USC Viterbi Information Sciences Institute

GUI-Primitives: Diagnosing Spatial Reasoning Failures in Vision-Language GUI Grounding

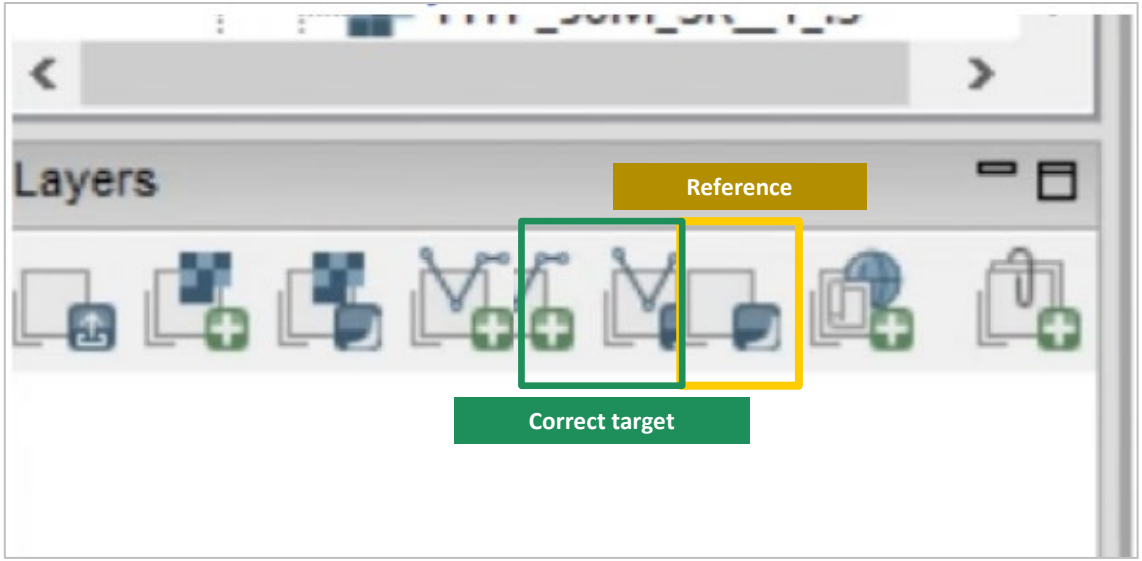
Md Abrar Jahin (USC, USC Information Sciences Institute)
Md Rizwan Parvez
(Qatar Computing Research Institute, QCRI)

EMNLP 2026 | Paper 9011

Can an AI find the right place to click when we describe where something is?

Where should the AI click?

A real screenshot from the benchmark and one short instruction



Finding where to click on a screen is called GUI grounding.

Instruction

Select the control to the left of 'Add various overlays'.

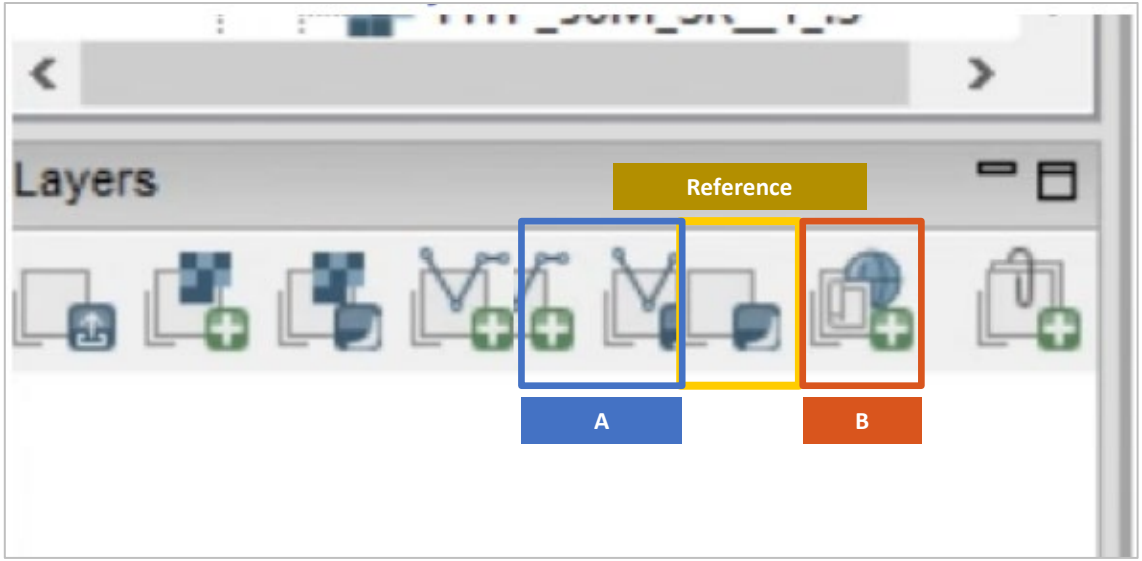
Gold box: the item named in the instruction.
Green box: the correct target.

The AI must answer with one click position on the screen.

Screenshot: UI-Vision item, Appendix figure (rel-pos-horizontal, real screenshot). Annotations added for the talk.

The same screen can need a different click

Two instructions, one word changed, two different correct answers



Note: in 253 of 497 pairs, other words in the sentence also vary. The relation word is what moves the answer.

A Select the control to the **left** of 'Add various overlays'.

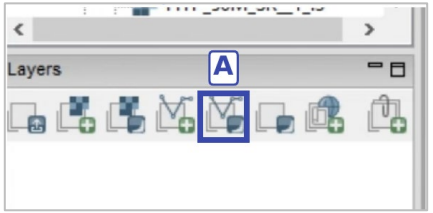
B Select the control to the **right** of 'Add various overlays'.

Always clicking the same popular icon passes one instruction and fails the other. We call these contrastive pairs.

Pair construction: Section 3.2 and Appendix, details of minimal-pair construction (253 of 497 pairs vary other wording).

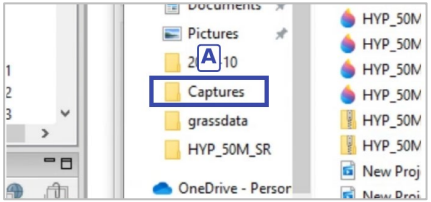
Seven basic position rules

Each tile is a real item from the benchmark. The box marks the correct target.



Left or right

left of / right of



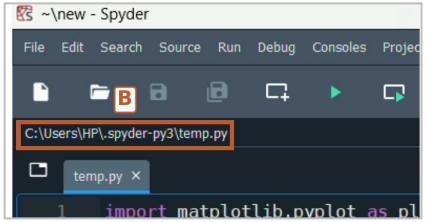
Above or below

above / below



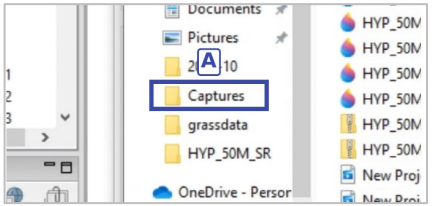
Inside or outside

inside / outside a dialog



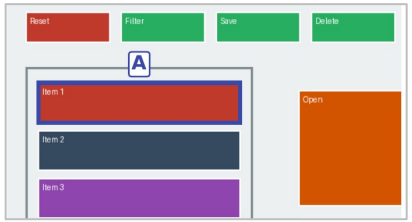
Same row or column

same row / same column



Nearest or farthest

nearest to / farthest from



Position in a list

first / second entry



Visible or partly covered

fully visible / partly hidden

Formal name in the paper: spatial primitives (Table 1).

Table 1; Appendix figures with three example pairs per primitive.

What we tested

Questions, models, screens, and the scoring rule

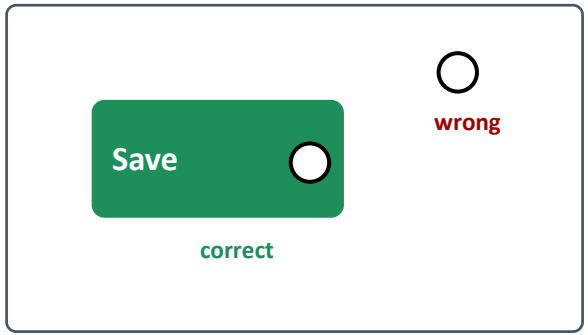
994
questions, in 497 pairs

19
vision-language models

290
use real screenshots

704
use generated screens

How a click is scored



A click counts only when it lands inside the correct item.

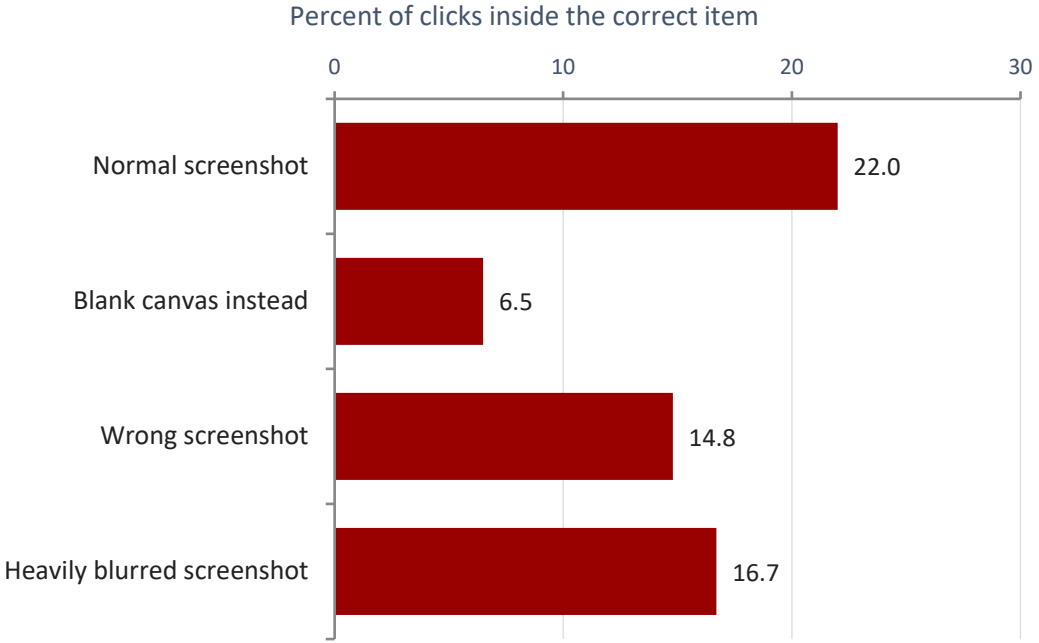
Formal name: point-in-box accuracy.

Five people checked a sample of 196 questions, not all 994 (details in backup).

Counts: Section 3.3 (290 real, 704 synthetic), Section 4 (19 models), Section 3.2 (497 pairs). Scoring: Section 4, Metrics.

Does the model actually use the screen?

A check that the test is honest: Qwen2.5-VL-7B on all 994 questions



Remove the screen, or show the wrong one, and accuracy collapses. The model really uses the picture.

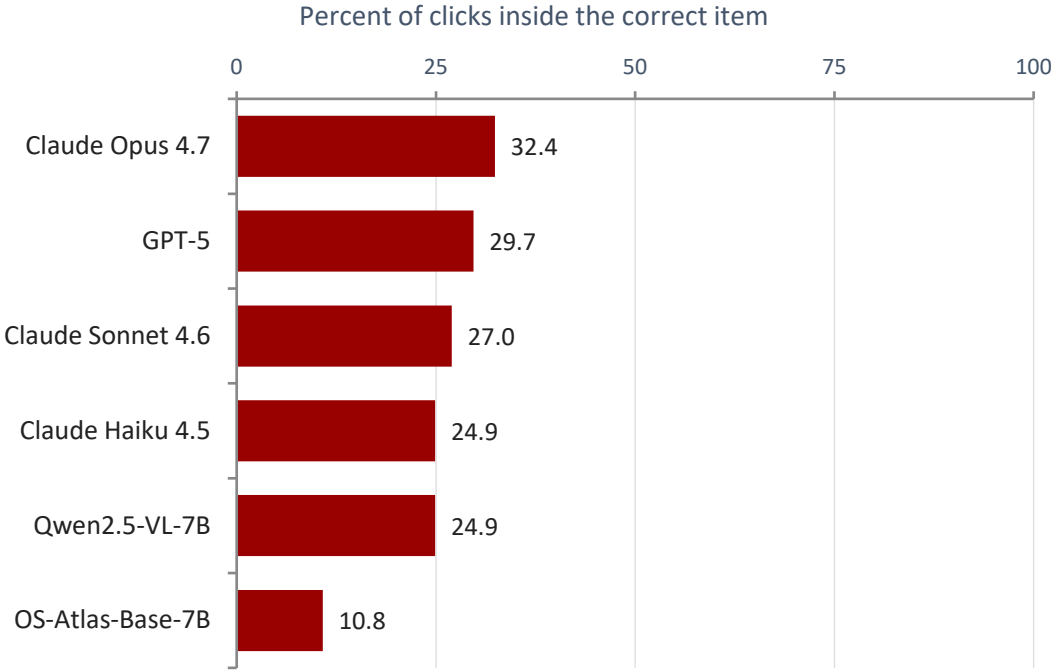
Heavy blur removes less of the signal (22 to 17 percent). The model seems to lean on overall layout more than on fine text.

All three drops are statistically significant; the blur drop is below our pre-registered ten point threshold.

Figure 4 and Section 5.1: blank canvas 22.0 to 6.5, shuffled 14.8, blur sigma 12 px 16.7; paired McNemar p < 0.001 for all three.

Even the best model misses many clicks

Strict click accuracy on the 185 human-checked questions



The best model lands inside the correct item on about one click in three.

The strongest commercial models cluster between 25 and 32 percent.

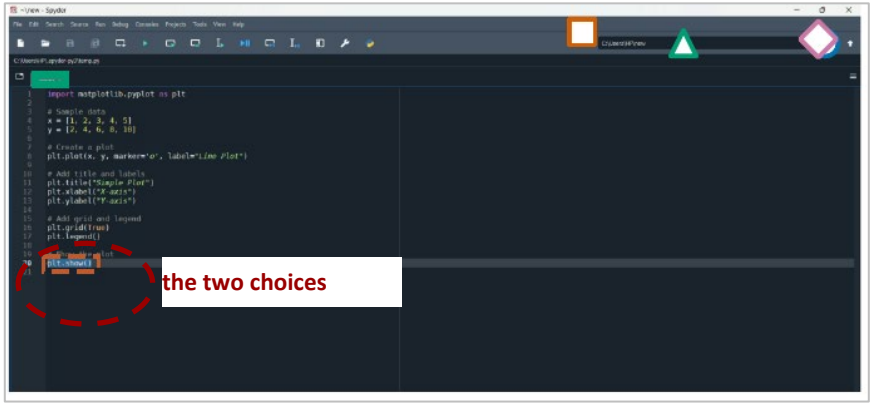
A 7B open model, Qwen2.5-VL, is close to Claude Haiku 4.5 here.

Human annotators are not on this chart: they chose between two given options (96.9 percent), a different task.

Table 2, Clean column (n = 185). Six representative models shown; all 19 in backup and Appendix.

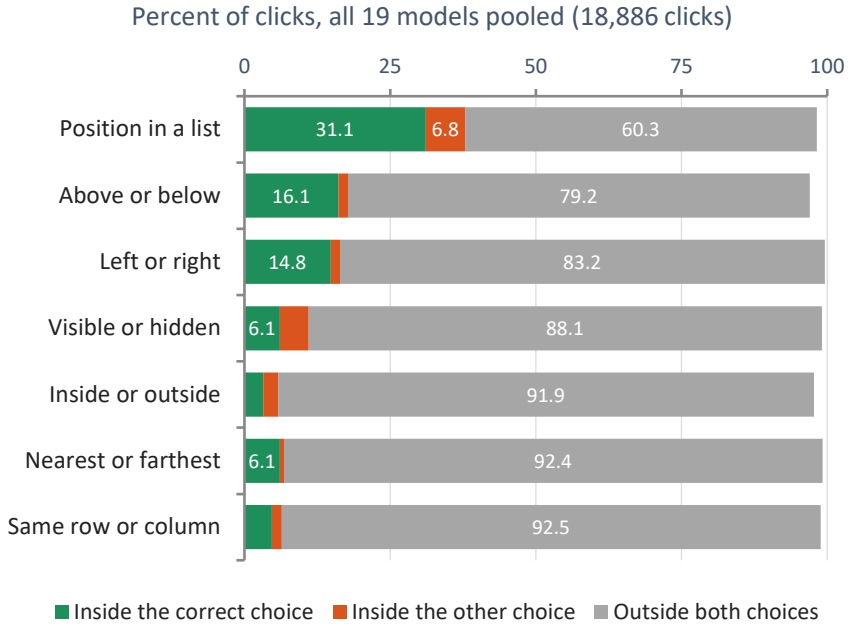
Most clicks miss both choices

Where did the click land: inside the correct choice, inside the other choice, or outside both?



Real example. Instruction: select the button nearest to 'current file directory'. All four models click in the top toolbar, far from both choices.

Across all 19 models, 60 to 92 percent of clicks land outside both choices.



Segments under 5 percent are unlabeled (exact values in the backup table). Clicks with no readable coordinate (0.4 to 3 percent) are not shown.

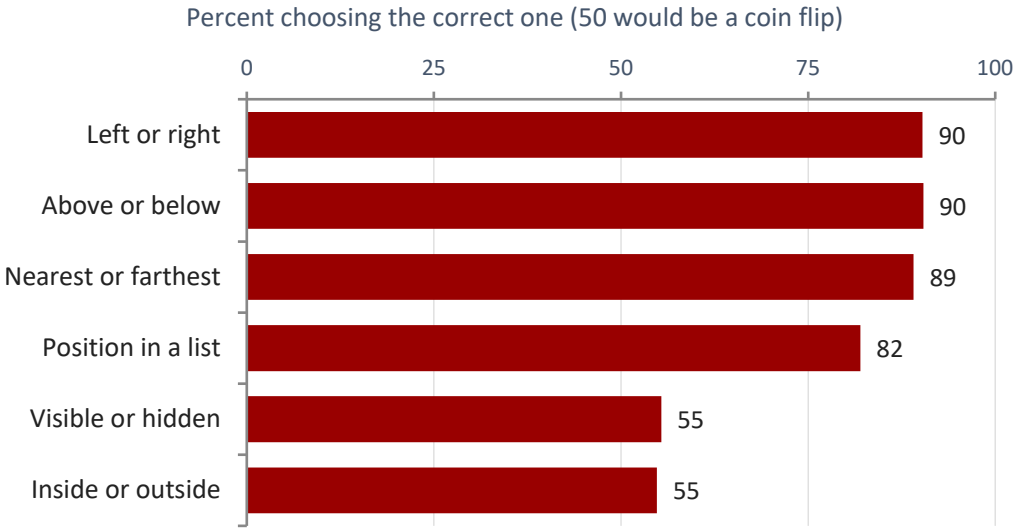
Appendix, candidate-level classification table (Tgt, Dis, Neither columns), strict rule, pooled over 19 models. Example: Appendix case study 2.

Finding versus choosing: two different steps

Among clicks that reached one of the two choices, how often was it the correct one?

Step 1 Find the two relevant choices → **Step 2** Choose the correct one

This slide looks only at Step 2.



When a click reaches one of the two choices, it is usually the correct one for four rules (82 to 90 percent).

Inside or outside and visible or hidden: **about 55 percent, not different from a coin flip.** These two rules use generated screens.

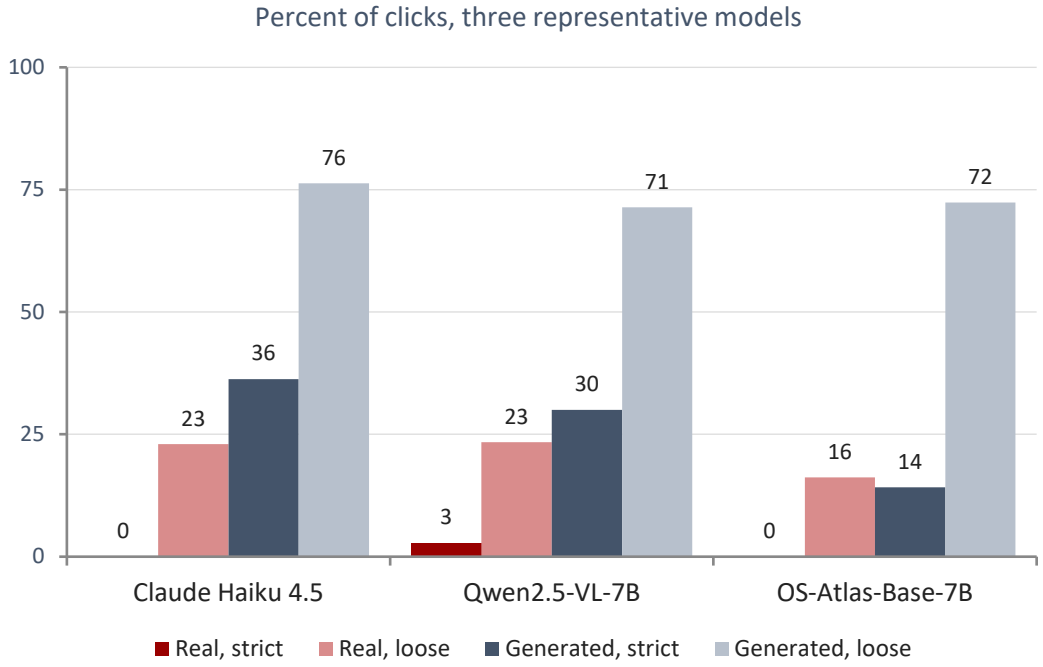
Same row or column is not shown: its value changes with the counting rule.

Only clicks that reached one of the two choices are counted here (157 to 1,022 clicks per rule).

Appendix candidate-level table, Sel. column (strict rule) and binomial test against 0.50; alignment 0.728 strict vs 0.529 permissive, not interpreted.

Real screenshots are harder

Real versus generated screens. Strict: inside the item. Loose: within two box diagonals of its centre.



On generated screens, many misses are small: the loose score reaches 71 to 76 percent.

On real screenshots, even the loose score stays low (16 to 23 percent). The model picks a different item, not a slightly wrong spot.

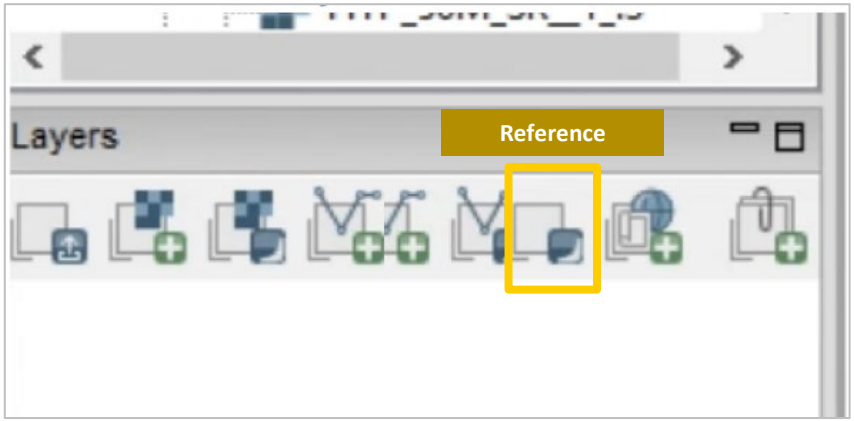
On real screenshots, 96 percent of clicks fall outside both choices (all 19 models pooled). When a click does reach a choice, it is the correct one 75 percent of the time.

Table 3 (Appendix, real vs synthetic); Section 5.4; Appendix candidate-level analysis, real arm: 96.2 percent neither, selection 0.747 pooled.

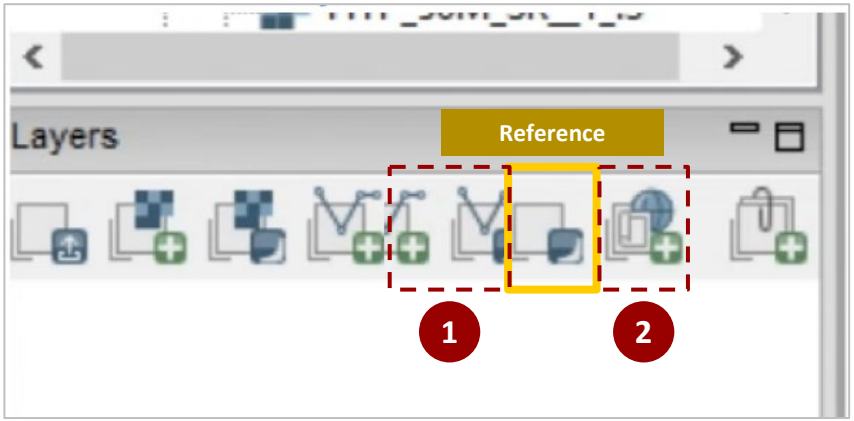
What happens when we mark the choices?

We give the model the two choices. It then selects a mark.

Original



Choices marked (illustration of the method)



The model receives the two relevant choices.

+57	+42	+35	+9
GPT-5 30 to 87 percent	OS-Atlas-Base-7B 10 to 52 percent	Qwen2.5-VL-7B 22 to 57 percent	Claude Opus 4.7 31 to 40 percent, not significant

Section 7.1 and Figure 3 (intervention deltas, Holm-adjusted). Points are percentage points. Diagnostic upper bound, not a deployable method (Section 8).

Extra reasoning did not clearly improve this test

Two other ideas, tested on Qwen2.5-VL-7B

Ask the model to reason step by step before clicking

+0.5

points, not significant

Adjust the model's internal activations toward correct answers

+1.5

points, not significant

Mark the two choices on the screen (previous slide)

+35

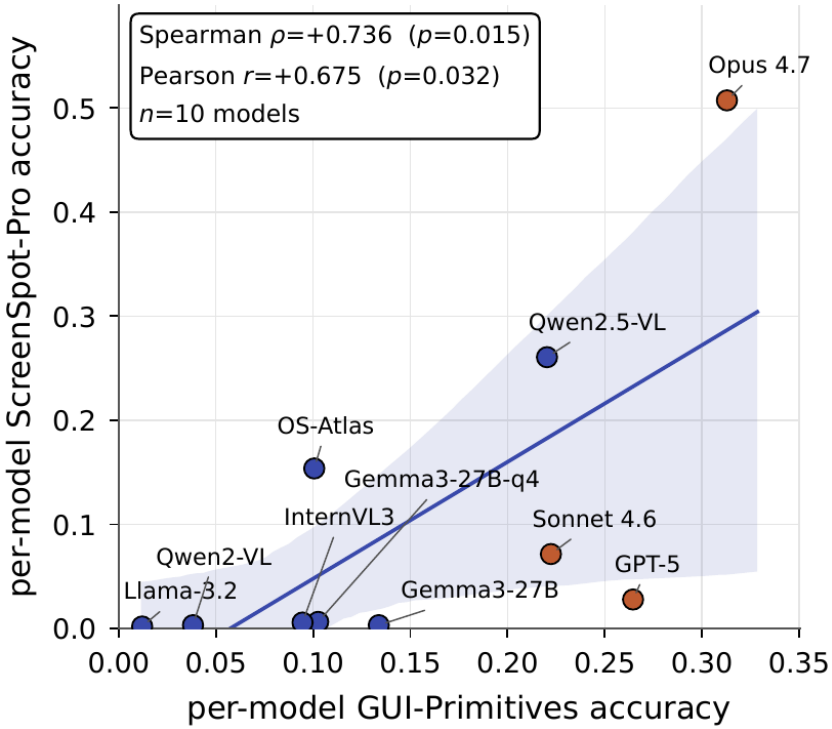
points, significant

One model and two methods. We do not generalize this to every model or every reasoning method.

Section 7.2: CoT +0.5 (p = 0.51), activation steering +1.5 (p = 0.066), both on Qwen2.5-VL-7B. Marking +35.1 on the same model, Section 7.1.

Does this relate to another screen test?

Our benchmark against ScreenSpot-Pro, one point per model



Across ten models, those that did better on our test also tended to do better on ScreenSpot-Pro.

Rank correlation (Spearman) about 0.74, p = 0.015.

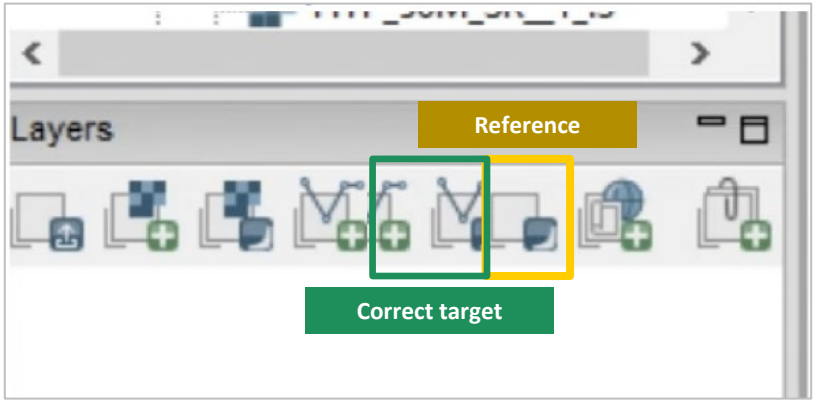
The pattern holds on both the real and the generated screens.

This is an association across ten models. It does not prove that improving this benchmark will improve complete computer-use tasks.

Figure 6 and Section 6: Spearman 0.736 (p = 0.015, n = 10); real subset 0.705, synthetic subset 0.760; leave-one-out range 0.62 to 0.86.

What should we take away?

Back to our first screen



Checking the final click tells us whether the model failed.

Looking at where it clicked helps us understand the failure.

For most rules, the hard step is finding the two relevant items. For inside or outside and visible or hidden, the relation itself is also not followed.

Code github.com/Abrar2652/gui-primitives
Data huggingface.co/datasets/kagnlp/gui-primitives
Site abrar2652.github.io/gui-primitives

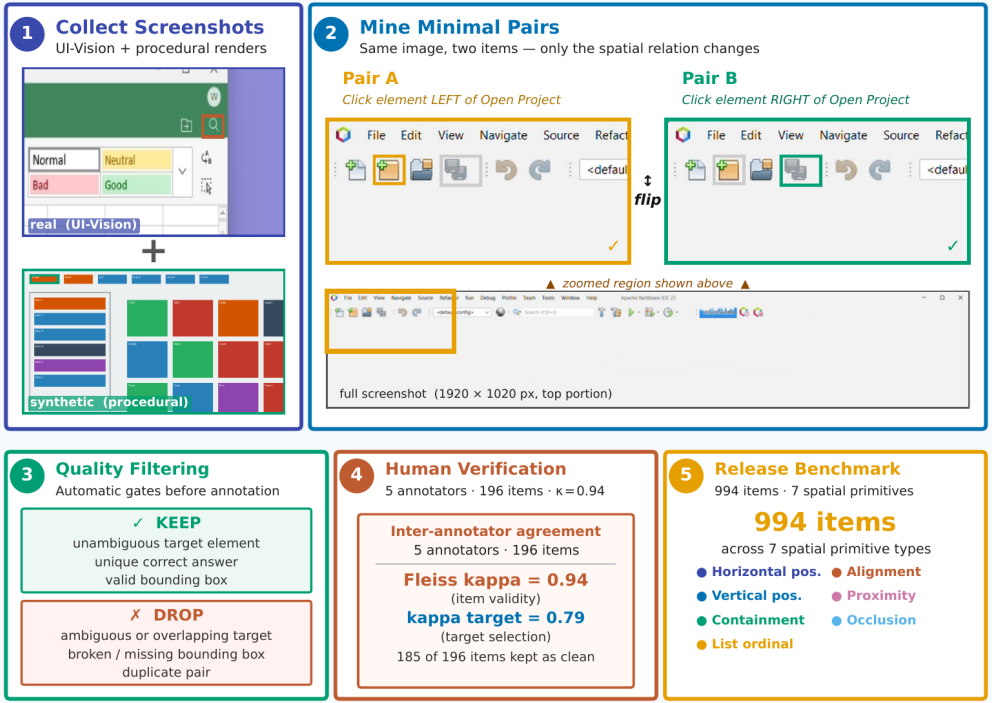
Thank you. Questions?

Md Abrar Jahin, jahin@usc.edu

Links as printed in the paper front matter. Summary statements: Sections 5.3, 8, and Appendix candidate-level analysis.

Backup: benchmark construction and human check

Construction pipeline and human verification



Two sources

290 real screenshots (UI-Vision) and 704 generated screens. Generated screens are needed for inside or outside and visible or hidden.

Pairs

497 pairs share a screenshot and a reference item; the relation changes and the target moves to the other choice. In 253 pairs, other wording also varies.

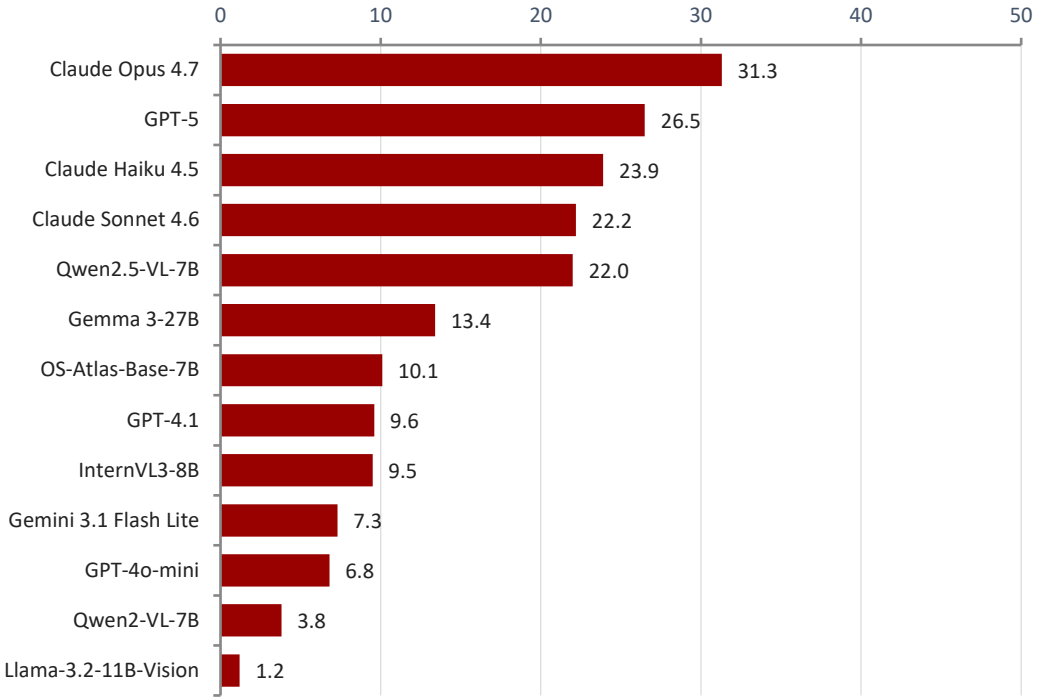
Human check

Five annotators reviewed 196 questions. Agreement: kappa 0.94 (well formed), 0.79 (correct target). 11 questions removed, 185 kept. On this two choice task, annotators picked the correct target 96.9 percent of the time.

Figure 1 (construction banner); Sections 3.2 to 3.4; Appendix on sources, construction, and verification.

Backup: results for all reported models

Strict click accuracy on all 994 questions (Table 2)



Strict accuracy

The click must land inside the correct item's box (point-in-box).

Loose accuracy

The click is within two box diagonals of the item's centre. Used to separate small misses from wrong items.

Confidence intervals

Bootstrap 95 percent intervals, 2,000 resamples. Paired comparisons use McNemar with Holm correction.

All 19 models, including Gemma 3 4B and 12B, PaliGemma 2, and MiniCPM-V, are in the paper appendix.

Table 2, Full column (n = 994); Section 4, Metrics.

Backup: real versus generated screens

Strict (inside the box) and loose (within two diagonals) accuracy, by screen source

Model	Real, strict	Real, loose	Generated, strict	Generated, loose
Claude Haiku 4.5	0.000	0.230	0.363	0.763
Qwen2.5-VL-7B	0.028	0.234	0.300	0.714
OS-Atlas-Base-7B	0.000	0.162	0.142	0.724
Gemini 3.1 Flash Lite	0.000	0.115	0.111	0.741
GPT-4o-mini	0.000	0.033	0.111	0.733
Qwen2-VL-7B	0.000	0.066	0.054	0.693
Llama-3.2-11B-Vision	0.000	0.062	0.017	0.616

Generated screens: loose accuracy is 0.62 to 0.76, so many misses are small position errors near the right item.

Real screenshots: strict accuracy is near zero and loose accuracy stays low (6 to 23 percent). The model picks a different item, not a slightly wrong spot.

On real screenshots, 96 percent of clicks fall outside both choices (all 19 models pooled).

Table 3 (Appendix, real vs synthetic strict and loose); Section 5.4; Appendix candidate-level analysis, real-screenshot arm (96.2 percent neither).

Backup: where every baseline click landed

All 18,886 clicks from 19 models, classified by the box they fall in (strict rule)

Rule	Correct %	Other %	Outside both %	No coord. %	Correct given reached	n	p vs 0.50
Inside or outside	3.2	2.6	91.9	2.3	**0.548**	157	0.26
Visible or hidden	6.1	4.9	88.1	0.8	**0.554**	298	0.07
Same row or column	4.7	1.7	92.5	1.1	**0.728**	173	<1e-8
Nearest or farthest	6.1	0.7	92.4	0.7	**0.891**	184	<1e-28
Left or right	14.8	1.6	83.2	0.4	**0.903**	443	<1e-72
Above or below	16.1	1.7	79.2	3.0	**0.904**	481	<1e-79
Position in a list	31.1	6.8	60.3	1.8	**0.820**	1022	<1e-99

Permissive rule (click within one box diagonal of the centre) agrees for six rules: inside or outside 0.539, visible or hidden 0.557, the four target-favoring rules 0.69 to 0.77. Same row or column moves from 0.728 to 0.529, so we do not interpret it. The only model with a significant excess of wrong-choice clicks is Gemma3-4B (0.328, n = 67).

Appendix, candidate-level classification table and the sensitivity paragraph on the inclusion rule.

Backup: intervention details and limitations

How each intervention was run, and what the study does not cover

Marking the choices (Set-of-Mark)

Numbered marks on exactly the two choices, IDs shuffled per item. The model answers with a mark number, mapped back to the box centre. Baseline and marked runs are not identical tasks.

Step by step reasoning (CoT)

A one paragraph prompt names the relation and asks for reasoning before the click. Qwen2.5-VL-7B: +0.5 points, p = 0.51.

Activation steering

Mean difference vectors from 200 held out items, added at decoder layers 12 to 19, scale 4.0 chosen from 1, 2, 4, 6. Qwen2.5-VL-7B: +1.5 points, p = 0.066. A verbosity control confirmed the pipeline works.

Limitations

Inside or outside and visible or hidden come only from generated screens, so their results have limited reach.

Per rule regression coefficients are small; we claim only the model level association.

We stop at ScreenSpot-Pro; full task success on OSWorld is future work.

English click targets only; drag targets and other layouts are planned.

Marking supplies the two choices; real systems must find them, and multi candidate scenes are untested.

Strict scoring is binary; the loose metric only partly separates small misses from wrong items.

Section 7, Appendix intervention implementation details; Section 9 Limitations (i) to (vi).