArena	VLABench	Assembly101	IndEgo	Bench2Drive-VL
same	its own episode (66% of tasks)	own episode with answer	same toy, other session	same task, other worker	its own future window
related	a different task's episode	three demos, without answer	same family, other toy	same scenario, other task	feature-matched, other route
unrelated	*none*	*none*	different family	any different task	*none*

Figure 1: **The arm ladder. (A)** The arms of a domain that admits a constructed demonstration; everything outside the demonstration block is held fixed. An arm names how the demonstration was selected; whether it contains the item's answer is recorded separately (Table 2). **(B)** What each arm is per domain. Every VLABench arm uses the item's own episode and varies the answer instead of the situation. Blue marks same-situation arms and green related-situation arms.

agree on the paired lift to within 0.01pp). The three smallest lifts, GPT-6 Astra at $+1.7$, Fable 5 at $+2.8$ and GPT-5.6 at $+4.0$pp, fall inside or just above the ± 3pp repeat-run band. The mean rollout shortens from 17.2 to 14.5 steps (Table 1). Across the seven models, the lift falls as the baseline rises (rank correlation -0.95, exact permutation $p = 0.002$). The lift and the step reduction both vary across categories as much as the no-demonstration scores do. A related-situation demonstration has a smaller and less consistent effect. Once the items whose demonstration held the answer are removed, a lift of $+0.096$ survives on driving and $+0.225$ on VLABench for Gemini 3.5-Flash, while Assembly101 retains nothing (Table 3). GPT-6 Astra and Opus 5.5 keep $+0.068$ and $+0.026$ on driving, and Assembly101's interval again covers zero. On VisualWebArena the seven-model mean lift is $+9.3$pp where the retriever returned the task's own episode and $+0.9$pp where it did not (Figure 2).

Our contributions are:

1. **An experiment design for studying the effect of demonstration conditioning.** Categorized by whether a demonstration is from same, related or unrelated situations plus a per-item answer-availability record (§2).
2. **A breadth-first cross-model and cross-domain analysis of Frontier Models on digital and physical agentic domains.** Models are evaluated on on web automation and browsing, manipulation, assembly, industrial logistics and repair, and self-driving domains. (§3).
3. **Evidence that demonstration-following, while uneven, generally improves performance** Patterns in the gain from a demonstration are characterized across models, domains and demonstration types.

2 Experimental Setup

Each domain runs in its own native harness, and the contrast between arms is the only element common to all five.

2.1 Definitions and Arm Construction

A **situation** is a task goal together with the scene or entity instance it is grounded in; two items share a situation when an episode of one contains the answer to the other. An **item** is one thing a benchmark scores: a web task, a driving frame with its question, an anticipation point in a recording,

or a manipulation query. An **episode** is one stored prior run; the collection a retriever draws from is the **bank**. A **rollout** is the agent's own run on the item being scored. An **arm** is one prompt configuration, identical outside the demonstration block. In VisualWebArena, where a retriever supplies the demonstration, an item is *matched* when the retriever returned that item's own episode and *mismatched* otherwise. **Answer availability** $a_i \in \{0, 1\}$ is our per-item indicator that item i's answer was present in its demonstration. No single test defines it across the five harnesses, so we use the strongest indicator each domain admits (Table 2): an exact test of the demonstration's contents on Bench2Drive-VL, Assembly101 and IndEgo, availability by construction on VLABench, and on VisualWebArena whether the retrieved episode is the item's own, from which we infer availability without measuring it.

The **lift** from demonstration conditioning is paired per item, $\Delta = s(d) - s(\emptyset)$, with s the benchmark's own scorer, and the **surviving fraction** ρ normalizes a related-situation arm's lift by the same-situation arm's lift on the same items, $\rho = (s_{\text{rel}} - s_{\emptyset})/(s_{\text{same}} - s_{\emptyset})$. Section 3.4 reports both, and Algorithm 1 in the appendix gives the full procedure.

We keep the instruction, observation, action space, scorer, decoding and step budget close to what the benchmark ships, modified only where a model requires it. Output-token budgets are fixed within each model but differ across models. These six settings are identical across arms, so any Δ belongs to the demonstration block within each model and each domain (Table 5, appendix).

2.2 The two axes

The model axis varies the model on the one domain with a live retriever, which is how the intervention is deployed in practice. All seven share a byte-identical system prompt, environment, retrieval settings, a 100-step budget and the same 177-episode bank. What differs is the browser interface each vendor exposes: computer-use tooling for the Opus rows and function calling for the rest, all normalizing to identical BrowserGym actions. This difference confounds cross-model *absolute* success but not the within-model lift. Retrieval takes the top-1 match over the bank from Qwen3-VL-Embedding-8B (Li et al., 2026); the retriever is never tuned against the score and is independent of the agent model. Top-1 returns the task's own episode on 115 to 118 tasks across runs (Appendix A).

The domain axis runs Gemini 3.5-Flash (Google DeepMind, 2026), GPT-6 Astra (OpenAI, 2026b) and Opus 5.5 (Anthropic, 2026d) on the same items and prompts in the four offline physical domains. We use this axis to measure how competence and lift disperse across domains. These domains are scored offline without a rollout, so step counts appear only on the model axis. The four domains build their arms differently and their ceilings differ in kind, so magnitudes are never compared between them (Table 2); what carries across is the direction of an effect and its dispersion inside a domain's own categories.

3 Results

We first measure the jaggedness of baseline competence, then ask how a single demonstration affects it across two setups: seven models on VisualWebArena and three models on the four offline physical domains.

3.1 Dispersion of baseline competence

On our 177-task VisualWebArena subset, the seven models' task success rates are jagged across websites, visual difficulty grades and workflow difficulty grades (Figure 2 top). The spread of baseline success rates runs from 10.4pp (Fable 5) to 23.9pp (Opus 4.7) across websites, and neither difficulty grade predicts the scores: on workflow difficulty, Fable 5, GPT-5.6, GPT-6 Astra and Opus 5.5 all score higher on *hard* than on *medium*. The same holds for all three models on the offline physical domains across Bench2Drive-VL routes, Assembly101 toy types, VLABench tabletop tasks and IndEgo industrial task types (Figure 3, top row). On Bench2Drive-VL, Gemini's route scores range from 0.070 to 0.947 across the 86 routes, and those of the other two models from 0.19 to 0.94–0.97.

Floors, ceilings and noise. Every arm we report is a single pass, so we estimate run-to-run variance from two separate passes of Opus 4.7, run outside the seven-model sweep at a matched configuration

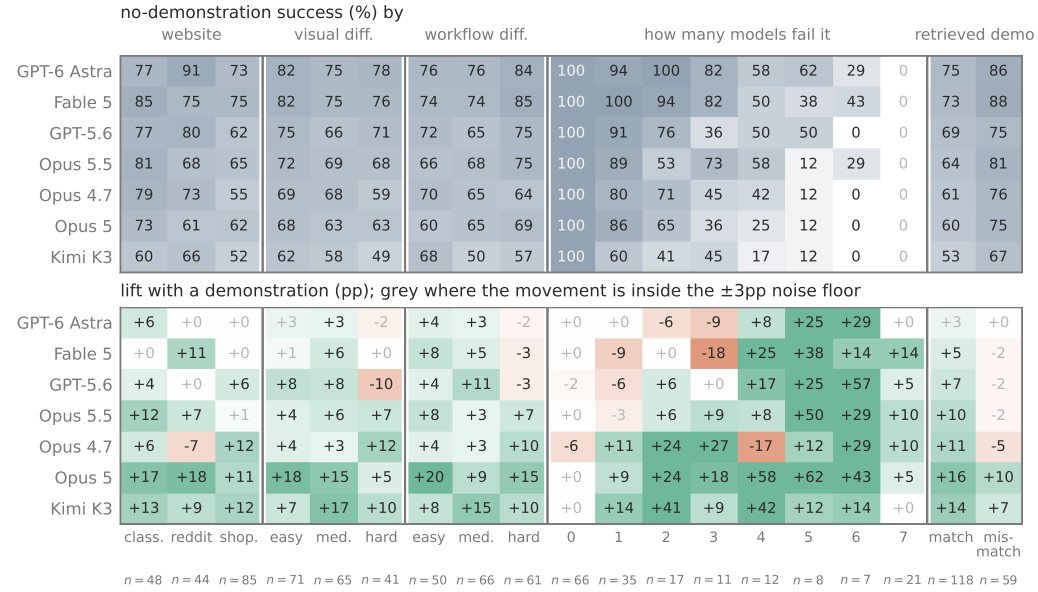


Figure 2: **VisualWebArena, seven models over five groupings of the** 177 **tasks.** The groupings are the website, the two difficulty grades the benchmark ships in its own task configs, an empirical grade bucketing each task by how many of the seven models fail it with no demonstration, and whether the retriever returned the task's own episode. Task counts are printed beneath. **Top:** tasks solved with no demonstration (%). Under *how many models fail it*, the 0 and 7 columns are grayed out because those buckets are defined by the same scores, so every model reads 100 and 0 there. The retrieved-demonstration counts printed beneath, 118/59, hold for four of the seven models; Opus 5, Kimi K3 and Opus 5.5 have 117, 116 and 115 matched tasks. **Bottom:** the change in percentage points, green for a gain and clay for a loss, gray where the change is under 3pp, which is the largest per-website disagreement between two repeat passes of Opus 4.7 at a matched configuration and a shorter step budget (§3.1). No per-cell significance is marked: 68 of the 133 cells change outcome on too few tasks for an exact McNemar test to reach $p < 0.05$, so a mark would confuse a null with an untestable cell.

but a shorter step budget (Appendix A). Their paired lifts agree to within 0.01pp, +6.21 and +6.20pp. Broken down by website, the two passes' lifts differ by at most 1.2pp on shopping (n=85) and 2.3pp on reddit (n=44). We round this up to ±3pp and gray out any cell of Figure 2 that changes by less.

For Gemini 3.5-Flash, Bench2Drive-VL runs between a 0.394 floor and a 0.983 ceiling, Assembly101's floor is 0.110, and IndEgo's no-demonstration arm scores 0.437, below the 0.495 obtained by answering with the demonstration's step at the item's own index, over the 291 items where such a step exists (Appendix B). IndEgo is therefore the most copy-contaminated domain we report: its floor is below that positional-copy score, and it shows the highest positional copying of any domain (§3.5), so its +0.273 is better read as an upper bound on transfer than as transfer itself. This holds only for Gemini: GPT-6 Astra and Opus 5.5 score 0.627 and 0.507 on IndEgo, above the positional-copy score.

3.2 Effect of a same-situation demonstration

All seven models improve on VisualWebArena, by +1.7 (GPT-6 Astra) to +14.1pp (Opus 5) over baselines of 102 to 139 of 177 tasks (Table 1), with Fable 5 at +2.8pp, GPT-5.6 at +4.0pp, Opus 4.7 and Opus 5.5 at +5.6pp and Kimi K3 at +11.3pp. A same-situation demonstration also helps in each of the four offline physical domains for Gemini 3.5-Flash: +0.589 on Bench2Drive-VL, +0.313 on VLABench, +0.273 on IndEgo and +0.088 on Assembly101, each on that domain's own metric, averaged over items (Figure 4). GPT-6 Astra and Opus 5.5 also gain in all four: +0.443 and +0.449 on Bench2Drive-VL, +0.113 and +0.172 on VLABench, +0.193 and +0.247 on IndEgo, and +0.073 and +0.094 on Assembly101 (Table 3).

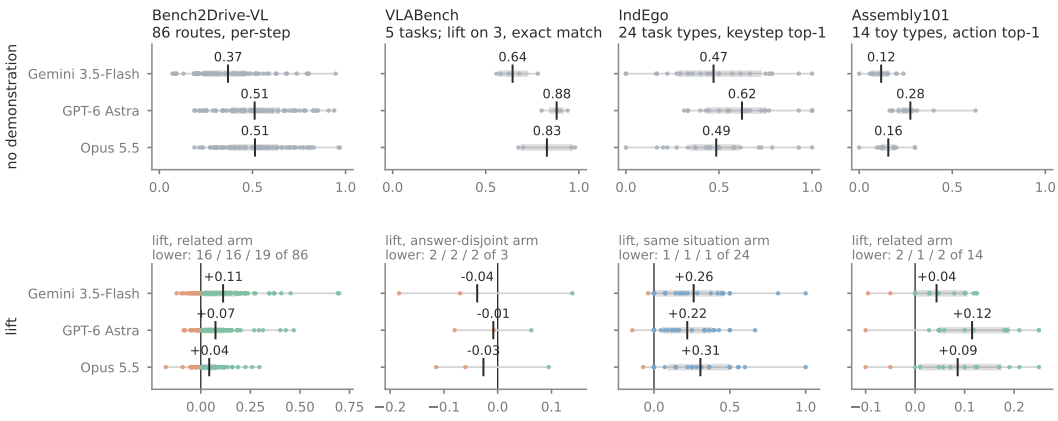


Figure 3: **Three models over each offline physical domain's own categories. Top:** the no-demonstration score, one dot per category and one strip per model, with the middle 50% shaded and the mean marked. **Bottom:** the per-category lift for the arm discussed in §3.2: the related arm on Bench2Drive-VL and Assembly101, the same-situation text arm on IndEgo, and the answer-disjoint arm on the three VLABench tasks that have one. Orange marks a category that scores lower with the demonstration; each panel title counts them per model. Means are unweighted over categories, so they differ from the item means of Table 3. IndEgo counts any valid next step as correct, as in Table 3. VLABench uses exact match; its flower task is the vague-instruction stratum on top and all strata below.

Table 1: **VisualWebArena, all seven models.** Every model runs the same 177 web tasks twice, with and without a retrieved demonstration, so each row compares a model against itself. Task success is printed as tasks solved, so $\Delta = c/177$ and *gap closed* $= \Delta/(1 - \text{none}) = c/(177 - n)$ for a change of c tasks from n solved. Read the imbalance of *newly solved* against *newly failed* rather than either count; their net is the change in tasks solved. Each number is one pass, against the 3pp repeat-run variation of §3.1. GPT-6 Astra runs at `effort=medium` and GPT-5.6 at `high`, so their absolute scores are not comparable, but each model's own lift is (Table 4).

	Task success, of 177			Mean steps	Tasks that changed			Lift by what was retrieved	
Model	none → demo	Δ (pp)	gap closed	none → demo	newly solved	newly failed	net	matched n=115–118	mismatched n=59–62
Fable 5	138 → 143	+2.8	12.8%	16.3 → 12.7	14	9	+5	+5.1	−1.7
GPT-5.6	125 → 132	+4.0	13.5%	19.8 → 17.4	14	7	+7	+6.8	−1.7
Opus 4.7	117 → 127	+5.6	16.7%	19.2 → 14.5	21	11	+10	+11.0	−5.1
Opus 5	115 → 140	+14.1	40.3%	14.3 → 13.8	27	**2**	+25	+16.2	+10.0
Kimi K3	102 → 122	+11.3	26.7%	20.1 → 16.6	29	9	+20	+13.8	+6.6
GPT-6 Astra	139 → 142	+1.7	7.9%	15.3 → 12.6	7	4	+3	+2.5	0.0
Opus 5.5	124 → 134	+5.6	18.9%	15.2 → 14.2	19	9	+10	+9.6	−1.6

Rollout length. The average step reduction on VisualWebArena is 2.6 steps, running from 0.5 for Opus 5 to 4.7 for Opus 4.7 (Table 1). It holds at 0.3 to 3.5 steps on the tasks the demonstration did not rescue, so it is not an artifact of the newly solved ones, though rollout inspection (Appendix D) shows that part of it is final-answer copying. The step reduction is also jagged, and in some cases steps increase: Opus 5's reddit rollouts lengthen from 11.9 to 15.7 steps because its baseline answered incorrectly within two or three steps, while with the demonstration it carried out the task.

Dispersion of the lift. Demonstration conditioning changes performance across VisualWebArena's task categories by different amounts and in opposite directions: per-site lift over the seven models runs from −6.8 to +18.2pp with both extremes on reddit tasks (Figure 2 bottom). Across the three websites within one model, the lift spans 3.4pp for Kimi K3 and 18.6pp for Opus 4.7, the same order as the baseline's 10.4 to 23.9pp. The two profiles do not align. Only Opus 4.7 reverses sign

across websites, from -6.8pp on reddit to $+11.8$pp on shopping, and a website's baseline does not predict its lift. The raw lift depends partly on the model's baseline. Across the seven models the rank correlation between baseline and lift is -0.95 (permutation $p = 0.002$; task-bootstrap 95% interval $[-0.96, -0.40]$), and -0.93 against gap closed, so the trend is not only a matter of stronger models having less room to improve. The best and worst models differ by 37 tasks without a demonstration and 21 with one.

On the related arm, 16 of 86 Bench2Drive-VL routes lose 0.007 to 0.122 against a mean $+0.154$ on the 68 routes that gain, and 2 of 14 Assembly101 toys lose 0.095 and 0.050 (GPT-6 Astra: 16 routes and 1 toy; Opus 5.5: 19 routes and 2 toys); 1 of 24 IndEgo task types loses ground on the same-situation arm for each of the three models (Figure 3). When the web tasks are bucketed by how many of the seven models fail them at baseline, a demonstration changes the bucket every model solves by 0 to -6pp, gains most in the middle buckets, by up to $+62$pp, and changes the bucket every model fails by 0 to $+14$pp. Seven middle cells are negative, down to -18pp, but none is significant, and two Opus 4.7 baseline passes differ by -22 to $+27$pp in the same buckets, so they are within repeat-run noise. A category's baseline and its lift share a term, so we estimate them on disjoint halves; only Bench2Drive-VL's routes give split-half estimates stable enough to correlate (reliability 0.90). There, difficulty predicts the lift at $r = -0.39$ for Gemini ($p = 0.0002$) against a naive -0.47, and at -0.31 ($p = 0.003$) and -0.26 ($p = 0.016$) for GPT-6 Astra and Opus 5.5, while the IndEgo and Assembly101 groupings are untestable rather than null (Table 8).

Demonstration conditioning partly measures refusal override. The rollout traces identify the largest single cause of one model's lift: of Opus 5's 27 rescued tasks, six are baseline *refusals* and three are stopped-short completions, and dropping the refusals takes its gap closed from 40.3% to 30.6%, next to Kimi K3's 26.7%. On those tasks the demonstration removed a refusal rather than supplying task content, which is consistent with safety tuning depressing the baseline and inflating the apparent lift.

The same dispersion appears on the domain axis. Within each model, how much a demonstration is worth still depends on the category. Across Bench2Drive-VL's 86 routes, the related arm runs from -0.122 to $+0.698$ about a mean of $+0.113$, with 16 routes losing ground. Across Assembly101's 14 toys it runs from -0.095 to $+0.125$ about a mean of $+0.043$, two toys negative and three unmoved. IndEgo's 24 task types spread even on the same-situation arm, -0.038 to $+1.000$ about a mean of $+0.262$. GPT-6 Astra and Opus 5.5 spread the same way (Figure 3, bottom row). VLABench has too few tasks for a spread, but its answer-disjoint arm still changes sign across the three tasks that have it, $+0.139$ on the bloom tasks, whose target arrangement we construct, against -0.070 on `nth` and -0.183 on `texas`; these three are measured over all instruction strata rather than the vague-instruction stratum reported elsewhere. GPT-6 Astra and Opus 5.5 repeat this sign pattern (Figure 3). In every domain the lift changes sign across categories that the benchmark average reports as a single score.

3.3 Matched against mismatched retrieval

Most of the web lift comes from tasks where the retriever returned the task's own episode. Split this way, the matched lift exceeds the mismatched lift for all seven models (Table 1, Figure 2 last band). The sign of the mismatched lift is not consistent: it is negative for Fable 5, GPT-5.6, Opus 5.5 and Opus 4.7, zero for GPT-6 Astra, and positive for Kimi K3 and Opus 5, the two models with the largest overall lift. The matched tasks are not the easier ones: they score 64.9% at baseline against the mismatched half's 78.3%, pooled over the seven models, so the bias runs against the finding. We hypothesize that the gap reflects both retriever accuracy and the composition of the demonstration bank.

3.4 Transfer when the demonstration omits the answer

The web domain cannot construct answer-absent arms, so the control comes from the four offline physical domains, where a related-situation demonstration is built by hand. A related-situation demonstration raises the score in three of the four for Gemini 3.5-Flash (Figure 4): by $+0.119$ on Bench2Drive-VL, by $+0.225$ and $+0.195$ on VLABench's two disjoint sets of three demonstrations, and by $+0.042$ on Assembly101. A related-situation demonstration has no effect on IndEgo, whose

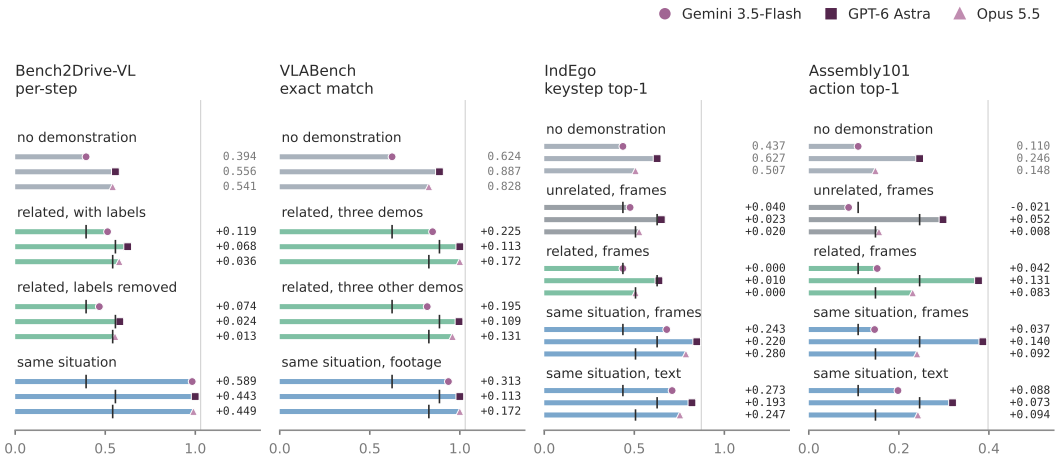


Figure 4: **Each domain's arm ladder for three models.** One row per arm and one marker per model. The bar color gives the arm: gray for no demonstration, dark gray for unrelated, green for related and blue for same situation. The black tick on each row is that model's no-demonstration score, so the gap from tick to marker is the lift. The right-hand column prints the no-demonstration score on its own rows and the lift over that model's floor on every other row. Every VLABench arm sends three demonstrations as footage. Panels use each domain's own metric and are not comparable; Table 3 gives the numbers.

related arm carries the item's answer only 5.0% of the time against 97.7% for its same-situation arm (Table 2). Once the items whose demonstration holds the answer are removed, or excluded by construction, a positive lift survives in two of the four domains (VLABench and Bench2Drive-VL, Table 3, appendix). On Bench2Drive-VL the lift on the $2{,}983$ answer-absent items is $+0.096$ ($[+0.060, +0.138]$) against $+0.209$ where the answer was present, and it grows slightly with how many of the demonstration's six answer positions match the item's (Table 7); for all three models the lift peaks at 5 of 6 matching positions, not at 6. On those same items, a same-situation demonstration adds $+0.654$ (0.325 to 0.979), so the surviving fraction from related-situation demonstrations is $\rho = 0.096/0.654 = 15\%$, held down by a nearly saturated ceiling: the same-situation arm feeds the item's own future window at 100% answer availability, so it is close to a copy target. Assembly101 retains -0.004 ($[-0.043, +0.034]$) on its 245 answer-absent items, a bounded null, and all of its $+0.042$ comes from the 275 answer-present items. VLABench needs no such split: both of its related arms, $+0.225$ and $+0.195$, are answer-absent by construction; the two draws share no demonstrations and differ by 0.030. Their ceiling sends the same three demonstrations with the answer restored and scores 0.937, which gives surviving fractions of $\rho = 0.225/0.313 = 72\%$ and $0.195/0.313 = 62\%$.

GPT-6 Astra and Opus 5.5 repeat the pattern (Table 3). With the answer absent, Bench2Drive-VL keeps $+0.068$ $[+0.045, +0.093]$ and $+0.026$ $[+0.011, +0.041]$ ($\rho = 13\%$ and 5%), Assembly101's interval covers zero for all three models, and IndEgo's related arm adds $+0.010$ and $+0.000$. Both reach 1.000 on the VLABench ceiling, which caps ρ at 100%, so VLABench cannot rank them.

3.5 Copying the demonstration

Copying produces short rollouts that are correct when the demonstration holds the item's answer and wrong otherwise; the traces show both. Given its own episode, Opus 4.7 emits the final answer of some VisualWebArena tasks in one step, skipping the intermediate ones, and emits a wrong one where the demonstration's answer is wrong (Appendix D). Another indicator of copying is that rescues concentrate on matched retrievals, 72 to 93% across the seven models. Only four of Opus 4.7's 11 regressions received the item's own episode, so a same-situation demonstration can hurt for reasons other than copying. On Assembly101, a different-family demonstration lowers Gemini's score below its no-demonstration floor, 0.088 against 0.110; it does not do so for GPT-6 Astra or Opus 5.5 (0.298 against 0.246 and 0.156 against 0.148).

Counting the two shortcuts. Positional copying and verbatim answer copying can both be counted per item (Table 9, Figure 10, appendix). Positional copying answers with the demonstration's step at the item's own index, which is wrong as soon as the two runs diverge in order. Its rate rises from 5.9 to 19.4% on the 170 IndEgo items where that step is wrong and from 4.9 to 10.5% on the 285 such Assembly101 items, against 0.5% for an unrelated demonstration. Verbatim copying of the demonstration's answer rises from 4.2 to 11.2% on the 2,983 Bench2Drive-VL items where that answer is wrong for the item, while exact-match accuracy rises from 12.2 to 16.9% on those same items, so the same demonstration supplies both a usable method and a plausible wrong answer. Both shortcuts persist in GPT-6 Astra and Opus 5.5 (Figure 10): verbatim copying rises from 6.4 to 20.4% and from 6.0 to 11.8%, and positional copying from 10.0 to 17.6% and from 13.5 to 20.0%.

4 Discussion

Agents need a reliability map. Every result above is invisible in the benchmark average: the dispersion across a benchmark's own categories, the sign reversal inside Opus 4.7's lift, the refusal component of Opus 5's lift, and the answer-availability split. Two models with the same success rate can differ on all four, so deploying an agent requires a *reliability map* (Gans, 2026) rather than an average alone.

Demonstration conditioning produces transfer as well as answer copying. With the answer removed, the lift survives on driving and on VLABench but not on Assembly101 or IndEgo, for all three models. A demonstration therefore does more than supply the answer, though not in every domain. The two components respond differently to better retrieval. A bank built from an agent's own history returns the item's own episode more often as retrieval improves, so each improvement moves the system toward its answer-present ceiling. Such a system then reports a gain on tasks the agent has already seen, which says little about the ones it has not. Conditioning on answer availability a_i removes answer copying but not the positional and vocabulary shortcuts (Appendix D), which exist even when the demonstration does not hold the answer. Verbatim copying depends on the demonstration's text labels: with the labels removed, its rate returns to within 1.7pp of the no-demonstration rate for all three models. The score gap between text and frames does not generalize: with frames, GPT-6 Astra and Opus 5.5 score higher or within 0.002 of text (Table 6). Refusal override is a separate effect of demonstration conditioning: on six of Opus 5's rescued tasks, the baseline declined to act rather than failing at the task.

Stronger models gain less on the web. On VisualWebArena the lift falls as the baseline rises, and the best-to-worst gap shrinks from 37 to 21 tasks (Table 1). On Bench2Drive-VL the stronger models also gain less from a related demonstration, $+0.068$ and $+0.036$ against Gemini's $+0.119$, but on Assembly101 GPT-6 Astra starts highest and gains most, $+0.131$ (Table 3). The jagged profile remains inside every model (Figures 2 and 3).

Demonstration-following and base model competence show different jagged patterns. The two dispersions are comparable in size but do not coincide (§3.2). A model's competence profile is no guide to where a demonstration will help it: on VisualWebArena, the website that gains most under one model loses most under another. Task difficulty is also a weak predictor: only one of the three testable category groupings in the four offline physical domains shows a meaningful correlation (§3.2), and that correlation changes with how difficulty and the task are defined. A lift measured on one task category does not carry to the next, which is consistent with the finding that the agent-by-task interaction dominates the agent main effect (Srinivasan, 2026).

5 Related work

Measuring agent competence and its dispersion. Work on agent reliability and holistic evaluation reports dispersion at the level of benchmark averages, for text agents on digital tasks (Rabanser et al., 2026; Kapoor et al., 2025). The most direct study finds that the agent-by-task interaction (7–23%) dwarfs the agent main effect (under 3%) (Srinivasan, 2026), and Gans (2026) models the same unevenness as an adoption problem in which scaling raises the average without removing the jaggedness and increases the need for an agent reliability map. We measure competence inside five

multimodal domains, over each domain's own categories, with each model held fixed, and again under the demonstration intervention.

In-context demonstrations and procedural memory. Few-shot prompting (Brown et al., 2020; Alayrac et al., 2022), its agentic descendants (Ding et al., 2026), and memory systems that surface a prior episode (Shinn et al., 2023; Wang et al., 2023; Luo et al., 2026; Jiang et al., 2026) report aggregate lift, usually on one benchmark and usually text-centric, and none records per item whether the demonstration contained the answer. The nearest work isolates procedural-memory retrieval from execution on ALFWorld (Kohar & Krishnan, 2025), varying the retriever; we vary the demonstration's relation to the item instead, across five domains.

6 Conclusion

We ran one controlled pass over five multimodal agentic domains, holding the prompt fixed and changing only the demonstration block. Baseline competence is jagged inside every benchmark's own categories, not only across benchmarks. A demonstration helps in all five domains and for all seven web models, with a jagged profile of its own. Stronger models gain less on the web tasks, and every model's profile stays jagged. Splitting the lift on answer availability and inspecting individual rollouts show procedural transfer alongside final-answer copying.

7 Limitations

We prioritize a preliminary breadth-first exploration over exhaustive coverage: four of five domains admit a constructed related-situation arm, seven models run on VisualWebArena and three models on the other four domains, and only GPT-6 Astra and Opus 5.5 cross both axes. Each domain's sample size is limited to the items with an available demonstration trajectory. Every arm is a single pass, and the repeat-run noise floor is read off one model at a shorter step budget (Appendix A). The IndEgo anticipation task is one we built, and Gemini's no-demonstration arm scores below a positional copy of the demonstration; Bench2Drive-VL's same-situation ceiling feeds the item's own future window, so that ceiling is effectively a copy target. The seven web models come from three vendors, six from Anthropic or OpenAI, so the baseline–lift correlation describes this set, not all models. VLABench saturates for GPT-6 Astra and Opus 5.5. Decoding cannot be equalized: GPT-6 Astra rejects a temperature setting, Opus 5.5 runs with thinking always on, and Gemini runs at temperature 0. Every task scores symbolic output, so continuous action is left to future work.

AI use statement

In this work, we used generative AI tools for experiment code, discovery of certain related work, drafting certain sections, and polishing the writing. We have not used generative AI tools for results verification and analysis drafting. We reviewed all AI-assisted work. We reviewed and verified LLM-generated code for correctness. We take responsibility for the final content of this work, including text, claims, or artifacts produced with the aid of generative AI.

Ethics statement

All five domains are public benchmarks with no new data collected from people. IndEgo and Assembly101 contain recordings of people performing tasks, used as released by their authors.

Reproducibility statement

All results come from inference against pinned API versions (Anthropic, 2026b;c;a; OpenAI, 2026a; Kimi, 2026; Google DeepMind, 2026; OpenAI, 2026b; Anthropic, 2026d), with no training or fine-tuning. Appendix A gives each model's id, decoding and retrieval settings and the interval method; Appendix B gives each domain's setup and deviations from its protocol; Appendix C gives every prompt. Every arm is a single pass, read against two repeat passes of one model (§3.1).

REFERENCES

Jean-Baptiste Alayrac et al. Flamingo: a visual language model for few-shot learning. *arXiv preprint arXiv:2204.14198*, 2022. doi: 10.48550/arXiv.2204.14198.

Anthropic. Claude fable 5 and mythos 5. https://www.anthropic.com/news/claude-fable-5-mythos-5, 2026a. Accessed 30 August 2026.

Anthropic. Introducing claude opus 4.7. https://www.anthropic.com/news/claude-opus-4-7, 2026b. Accessed 30 August 2026.

Anthropic. Introducing claude opus 5. https://www.anthropic.com/research/claude-opus-5, 2026c. Accessed 30 August 2026.

Anthropic. Claude Opus 5.5 system card. https://www-cdn.anthropic.com/fc1b44717c85dc068bc6ba5024219938094694bd/Claude%20Opus%205.5%20System%20Card.pdf, 2026d. Dated 22 September 2026. Accessed 25 September 2026.

Tom B. Brown et al. Language models are few-shot learners. *arXiv preprint arXiv:2005.14165*, 2020. doi: 10.48550/arXiv.2005.14165.

Vivek Chavan et al. Indego: A dataset of industrial scenarios and collaborative work for egocentric assistants. *arXiv preprint arXiv:2511.19684*, 2025. doi: 10.48550/arXiv.2511.19684.

Zihan Ding et al. Ui-mate: Advancing open-weight foundation gui agents with in-context demonstrations. *arXiv preprint arXiv:2608.15930*, 2026. doi: 10.48550/arXiv.2608.15930.

Joshua Gans. A model of artificial jagged intelligence. *arXiv preprint arXiv:2601.07573*, 2026. doi: 10.48550/arXiv.2601.07573.

Google DeepMind. Gemini 3.5 flash — model card. https://deepmind.google/models/model-cards/gemini-3-5-flash/, 2026. Accessed 30 August 2026.

Xiaosong Jia, Zhenjie Yang, Qifeng Li, Zhiyuan Zhang, and Junchi Yan. Bench2drive: Towards multi-ability benchmarking of closed-loop end-to-end autonomous driving. *arXiv preprint arXiv:2406.03877*, 2024. doi: 10.48550/arXiv.2406.03877.

Dongming Jiang et al. Anatomy of agentic memory: Taxonomy and empirical analysis of evaluation and system limitations. *arXiv preprint arXiv:2602.19320*, 2026. doi: 10.48550/arXiv.2602.19320.

Sayash Kapoor et al. Holistic agent leaderboard: The missing infrastructure for ai agent evaluation. *arXiv preprint arXiv:2510.11977*, 2025. doi: 10.48550/arXiv.2510.11977.

Kimi. Kimi k3: 2.8t open model for coding and knowledge work. https://www.kimi.ai/ai-models/kimi-k3, 2026. Accessed 30 August 2026.

Jing Yu Koh et al. Visualwebarena: Evaluating multimodal agents on realistic visual web tasks. *arXiv preprint arXiv:2401.13649*, 2024. doi: 10.48550/arXiv.2401.13649.

Ishant Kohar and Aswanth Krishnan. A benchmark for procedural memory retrieval in language agents. *arXiv preprint arXiv:2511.21730*, 2025.

Mingxin Li et al. Qwen3-vl-embedding and qwen3-vl-reranker: A unified framework for state-of-the-art multimodal retrieval and ranking. *arXiv preprint arXiv:2601.04720*, 2026. doi: 10.48550/arXiv.2601.04720.

Jinghao Luo et al. From storage to experience: A survey on the evolution of llm agent memory mechanisms. *arXiv preprint arXiv:2605.06716*, 2026. doi: 10.48550/arXiv.2605.06716.

OpenAI. Gpt-5.6: Frontier intelligence that scales with your ambition. https://openai.com/index/gpt-5-6/, 2026a. Accessed 30 August 2026.

OpenAI. GPT-6 Astra system card. https://deploymentsafety.openai.com/gpt-6-astra/gpt-6-astra.pdf, 2026b. Dated 3 September 2026. Accessed 25 September 2026.

Stephan Rabanser, Sayash Kapoor, Peter Kirgis, Kangheng Liu, Saiteja Utpala, and Arvind Narayanan. Towards a science of ai agent reliability. *arXiv preprint arXiv:2602.16666*, 2026. doi: 10.48550/arXiv.2602.16666.

Fadime Sener et al. Assembly101: A large-scale multi-view video dataset for understanding procedural activities. *arXiv preprint arXiv:2203.14712*, 2022. doi: 10.48550/arXiv.2203.14712.

Noah Shinn, Federico Cassano, Edward Berman, Ashwin Gopinath, Karthik Narasimhan, and Shunyu Yao. Reflexion: Language agents with verbal reinforcement learning. *arXiv preprint arXiv:2303.11366*, 2023. doi: 10.48550/arXiv.2303.11366.

Vasundra Srinivasan. Deployment decision reliability: A generalizability-theory framework for sizing long-horizon agent evaluations. *arXiv preprint arXiv:2608.11323*, 2026.

Xiaoyu Tian et al. Drivevlm: The convergence of autonomous driving and large vision-language models. *arXiv preprint arXiv:2402.12289*, 2024. doi: 10.48550/arXiv.2402.12289.

Guanzhi Wang et al. Voyager: An open-ended embodied agent with large language models. *arXiv preprint arXiv:2305.16291*, 2023. doi: 10.48550/arXiv.2305.16291.

Q. Wang et al. Vimrag: Navigating massive visual context in retrieval-augmented generation via multimodal memory graph. *arXiv preprint*, apr 2026. doi: 10.48550/arXiv.2602.12735. arXiv:2602.12735.

G. Zhang, Y. Ling, F. Meng, K. Wang, and S. Yan. Mem-w: Latent memory-native gui agents. *arXiv preprint*, may 2026. doi: 10.48550/arXiv.2605.09317. arXiv:2605.09317.

Shiduo Zhang et al. Vlabench: A large-scale benchmark for language-conditioned robotics manipulation with long-horizon reasoning tasks. *arXiv preprint arXiv:2412.18194*, 2024. doi: 10.48550/arXiv.2412.18194.

A METHOD DETAILS

Algorithm 1 One controlled pass over a domain. The demonstration block is the only thing that varies.

Require: items $\mathcal{I}$, arms $\mathcal{A}$ (floor $\emptyset$, related, same-situation, unrelated), the benchmark's scorer s, its harness H

for each item $i \in \mathcal{I}$ **do**
  for each arm $a \in \mathcal{A}$ the domain supports **do**
    $d \leftarrow \textsc{SelectDemo}(i, a)$ ▷ fixed rule, or top-1 retrieval, never the score
    $p \leftarrow \textsc{Render}(i, d)$ ▷ identical to the floor outside the block
    $y_{i,a} \leftarrow H(p)$; $s_{i,a} \leftarrow s(y_{i,a}, y_i^\star)$
    $a_i^{(a)} \leftarrow \textsc{AnswerAvailable}(d, y_i^\star)$ ▷ domain-specific test
  end for
end for
$\Delta_a \leftarrow \text{mean}_i \left[s_{i,a} - s_{i,\emptyset} \right]$ ▷ paired, clustered at the dependence unit
$\Delta_a \mid a_i{=}1, \quad \Delta_a \mid a_i{=}0$ ▷ the same lift, split by what the demonstration held

Table 2: **How each domain builds the four arms, and what answer availability turns out to be.** The tier names describe how a demonstration was *selected*; a_i is whether it turns out to contain the answer, recorded per item and never inferred from the tier. The two do not track each other across domains: Assembly101's unrelated arm carries the answer more often than IndEgo's related arm, because selecting a demonstration by situation does not prevent it from containing the item's answer. VLABench is the exception: its arms are built answer-free or answer-bearing by construction, so availability is set rather than measured.

Domain	Tier	Arm as the harness builds it	Share with $a_i{=}1$
VisualWebArena	same-situation	top-1 returned the item's own episode, the *matched* subset	65.0–66.7% matched, by self-hit flag
	related	top-1 returned a different task's episode, *mismatched*	
	unrelated	—	
Bench2Drive-VL	same-situation	the item's own future window	100%
	related	feature-matched demonstration from a different route	20.4%, graded 0 to 6 by matching positions (Table 7)
	unrelated	—	
Assembly101	same-situation	same toy, different session, as text or frames	64.2%
	related	same family, different toy	52.9%
	unrelated	different family	31.2%
IndEgo	same-situation	same task, different worker, as text or frames	97.7%
	related	different task, same official scenario	5.0%
	unrelated	any different task	2.7%
VLABench	same-situation	three demonstrations as footage, one carrying the item's answer	100% by construction
	related	three neutralized demonstrations, every answer different from the item's; two disjoint sets of three	0% by construction
	unrelated	—	

Intervals and tests. Intervals are 95% percentile bootstrap over 10,000 resamples of paired per-item differences, clustered at route (Bench2Drive-VL, 86), recording (Assembly101 27, IndEgo 268) or item (VLABench). The sign test over models is one-sided on the seven paired lifts, $p = (1/2)^7 = 0.008$. Where a category's difficulty is related to its lift, the baseline is estimated on that category's even-indexed items and the lift on the odd-indexed ones; for websites, which are too small to halve, the difficulty proxy is the mean of the other six models' baselines there, which does not predict the lift ($r = 0.27$, $p = 0.23$).

Run-to-run variation. The noise floor in §3.1 comes from two passes of Opus 4.7 at a matched configuration and a 50-step budget; one of them lost 45 of 177 cells to transport errors. Restricted to the 129 cells clean in all four arms they give baselines of 56.8% and 59.8%, demonstration arms

Table 3: **Three models across the four offline physical domains.** Scores use each domain's own metric and are not comparable between domains. Δ is the paired lift over the no-demonstration arm. The related arm's lift is split by whether its demonstration held the item's answer (a_i=1 or a_i=0); the bold $\Delta \mid a_i{=}0$ is the transfer estimate, with its 95% interval, and ρ divides it by the same-situation lift on the same items. VLABench rows use the vague-instruction stratum; their related arms hold no answer by construction, and their $\Delta \mid a_i{=}1$ column shows the same-situation lift. A dash marks a split or ratio with no lift to split or normalize. The 9 Bench2Drive-VL items whose demonstration differs from Gemini's are left out of the a_i split (Appendix B).

Domain	n	none	related	same-sit.	Δ related	$\Delta \mid a_i{=}1$	$\Delta \mid a_i{=}0$	ρ
Gemini 3.5-Flash								
Bench2Drive-VL	3,746	0.394	0.513	0.983	+0.119	+0.209	**+0.096** [+0.060, +0.138]	15%
VLABench three demos	218	0.624	0.849	0.937	+0.225	+0.312	**+0.225**	72%
VLABench three others	221	0.624	0.819	0.937	+0.195	+0.312	**+0.195**	62%
IndEgo	300	0.437	0.437	0.710	+0.000	—	—	—
Assembly101	520	0.110	0.152	0.198	+0.042	+0.084	**−0.004** [−0.043, +0.034]	—
GPT-6 Astra								
Bench2Drive-VL	3,746	0.556	0.625	1.000	+0.068	+0.072	**+0.068** [+0.045, +0.093]	13%
VLABench three demos	221	0.887	1.000	1.000	+0.113	+0.113	**+0.113**	100%
VLABench three others	221	0.887	0.995	1.000	+0.109	+0.113	**+0.109**	96%
IndEgo	300	0.627	0.637	0.820	+0.010	—	—	—
Assembly101	520	0.246	0.377	0.319	+0.131	+0.225	**+0.024** [−0.017, +0.063]	—
Opus 5.5								
Bench2Drive-VL	3,746	0.541	0.578	0.990	+0.036	+0.077	**+0.026** [+0.011, +0.041]	5%
VLABench three demos	221	0.828	1.000	1.000	+0.172	+0.172	**+0.172**	100%
VLABench three others	221	0.828	0.959	1.000	+0.131	+0.172	**+0.131**	76%
IndEgo	300	0.507	0.507	0.753	+0.000	—	—	—
Assembly101	520	0.148	0.231	0.242	+0.083	+0.178	**−0.024** [−0.065, +0.018]	—

of 62.8% and 65.9%, and paired lifts of +6.21pp and +6.20pp; per site the lifts differ by 1.2pp on shopping (n=85) and 2.3pp on reddit (n=44). The step budget differs from the 100 used throughout, so these passes indicate the noise of the protocol rather than directly bounding the reported values.

Refusal classification (§3.2). A baseline cell is a *refusal* when its terminal answer carries an explicit decline, matched on text normalized for escapes and apostrophes ("outside what I can do", "I declined", "I can't post / send / submit", "on a user's behalf"), and *stopped-short* when it answered but never performed the required action, such as adding to a cart without placing the order. Of Opus 5's 27 rescued tasks, six are refusals (`31`, `323`, `374`, `377`, `381`, `386`) and three stopped-short (`705`, `758`, `805`).

Model configuration. Table 4 gives every model's pinned id and decoding settings. Beyond what §2.2 lists as shared, the web rows also share 1280 × 720 JPEG (q= 95) screenshots and actions that normalize to identical BrowserGym `coord`/`nav`/`chat` calls; only the vendor-native tool surface differs. Reasoning is on for every model at whatever setting its vendor exposes: `effort=high` for GPT-5.6, the API default for Opus 4.7 and Opus 5, adaptive and always on for Fable 5, and thinking on for Kimi K3, which offers no effort setting. GPT-6 Astra runs at `effort=medium` on every domain, and Opus 5.5 at its default, `medium`, with thinking always on. Neither sends a temperature. On the offline physical domains, the demonstration arms go through each vendor's batch API; a batch request carries the same body as a synchronous one. Opus 5.5's JSON schema cannot fix a list length, so only the prompt enforces the lengths of its top-5 and six-step lists. All of its Bench2Drive-VL and IndEgo answers have the correct length; on Assembly101, 4 of 2,600 answers do not and 3 stop at the output cap. Top-1 reads the first entry in every case.

Retrieval configuration. Qwen3-VL-Embedding-8B (Li et al., 2026) in `bfloat16`, emitting L_2-normalized float32 vectors of dimension 4096 at a 65,536-token context, with the query-side instruction *Retrieve GUI demonstration trajectories matching this web task.* Search is exact, a FAISS `IndexFlatIP` over the 177-trajectory bank at k=1 with no similarity floor, so every task receives a demonstration. The index is built once and never tuned against the score. Top-1 is not always the

Table 4: **Model configuration, one row per model.** The upper block is the closed-loop web sweep, the lower the four offline physical domains. `maxOutputTokens` is the whole output budget on Gemini, thinking included, so VLABench raises it: its plans are the only outputs that need long reasoning: 86 of its items spend more than 8,192 thinking tokens and one spends 62,915, whereas Bench2Drive-VL's longest answer is 186 tokens and IndEgo's is 52. Every VLABench arm we report ran at 65,536, so no contrast crosses the two settings.

Model	Pinned id	API surface and tools	Reasoning	Decoding
Fable 5	`claude-fable-5`	Anthropic Messages; flat function tools	adaptive, always on (high)	`max_tokens` 8,192
Opus 4.7	`claude-opus-4-7`	Anthropic Messages beta `computer-use-2025-11-24`; `computer_20251124` plus `goto`, `go_back`, `go_forward`, `stop`	extended thinking, API default (high)	`max_tokens` 4,096
Opus 5	`claude-opus-5`	as Opus 4.7, same runner	API default (high)	`max_tokens` 4,096
GPT-5.6	`gpt-5.6-sol`	OpenAI Responses; flat function tools, `previous_response_id`	`effort=high`, `summary=auto`	`truncation=auto`
Kimi K3	`kimi-k3`	Moonshot chat completions; flat function tools, `tool_choice =required`	thinking on, no effort knob	`max_tokens` 4,096, `top_p` 0.95, 8 retries
GPT-6 Astra	`gpt-6-astra`	as GPT-5.6, same runner	`effort=medium`	`truncation=auto`
Opus 5.5	`claude-opus-5-5`	Anthropic Messages; `computer_toolset_20260801` (zoom off) plus `goto`, `go_back`, `go_forward`, `stop`; one turn may batch actions, each one step	adaptive, always on (medium)	`max_tokens` 4,096
Gemini 3.5-Flash	`gemini-3.5-flash`	`v1beta:generateContent`; JSON response schema (constrained decoding)	default	temperature 0.0, `maxOutputTokens` 8,192 and 65,536 on VLABench, 60 s timeout, backoff 2–32 s
GPT-6 Astra	`gpt-6-astra`	OpenAI Responses and Batch; strict JSON schema, list lengths enforced	`effort=medium`	no temperature (rejected), `max_output_tokens` 8,192 and 65,536 on VLABench
Opus 5.5	`claude-opus-5-5`	Anthropic Messages and Message Batches; JSON schema without list-length bounds	adaptive, always on (medium)	API-default temperature, `max_tokens` 8,192 and 65,536 on VLABench

same: across the seven runs the self-hit count ranges from 115 to 118, and the self-hit flags of any two runs differ on up to 6 tasks.

B TASK DETAILS

Table 5: Per-domain setup. Seven models run VisualWebArena and three run each offline physical domain.

Domain	Task	Model(s)	Metric	n	Cluster	Demonstration arms
VisualWebArena	closed-loop web task, 177-task subset	Fable 5, GPT-5.6, Opus 4.7, Opus 5, Kimi K3, GPT-6 Astra, Opus 5.5	task success	177	task	no-demo, retrieved (k=1)
VLABench	VLM track, vague-instruction stratum	Gemini 3.5-Flash, GPT-6 Astra, Opus 5.5	exact match	221	item	no-demo, same-situation footage, two answer-disjoint related sets; every demonstration arm sends three
Assembly101	coarse action anticipation	Gemini 3.5-Flash, GPT-6 Astra, Opus 5.5	action top-1	520	recording (27)	no-demo, same-situation (text, frames), related, unrelated
IndEgo	next-keystep anticipation (new task)	Gemini 3.5-Flash, GPT-6 Astra, Opus 5.5	keystep top-1	300	recording (268)	no-demo, same-situation (text, frames), related, unrelated, positional copy
Bench2Drive-VL	question 50, two-turn chain	Gemini 3.5-Flash, GPT-6 Astra, Opus 5.5	per-step, weighted-F1	3,746	route (86)	no-demo, related, related with labels removed, same-situation

The VisualWebArena subset. The 177 tasks are the benchmark's own test split intersected with available human trajectories, minus zero-step episodes: classifieds 48, reddit 44, shopping 85. Nothing is excluded before the sweep, and every number is success over all 177. By evaluation type the split is 79 `program_html`, 55 `url_match`, 39 `string_match` and 4 combined, so 76% require landing on a specific page.

Per-domain deviations. No row of Table 3 should be read against a published score, since each domain departs from its protocol in one respect. IndEgo ships no anticipation benchmark, so we built the one described below.

Bench2Drive-VL. Question 50, "The ego vehicle is driving at the speed of {speed} m/s, and it wants to {nav command}.", with pixel-coordinate object tags stripped from the demonstration text, since its camera coordinates point at nothing in the item frame. This is not the closed-loop driving score.

Assembly101. The official coarse anticipation protocol ($\alpha = 1$s, 6s horizon, $\geq$ 2s history, top-k), labels and splits. We predict coarse labels rather than fine ones, since fine labels describe hand micro-manipulations that no other toy can inform.

VLABench. The VLM track with upstream prompt assembly, restricted to the stratum where the instruction does not name its target. Gemini's answer-disjoint arm on `texas` covers 15 of 44 items; GPT-6 Astra and Opus 5.5 cover all 44.

The IndEgo anticipation task. The model predicts the next keystep from the goal, the realized history and 14 frames, with the task's own checklist withheld, so that a demonstration is the only source of it. The label space is the 112 keysteps in the union of all task checklists, shown sorted; timing follows Assembly101 ($\alpha = 1.0$s gap, 6.0s window); the answer is five ranked steps under a server-enforced enum. We fix a set of 300 items from 5,875 prediction points over 24 tasks and count any valid next step as correct, since about 20% of points admit more than one valid next step. Three reference points: the majority label scores 0.037, uniform choice 0.009, and a positional copy 0.495 over the 291 items where such a step exists.

Table 6: **Episode modality structure.** The three rows use different designs. Assembly101 and IndEgo swap the modality outright, replacing the written action list with one frame per demonstration step; both point the same way, and for Gemini 3.5-Flash neither difference's interval excludes zero: Assembly101's $+0.051$ and IndEgo's $+0.025$ $[-0.017, +0.066]$ over 268 recordings. Bench2Drive-VL instead keeps the frames in both arms and removes only the demonstration's labels, so it bounds what the text labels add rather than comparing two encodings; this difference excludes zero, $+0.046$ $[+0.022, +0.074]$ over 86 routes. For GPT-6 Astra and Opus 5.5 the modality swap does not point the same way: with frames they score higher or within 0.002 of text. Their label ablation stays above zero, $+0.044$ $[+0.030, +0.059]$ and $+0.023$ $[+0.013, +0.035]$.

Domain	Design	Gemini	GPT-6 Astra with text / without	Opus 5.5
Assembly101	modality swap, Δ top-1	$+0.088$ / $+0.037$	$+0.073$ / $+0.140$	$+0.094$ / $+0.092$
IndEgo	modality swap, top-1	0.710 / 0.680	0.820 / 0.847	0.753 / 0.787
Bench2Drive-VL	labels removed, per-step	0.513 / 0.467	0.625 / 0.581	0.578 / 0.554

Compute and artifacts. The closed-loop sweeps take around 150 agent-hours over 1,770 rollouts for Fable 5, GPT-5.6, Opus 4.7, Opus 5 and Kimi K3, plus 708 rollouts for GPT-6 Astra and Opus 5.5; each item in the four offline physical domains takes one forward pass, more than 22,000 calls for Gemini and 21,426 for the other two models. Every run is stored as one row, so the splits of §3.4 recompute from the per-run files.

Closed-loop rollout: task id, arm, score, step count, wall time, error field, retrieval self-hit flag.

Offline physical-domain item: arm, item id, prediction, ground truth, per-step correctness, demonstration id, answer-availability flag.

Table 7: **Retrieval recall, and task success lift by answer overlap.** *Left:* top-1 recall of the item's own episode on VisualWebArena; the retriever never sees the model under test. *Right:* Bench2Drive-VL's related arm split by how many of the demonstration's six answer positions match the item's. The lift rises with overlap but not monotonically; for all three models it peaks at 5 of 6. Item counts are Gemini's; the other two models differ by at most 3 per row (Appendix B).

VisualWebArena, top-1 recall			Bench2Drive-VL, answer overlap, Δ per-step				
Model	own episode	share	matching steps	items	Gemini	Astra	Opus 5.5
Fable 5	118/177	66.7%	0 of 6	811	+0.025	+0.026	−0.034
GPT-5.6	118/177	66.7%	1 of 6	534	+0.135	+0.007	−0.027
Opus 4.7	118/177	66.7%	2 of 6	501	+0.005	+0.032	+0.012
Opus 5	117/177	66.1%	3 of 6	472	+0.158	+0.067	+0.062
Kimi K3	116/177	65.5%	4 of 6	352	+0.124	+0.172	+0.116
GPT-6 Astra	118/177	66.7%	5 of 6	313	+0.238	+0.223	+0.141
Opus 5.5	115/177	65.0%	6 of 6	763	+0.209	+0.072	+0.077

Table 8: **Does a category's difficulty predict how much a demonstration helps?** Difficulty is the category's own no-demonstration score. *Naive* correlates it against the lift, which shares a term and is biased negative: on simulated data with an identical lift everywhere it returns -0.18 against a true zero. *Held-out* estimates baseline and lift from disjoint halves of a category's items, removing the bias but adding noise, so it is informative only where the halves agree. *Reliability* is how much they do. Every row correlates difficulty against the lift of that domain's *related-situation* arm, so each row tests the same contrast. IndEgo's 21 types are the ones holding at least four items, the fewest a split can use; Figure 3 plots all 24. Only Bench2Drive-VL clears both bars, for all three models; VLABench's per-task rows carry one arm score each, too few to split.

Domain	Categories	Model	Reliability	Naive r	Held-out r	p
Bench2Drive-VL	86 routes	Gemini	0.90	−0.47	**−0.39**	0.0002
		GPT-6 Astra	0.95	−0.42	**−0.31**	0.003
		Opus 5.5	0.96	−0.37	**−0.26**	0.016
IndEgo	21 task types	Gemini	0.47	−0.21	−0.35	0.124
		GPT-6 Astra	0.34	−0.10	+0.01	0.983
		Opus 5.5	0.28	−0.36	−0.09	0.698
Assembly101	14 toy types	Gemini	0.01	−0.63	−0.22	0.442
		GPT-6 Astra	0.25	+0.04	−0.16	0.594
		Opus 5.5	−0.01	−0.41	−0.28	0.327
VLABench	5 tasks	Gemini	—	+0.13	—	—

C PROMPTS ACROSS ARMS

Every arm is the floor plus one inserted block. Excerpts come from the harness dumps.

VisualWebArena, all seven models.

```
You are an autonomous intelligent agent tasked with navigating a web browser. You
will be given web-based tasks. These tasks will be accomplished through the use of
specific actions you can issue.

Here's the information you'll have:
The user's objective: This is the task you're trying to complete.
The current web page screenshot: A screenshot of the webpage.
The current web page's URL: This is the page you're currently navigating.
The previous actions: This is the action you just performed.

The actions you can perform fall into several categories:
  click (x, y) / type [content] / hover (x, y) / press [key_comb]
  scroll (x, y) [down|up] / drag (x0, y0) (x1, y1) / wait [duration]
  goto [url] / go_back / go_forward
  stop [answer] ... (full argument descriptions elided)

Homepage:
If you want to visit other websites, check out the homepage at http://homepage.com.

To be successful, it is very important to follow the following rules:
```

```
1. You should only issue an action that is valid given the current observation.
2. You should only issue one action at a time.
3. You should reason step by step and then issue the next action.
4. Issue the `stop` action when you think you have achieved the objective.
```

Bench2Drive-VL.

```
Here is a demonstration of an expert driving through a similar situation.
Use it to inform your judgement.

1. t=12.5s, held 1.5s
   observation: [front camera frame]
   situation: <expert's situation text>
   behavior: FOLLOW_LANE, DECELERATE
... (one entry per command run)

The ego vehicle is driving at the speed of 7.4 m/s, and it wants to follow the road.
So far in this drive, you issued:
  t-1.5s DECELERATE, FOLLOW_LANE
  t-1.0s DECELERATE, FOLLOW_LANE

Provide the appropriate behavior for the ego vehicle, which consists of two keys:
the Direction key, which can be 'FOLLOW_LANE', 'CHANGE_LANE_LEFT', ... ; and the
Speed key, which can be 'KEEP', 'ACCELERATE', 'DECELERATE', or 'STOP'.
```

Assembly101.

```
You are watching a person assemble a toy vehicle.
The person is assembling a toy crane.

Below is a demonstration trajectory of an assembly task, performed by a human,
given as the ordered list of ACTIONS they performed. Use it as a reference for
assembly patterns. The task you must complete appears AFTER the demonstration.
  1. attach base 2. attach hook 3. attach arm
  4. attach turntable base 5. attach interior
  ... (15 steps)

The task you must complete:
The frames below are in chronological order. The next action begins about
1 second after the final frame.
  [11 frames]
```

VLABench.

```
[preamble, verbatim from the benchmark: the skill catalog -- pick, place,
 press, open_door, insert, pull, lift, pour -- each with its JSON call format,
 then the generation steps and the list of valid sub-skill sequences]

Task format note (structure only -- every id value must be read from YOUR
numbered input image): every valid plan for this task has exactly this shape,
with <ID> replaced by the correct mask number from your scene (ids are
integers): {"skill_sequence": [{"name": "pick", "params":
{"target_entity_name": <ID>}}, {"name": "insert", "params":
{"target_container_name": <ID>}}]}

                  ---- PAYLOAD begins ----
Retrieved trajectory 1: a solved episode of this task family from a DIFFERENT
scene (its object ids do not apply to your scene). Its instruction, starting
scene, and execution footage follow. Footage is two 4x4 contact sheets of 16
frames each (read left-to-right, top-to-bottom); each sheet is followed by the
robot end-effector states for its 16 frames.
Trajectory 1 instruction:
Please insert the tulip into the vase.
Trajectory 1 starting scene:
  [unmasked scene image]
  [4x4 contact sheet, frames 1-16]
  t=0: ee=[0.00, -0.17, 1.21], gripper=closed (16 such lines)
  [4x4 contact sheet, frames 17-32]
  t=17: ee=[0.02, -0.31, 1.04], gripper=open (16 such lines)
                  ---- PAYLOAD ends ----

Input picture
  [query scene image]
Input picture with numbered tags
  [query scene image with numbered mask overlay]
Language instruction:
Please Insert the proper flower into the container.
Please give the output skill sequence
```

IndEgo, no-demonstration arm.

```
You are watching an egocentric recording of a worker performing an industrial task.

The task you must complete:
Steps this worker has already completed in this recording, in order:
  1. bring tripod
  2. level tripod

The frames below are in chronological order. The next step begins about
1 second after the final frame.
  [14 frames]

Valid steps, choose exactly one:
  add battery / add ram / attach camera / ... (112 labels, sorted)
```

IndEgo, same-task arm, frames.

```
You are watching an egocentric recording of a worker performing an industrial task.

Below is a demonstration trajectory of an industrial task, performed by a human.
Each step shows a frame from the middle of that step and the STEP it is. Use it
as a reference for industrial patterns. The task you must complete appears AFTER
the demonstration.
  [frame] level tripod
  [frame] fasten legs
  [frame] level and fasten head

The task you must complete:
  ... (identical to the no-demonstration arm from here)
```

GPT-6 Astra and Opus 5.5. Both use the prompts above unchanged. What differs is the API surface around the prompt. On the web, GPT-6 Astra uses GPT-5.6's runner and flat function tools unchanged. Opus 5.5 rejects `computer_20251124`, so its runner declares the newer toolset and keeps the same navigation tools:

```
tools = [
  {"type": "computer_toolset_20260801", "configs": {"zoom": {"enabled": false}}},
  {"name": "goto", ...}, {"name": "go_back", ...},
  {"name": "go_forward", ...}, {"name": "stop", ...}] # as Opus 4.7

GPT-6 Astra / GPT-5.6 function tools:
  click, type, hover, press, scroll, drag, wait, screenshot,
  goto, go_back, go_forward, stop
```

On the offline physical domains the answer schema is Gemini's. GPT-6 Astra sends it as a strict `json_schema`, which enforces list lengths. Opus 5.5 sends it as `output_config.format` with `maxItems` removed and `minItems` kept only at 0 or 1, since its API rejects both bounds:

```
GPT-6 Astra: {"text": {"format": {"type": "json_schema", "strict": true,
              "schema": {... "steps": {"type": "array", "minItems": 5, "maxItems": 5}}}},
             "reasoning": {"effort": "medium"}}
Opus 5.5: {"output_config": {"effort": "medium", "format": {"type": "json_schema",
             "schema": {... "steps": {"type": "array"}}}}}
```

D ROLLOUT EXAMPLES

Each rollout figure is one task with the model held fixed: columns are the two arms, rows consecutive steps, and both arms open on the same page, so the row where they part is visible. Step 1 is the initial observation and carries no action, so step totals run one higher than an action count.

D.1 REUSE THAT HELPS THE ITEM

In both examples, the arms run identically until the step where the block's contents start to matter.

Table 9: **Where a demonstration helps, where it hurts, and the shortcut it invites, by domain.** The shortcut column reports the rate on the subset where the shortcut is identifiable, against the same quantity on those items' no-demonstration predictions. Magnitudes are on each domain's own metric. IndEgo's *positional-copy floor* is the 0.495 scored by answering with the demonstration's step at the item's own index, measured over the 291 items where such a step exists (Appendix B). For Gemini, its no-demonstration and related arms both score 0.437, below that floor (§3.1). Offline physical-domain cells read Gemini 3.5-Flash first, then GPT-6 Astra and Opus 5.5 in brackets.

Domain	Where it helps	Where it hurts	The shortcut it invites
VisualWebArena	every model, $+1.7$ to $+14.1$pp	mismatched retrieval, to -5.1pp; 2 to 11 solved tasks broken per model	copying the item's own answer: top-1 returns that episode on 115–118 of 177 tasks, where 72–93% of rescues occur
VLABench	vague-instruction stratum, $+0.225$ and $+0.195$ on two disjoint sets of three [$+0.113$, $+0.109$; $+0.172$, $+0.131$]	the same copy-proof arm costs -0.070 on `nth` and -0.183 on `texas` [-0.080, -0.007; -0.060, -0.114]	ordinal copying on `select_nth_largest_poker`: the copy-proof arm loses ground only where no demonstration shares the item's ordinal, with a consistent sign across three arms, with per-cell counts too small to quantify (Gemini only)
Assembly101	same-toy arm, $+0.088$ as text [$+0.073$; $+0.094$]	different-family arm scores below the no-demonstration floor (Gemini only)	vocabulary anchoring, $34.0\% \rightarrow 53.1\%$ [$51.3 \rightarrow 77.3$; $42.9 \rightarrow 66.0$]; positional copying, $4.9\% \rightarrow 10.5\%$ [$5.3 \rightarrow 9.8$; $5.6 \rightarrow 12.3$] on the 285 items where that step is wrong
IndEgo	same-task arm, $+0.24$ to $+0.27$ [$+0.19$ to $+0.22$; $+0.25$ to $+0.28$]	the related and unrelated arms both score below the positional-copy floor (Gemini only)	positional copying, $5.9\% \rightarrow 16.5\%$ as frames and 19.4% as text [$10.0 \rightarrow 15.3$, 17.6; $13.5 \rightarrow 20.0$, 20.0], on the 170 items where that step is wrong
Bench2Drive-VL	scene-matched arm, $+0.113$, and $+0.096$ with the answer absent [$+0.074$, $+0.068$; $+0.043$, $+0.026$]	16 of 86 routes score lower with a demonstration [16; 19]	copying the demonstration's own answer, $4.2\% \rightarrow 11.2\%$ as text and 5.9% as frames [$6.4 \rightarrow 20.4$, 7.4; $6.0 \rightarrow 11.8$, 6.7], on the $2{,}983$ items whose demonstration answer is wrong

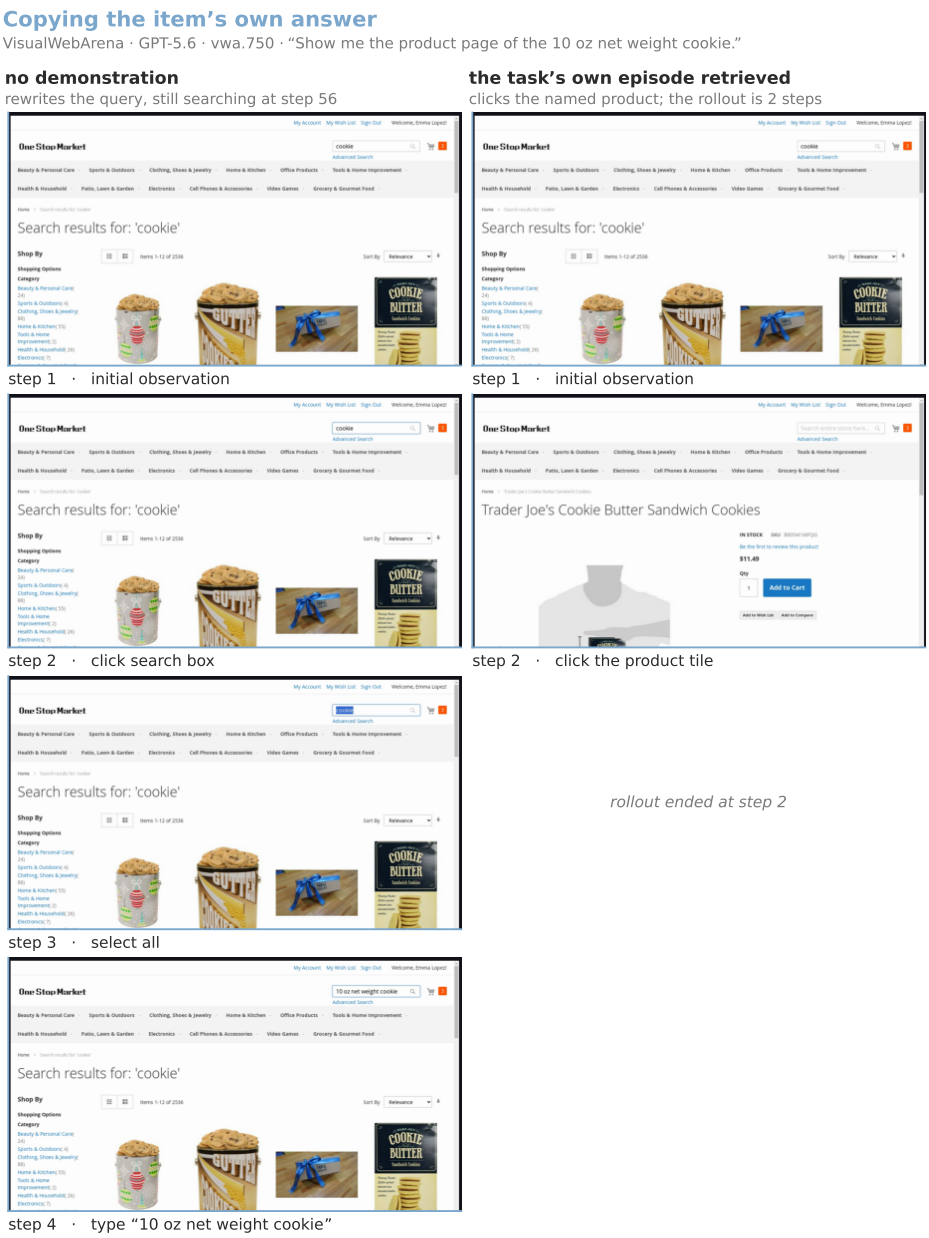


Figure 5: **Copying the item's own answer.** VisualWebArena, `vwa.750`, GPT-5.6. The episode retrieved is the task's own and names the target product. The arms part at step 2: with it, the model clicks that product from the opening page and the rollout is over; without it the same model reopens the search box, retypes the task string, and is still searching at step 56.

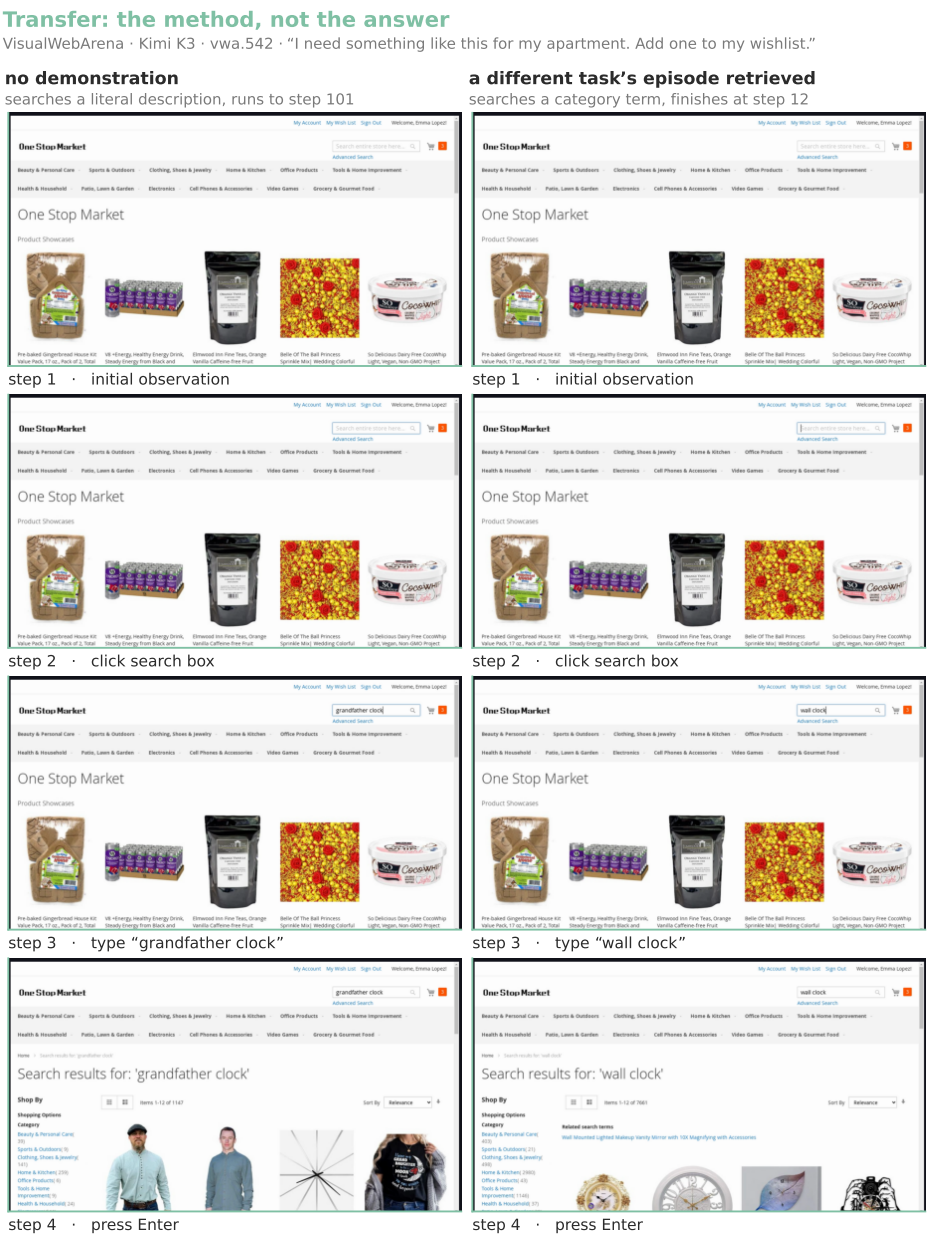


Figure 6: **Reusing the demonstration's method.** VisualWebArena, `vwa.542`, Kimi K3. The episode retrieved belongs to a different shopping task and cannot name this task's answer. Both arms take the same two actions and part at step 3 on the item alone: the baseline types the literal *grandfather clock*, reaches the results page at step 4, and runs out of step budget; the demonstration arm types the category term *wall clock* and finishes at step 12.

D.2 REUSE THAT MISLEADS

The offline physical domains store a prediction, a ground truth and a demonstration id per item, so three of the four failure modes below (answer copying, positional copying and vocabulary anchoring) are also counted over every item rather than only illustrated (Table 9).

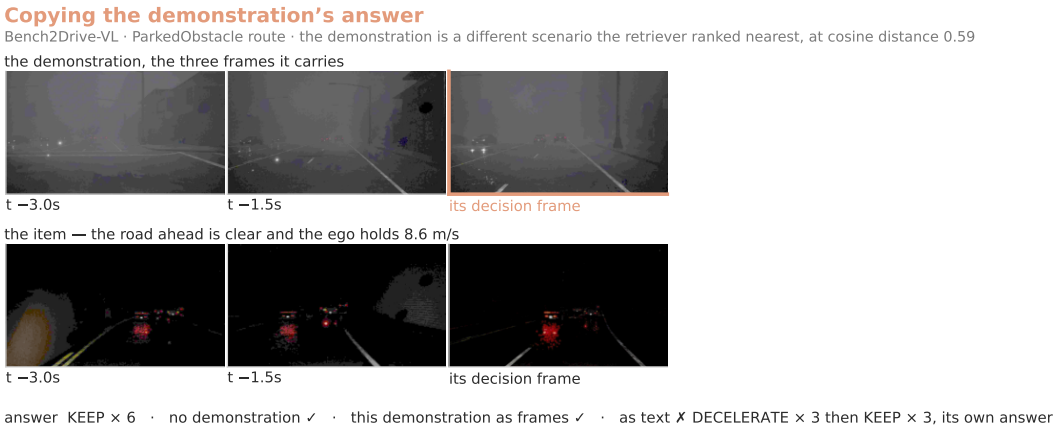


Figure 7: **Copying the demonstration's answer.** Bench2Drive-VL, item `ParkedObstacle_Town03_Route103_Weather25/00145`, which the no-demonstration arm answered correctly. The retrieved episode is from another scenario and its own answer opens with three DECELERATEs, which the demonstration arm reproduces on a clear road. Delivered as frames with its text removed, the same episode leaves the answer correct, placing this shortcut in the text channel. Night frames, gamma-lifted for print.



Figure 8: **Searching for the demonstration's subject.** VisualWebArena, `vwa.123`, GPT-5.6. The baseline scrolls the page it is already on and solves at step 4. Given an off-target classifieds episode, the same model leaves that page at step 2 to search for the demonstration's subject and runs out of step budget. No text is copied verbatim; the demonstration's subject displaces the task's own goal.

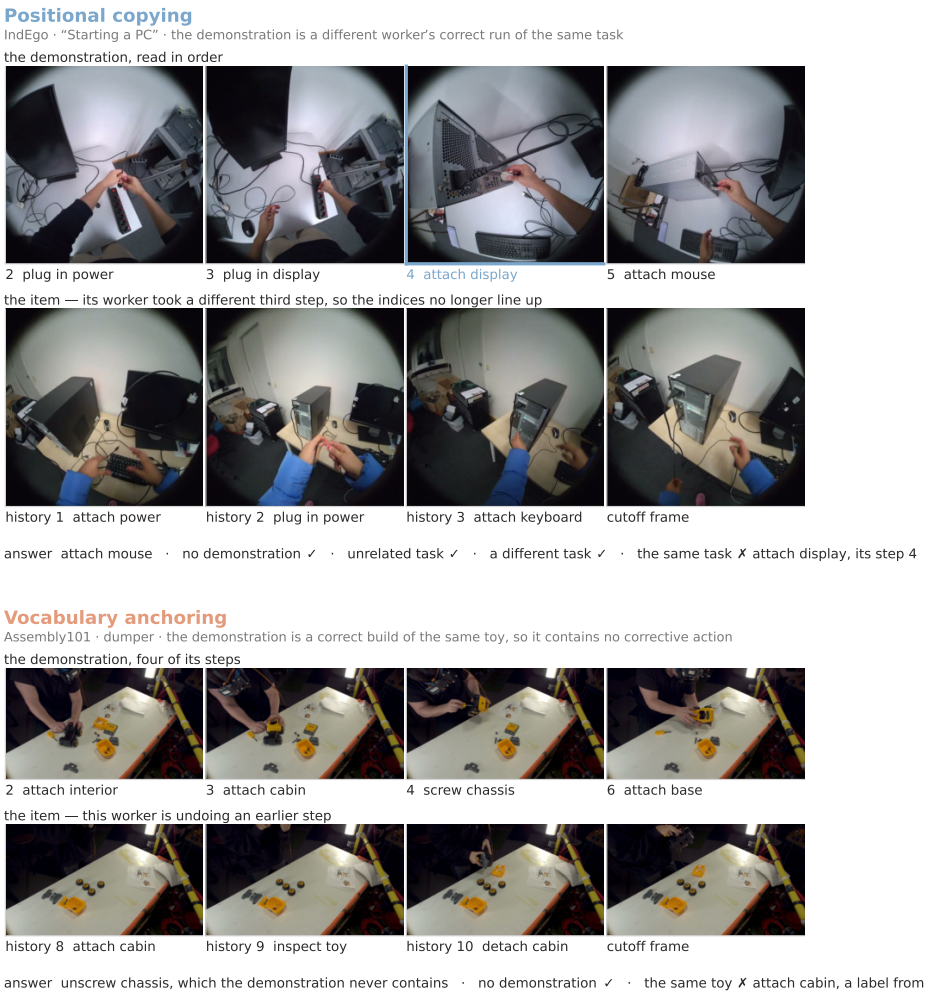


Figure 9: **Positional copying and vocabulary anchoring.** IndEgo and Assembly101, one item each, both answered correctly by the no-demonstration arm and turned wrong by a demonstration. **Positional copying:** the item's worker attached the keyboard at step three, where the demonstration's worker plugged in the display, so the ground truth and the demonstration diverge, and the model answers with the demonstration's fourth step. The correct answer, *attach mouse*, appears one step later in that same demonstration, so the label was available and the alignment is what failed. **Vocabulary anchoring:** the item's worker is undoing an earlier step, and *unscrew chassis* appears in no correct build; the model answers *attach cabin*, which the demonstration contains.

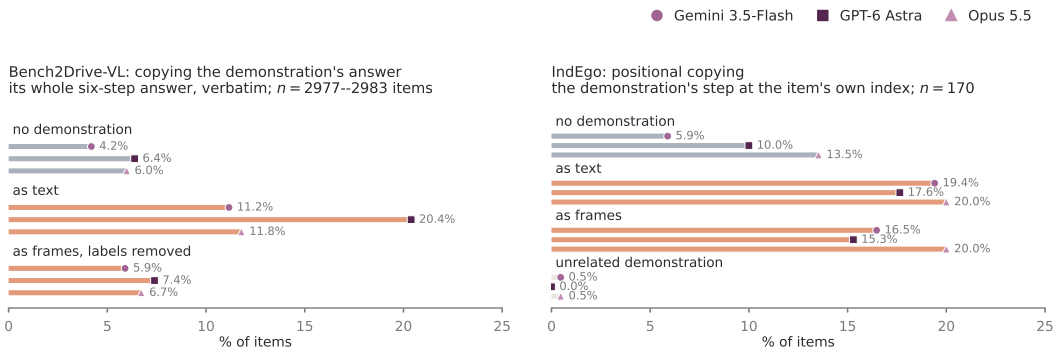


Figure 10: **How often each model copies from the demonstration.** We count only items where copying gives a wrong answer. On Bench2Drive-VL, the model outputs the demonstration's six-step answer word for word. On IndEgo, the model outputs the demonstration's step at the item's own position. The no-demonstration bar applies the same test to the no-demonstration answers, which can match by chance. For every model, each demonstration arm differs from that bar at McNemar $p < 0.05$ (largest $p = 0.043$). IndEgo's unrelated arm is the control (215 items).