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

Md Abrar Jahin[♡♠†] **Md Rizwan Parvez**[♣†]
[♡]University of Southern California [♠]USC Information Sciences Institute
[♣]Qatar Computing Research Institute (QCRI)
jahin@usc.edu mparvez@hbku.edu.qa

Abstract

Computer-use agents ground natural-language instructions in screenshots to locate interface elements, yet existing benchmarks do not isolate whether models bind relational language to the correct element. We introduce GUI-PRIMITIVES, a 994-item benchmark of contrastive instruction pairs over seven spatial relations in graphical user interfaces (left/right, above/below, containment, alignment, proximity, list ordinal, occlusion). Each pair holds the screenshot and anchor fixed while changing the relation expression, so the correct target moves between two designated candidates. Five annotators validate a 196-item subset ($\kappa = 0.94$ well-formedness; $\kappa = 0.79$ target selection). Nineteen vision-language models reach at most 32% strict point-in-box accuracy. Because models emit unconstrained coordinates, we classify each prediction by the candidate region it falls within. Predictions fall outside both candidates on 60–92% of items. Conditional on falling within a candidate region, target selection reaches 0.82–0.90 for horizontal position, vertical position, proximity, and list ordinal, but does not differ significantly from 0.50 for containment and occlusion: most failures reflect candidate localization rather than relation understanding. Across ten models, benchmark accuracy correlates with ScreenSpot-Pro accuracy (Spearman $\rho = +0.74$), an exploratory association at this sample size. Marking the two designated candidates raises selection accuracy by 35–57 percentage points, an oracle diagnostic that supplies the candidate set rather than a deployable method. We release the benchmark, predictions, and code.

[†]Corresponding Author(s)

Source Code: https://github.com/Abrar2652/gui-primitives/
Dataset: https://huggingface.co/datasets/kagnlp/gui-primitives/
Website: https://abrar2652.github.io/gui-primitives/

1 Introduction

A new class of *computer-use agents* now reads a screenshot, decides what to do, and emits a click coordinate or a keystroke (Xie et al., 2024; Cheng et al., 2024; Qin et al., 2025; Gou et al., 2025; Wu et al., 2025b; Wang et al., 2025a). These agents are beginning to be deployed; their first job, before any agent loop, is the same job a person does when they open a window: read the screen, find the right control, and put the pointer in the right place. We call this *GUI grounding*, and a growing body of literature shows that frontier vision-language models still trail humans in this task (Li et al., 2025; Yang et al., 2025a).

A natural question follows. If we strip the agent away and ask a present-day VLM only to handle the elementary spatial reasoning that GUI grounding requires: to the right of `Save`, inside the `Layers` panel, the third item in the menu, does it succeed? Existing benchmarks do not isolate this skill. End-to-end benchmarks like OSWorld (Xie et al., 2024) combine grounding with planning and action; coarse grounding benchmarks like ScreenSpot (Cheng et al., 2024) and ScreenSpot-Pro (Li et al., 2025) measure whether the click lands within the target box but do not separate failures by spatial primitive. As a result, when an agent clicks the wrong button, we do not yet know *which* elementary skill broke.

We close this gap with a controlled diagnostic. GUI-PRIMITIVES is a benchmark of 994 contrastive-pair questions over seven spatial relations in GUI screenshots. Each question is paired with a twin that keeps the screenshot and anchor fixed and changes the relation expression (`left` ↔ `right`, `inside` ↔ `outside`, and so on), so that the correct target moves to the other designated candidate. In 253 of 497 pairs, the surrounding instruction template also varies lexically (verb, head noun, anchor quoting); this variation is not confounded

Proceedings of the 2026 Conference on Empirical Methods in Natural Language Processing, pages 59530–59564
October 24-29, 2026 ©2026 Association for Computational Linguistics

with the relation term (Appendix J.2). This construction, adapted from the minimal-pair tradition in linguistic evaluation (Linzen et al., 2016; Marvin and Linzen, 2018; Kamath et al., 2023), controls for screenshot-specific salience and fixed answer preferences: a model that always clicks the most salient or central element scores correctly on one twin and incorrectly on the other. We evaluate 19 vision-language models spanning open-weight and proprietary systems. Five annotators independently validated a 196-item subset (Fleiss $\kappa = 0.94$ for well-formedness) and selected the instruction-consistent candidate on 96.9% of retained items; this is annotator target-selection accuracy on a two-candidate task, not a model-matched grounding score.

The evaluation yields three main findings. Claude Opus 4.7, the strongest model, reaches 31% strict point-in-box accuracy on the human-clean subset. On four of seven primitives (containment, occlusion, alignment, proximity), every model we test falls below the two-candidate reference level of 0.50. Candidate-level analysis (§5.3) indicates that this below-reference accuracy is primarily attributable to predictions outside both candidate regions rather than to the selection of the contrastive element. On real desktop screenshots from UI-Vision (Nayak et al., 2025), accuracy is lower still, and candidate-level analysis attributes this primarily to predictions outside both candidate regions (96.2% of real-subset predictions) rather than to relation errors. Pair consistency is below the value expected under independent responses because carrier-template variation and off-candidate prediction also contribute to this dependence; therefore, candidate-level analysis (§5.3) is required to interpret it.

Yet the diagnosis is not only a problem statement. Three results follow. First, the benchmark tracks downstream performance. Across ten models for which we also run ScreenSpot-Pro, GUI-PRIMITIVES accuracy correlates with grounding accuracy at Spearman $\rho = +0.74$ ($p = 0.015$). The correlation is stable across the real-screenshot and synthetic slices of GUI-PRIMITIVES. Item-level logistic regression on $n = 15{,}810$ model–item pairs corroborates this association (pseudo-$R^2 = 0.40$); we report it conservatively. Second, we test three training-free interventions: Set-of-Mark (SoM) prompting (Yang et al., 2023), primitive-aware chain-of-thought (CoT), and activation steering (Kang et al., 2025; Sivakumar et al., 2025). Marking the two designated candidates (an oracle two-candidate diagnostic) raises selection accuracy by 35–57 percentage points, lifting GPT-5 from 30% to 87% and OS-Atlas from 10% to 52%; CoT and activation steering yield no significant gain. Because the intervention supplies the candidate set that the baseline task requires the model to locate, it is an upper bound on candidate discrimination rather than a deployable method (§8). Third, three shortcut controls (blank canvas, shuffled instructions, heavy blur) reduce accuracy as expected for a grounded benchmark; the blur control produces a significant but sub-threshold drop, and the negative results for CoT and activation steering are reported alongside the positive result for SoM.

We make four contributions. First, we introduce GUI-PRIMITIVES, a 994-item contrastive-pair diagnostic benchmark covering seven spatial relations in GUI screenshots, together with a five-annotator verified 196-item core ($\kappa = 0.94$). Second, we evaluate 19 models across five families and localize the dominant failure to candidate localization, predictions fall outside both candidate regions on 60–92% of items, with relation-word errors confined to containment and occlusion, alongside a 65-point gap from human performance. Third, we show that primitive competence correlates with downstream GUI grounding on ScreenSpot-Pro (Spearman $\rho = +0.74$, $p = 0.015$). Fourth, we find that SoM transfers across both open and closed models, improving GPT-5 by 57 points, whereas CoT prompting and activation steering do not; we release GUI-PRIMITIVES, all model predictions, and analysis code.

2 Related Work

GUI grounding and computer-use agents. A line of recent work targets agents that operate real software. OSWorld (Xie et al., 2024) provides a real Linux and Windows VM environment with execution-based scoring on 369 tasks; a subsequent human-verified release closes ambiguous tasks (XLANG Lab, 2025). ScreenSpot (Cheng et al., 2024) and ScreenSpot-Pro (Li et al., 2025) measure pixel-grounding in isolation, the latter targeting professional software at high resolution; MMBench-GUI (Wang et al., 2025b) adds a hierarchical multi-platform evaluation. Grounding models trained on screenshots include SeeClick, OS-Atlas (Wu et al., 2024), UI-TARS (Qin et al., 2025), UGround (Gou et al., 2025), Aria-UI (Yang

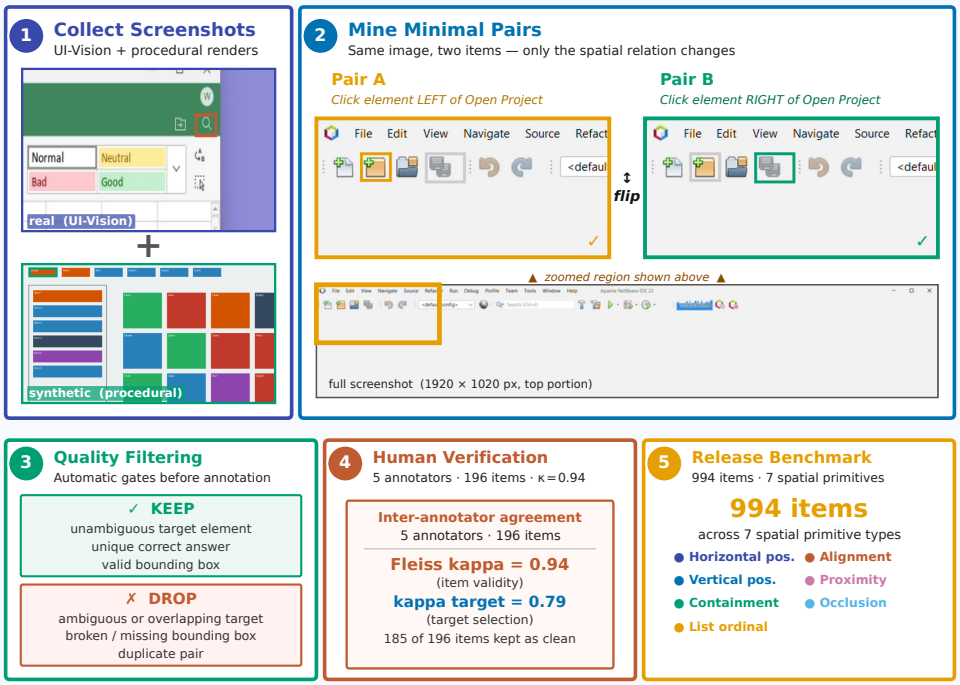

Figure 1: **GUI-PRIMITIVES dataset construction pipeline.** Two source corpora: real desktop screenshots from UI-Vision and controlled synthetic stimuli from our procedural renderer, feed into per-primitive minimal-pair miners, pass through automatic quality gates, are cross-checked by a five-annotator human verification pass (Fleiss κ=0.94), and released as the 994-item benchmark. Stage colors follow the primitive palette used throughout the paper.

et al., 2025b), GUI-Actor (Wu et al., 2025b), Jedi (Xie et al., 2025), OpenCUA (Wang et al., 2025a), and the test-time scaling agent GTA1 (Yang et al., 2025a). Closest in spirit, GUI-Cursor (Zhao et al., 2025) probes a cursor-based agent's spatial reasoning under reinforcement learning, and Beyond Semantics (Qi et al., 2025) argues that large vision-token norms suppress positional encoding, a mechanistic candidate for our failures. These works ask whether models land the click or how to train them; we ask which elementary spatial skill they lack when they miss.

Spatial reasoning in vision-language models. Several benchmarks reveal that VLMs struggle with spatial relations even on natural images. WHAT'S-UP (Kamath et al., 2023) uses controlled minimal pairs to show that contrastive VLMs handle left/right and on/under poorly despite high captioning accuracy. VSR (Liu et al., 2023) catalogs 66 spatial relations in natural images; BLINK (Fu et al., 2024) bundles 14 perceptual primitives and finds even GPT-4V well below humans; CV-Bench (Tong et al., 2024) shows that 2D/3D spatial primitives are bottlenecked by the visual encoder. SpatialVLM (Chen et al., 2024) addresses these gaps by co-training on synthetic 3D-grounded data. RocketScience (Hoehing et al., 2025) uses contrastive pairs to disentangle object localization from spatial reasoning, and finds that only top reasoning models succeed. GUI-PRIMITIVES extends this controlled methodology to GUI screenshots, where the visual prior is starkly different: text labels everywhere, near-identical icons, dense layout.

Minimal-pair evaluation. Minimal pairs originate in psycholinguistic evaluation (Linzen et al., 2016; Marvin and Linzen, 2018), where a single lexical or syntactic edit creates a contrast whose disambiguation requires understanding rather than surface statistics. WHAT'S-UP brought the design to vision-language. We adopt the same logic for GUI screenshots and add pair-consistency (Kamath et al., 2023) as an internal diagnostic of relational failure (§5.5).

Inference-time interventions. SoM prompting (Yang et al., 2023) overlays numbered marks on visual regions so the model names the mark rather than regressing a coordinate. SoM-LLaVA (Yan et al., 2024) shows that open-source MLLMs need explicit SoM training data to match closed models. Visual prompting is now a standard tool in GUI-agent stacks, with recent training-free zoom and modality-aware variants: DiMo-GUI (Wu et al.,

2025a) decouples text and icon modalities and applies test-time scaling; RegionFocus (Luo et al., 2025) reaches 61.6% on ScreenSpot-Pro with a Qwen2.5-VL-72B backbone by iteratively zooming into a focal region. Activation and attention steering, including the localization-head method of Kang et al. (2025) and the SteerVLM framework (Sivakumar et al., 2025), modify model internals at inference time without weight updates. We test one representative from each family.

3 The GUI-PRIMITIVES Benchmark

GUI-PRIMITIVES is designed for one purpose: to ask whether a vision-language model has each of the seven elementary spatial skills that GUI grounding requires. Three design choices follow from this purpose: the seven primitives (§3.1), the minimal-pair construction (§3.2), and the human verification protocol (§3.4). The full construction pipeline is summarized in Figure 1.

3.1 Seven elementary primitives

We chose primitives that are unambiguous, common in everyday GUI language, and decomposable. Table 1 lists all seven together with their relation words and chance level. The primitives split into three groups. *Cardinal direction* covers horizontal and vertical relative position. *Topology* covers containment (inside or outside a panel). *Layout* covers alignment (sharing a row or a column), proximity (which of several elements is nearest), list ordinal (the third item of a menu), and occlusion (visible or partly covered by a pop-up).

Why these seven? We require each primitive to be lexically realizable in a short user instruction, separable by a single relation-expression change into a contrastive pair, and a basic building block of compositional GUI relations. The seven jointly cover the four broad classes cataloged by prior natural-image work: cardinal direction, topology, ordinality, and layout-distance (Liu et al., 2023; Kamath et al., 2023; Hoehing et al., 2025). We do not claim coverage of every spatial relation, only of the elementary ones from which the rest compose.

3.2 Minimal-pair construction

Every item in GUI-PRIMITIVES appears as one half of a contrastive pair: the two members share one screenshot and one anchor element, and the relation expression changes, with the correct target as the *contrastive* element (Figure 2, details in Appendix J.1). This construction removes the shortcuts that inflate scores on coarser benchmarks: a model that always selects the most central, salient, or frequently labeled element earns one point and loses one on a pair, so the pair-level score is zero. We measure this directly as *pair-consistency* (§5.5). Three representative pairs for each of the remaining six primitives are shown in Appendix P.

Table 1: The seven elementary spatial primitives in GUI-PRIMITIVES, with their canonical relation words and two-candidate reference levels. All seven primitives, including *list-ordinal*, are realized as binary minimal-pair contrasts (71 items per side), so the reference level is 0.50 throughout; §4 explains why this is a forced-choice reference rather than the chance level of unconstrained clicking.

Primitive	Relation words	Group	Ref. level
rel-pos-horizontal	left / right	cardinal	0.50
rel-pos-vertical	above / below	cardinal	0.50
containment	inside / outside	topology	0.50
alignment	same row / col.	layout	0.50
proximity	nearest / farthest	layout	0.50
list-ordinal	first / second	layout	0.50
occlusion	visible / hidden	layout	0.50

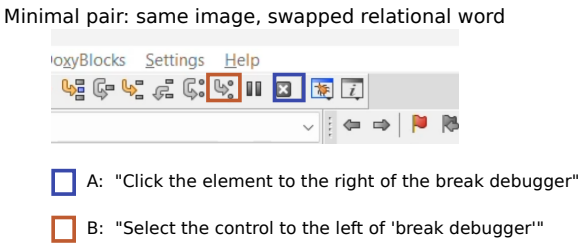

Figure 2: A contrastive pair from GUI-PRIMITIVES (details in Appendix P). The screenshot is fixed; one word flips, and the correct target shifts to the other candidate. A model exploiting an answer prior passes one twin and fails the other.

3.3 Sources: real and synthetic screenshots

GUI-PRIMITIVES draws from two sources (Appendix J.2 and K), both reported separately throughout. The *real* arm ($n = 290$) uses element-grounding annotations from UI-Vision (Nayak et al., 2025) on Web, MacOS, VS Code, Office, and professional applications, covering five primitives whose annotations are recoverable from element labels: horizontal and vertical relative position, alignment, proximity, and a 24-item list-ordinal slice mined from real menus and lists. The *synthetic* arm ($n = 704$) is procedurally rendered with

known positions, controlled distractors, and a fixed layout language; it is the only way to obtain *containment* and *occlusion* cleanly, and it supplies the remaining *list-ordinal* items, since real screenshot annotations do not ship the parent–child or overlay–target metadata. We treat the synthetic arm as a controlled-stimulus arm in the WHAT'S-UP tradition and disclose every result split by source.

3.4 Human verification

Five annotators independently judged a 196-item stratified subset (*human-verified core*), answering two questions per item: well-formedness and correct target (Appendix J.3). Agreement is high: Fleiss $\kappa_{Q1} = 0.942$ ("almost perfect" on the Landis–Koch scale, Landis and Koch, 1977) and $\kappa_{Q2} = 0.787$ ("substantial"). Eleven items judged invalid by the majority were dropped, leaving the 185-item *human-clean core*; human accuracy on it is 96.9%. Annotator instructions are in Appendix F.

4 Models and Protocol

Models. We evaluate 19 vision-language models: seven proprietary models accessed through commercial APIs (Claude Opus 4.7, Sonnet 4.6, Haiku 4.5, GPT-5, GPT-4.1, GPT-4o-mini, Gemini 3.1 Flash Lite); three Qwen-family open (Qwen2.5-VL-7B, Qwen2-VL-7B, OS-Atlas-Base-7B); InternVL3-8B; Meta Llama-3.2-11B-Vision; three sizes of Gemma 3; two PaliGemma 2 variants; and MiniCPM-V. Identifiers and endpoints are in Appendix B. Pixtral-12B and Idefics3 8B are excluded because their chat template formats are incompatible with our generic wrapper.

Prompting and decoding. All models use greedy decoding ($T = 0$) and a fixed seed. Per-family coordinate conventions (pixel, normalized $[0, 1]$, normalized $[0, 1000]$) are handled by a robust parser whose unit tests are bundled with the artifact. The three closed providers silently downsample large screenshots server-side (Anthropic $\leq$ 1568 px, OpenAI $\approx$ 1024 on the long side, Gemini variable); we pre-resize to a known long side and rescale predicted coordinates back to the original pixel space. The rescale was needed on a large fraction of items (e.g., 1,297 of 1,581 Sonnet ScreenSpot-Pro records); Appendix E documents the procedure and the per-model record counts.

Metrics. The primary metric, following ScreenSpot-Pro, is *point-in-box* accuracy: a prediction $\hat{p} = (\hat{x}, \hat{y})$ is correct iff $\hat{x} \in [x_1, x_2]$ and $\hat{y} \in [y_1, y_2]$ in the original pixel frame. *Loose* accuracy (prediction within $2\times$ target diagonal of the box center) separates sub-pixel imprecision from element misidentification on small icons. All accuracies carry bootstrap 95% CIs (2,000 resamples); paired comparisons use McNemar (McNemar, 1947) with Holm–Bonferroni (Holm, 1979) correction across the seven primitives. All seven primitives are binary minimal-pair contrasts (Table 1), so 0.50 serves as a *two-candidate reference level* throughout. Because models emit an unconstrained coordinate rather than selecting among candidates, 0.50 is a forced-choice reference rather than the chance level of the prediction task, and a prediction can fall outside both candidate regions. We flag a cell as *below the reference level* only when the upper bootstrap bound falls below 0.50. To distinguish selection of the contrastive element from predictions outside both regions, we classify every baseline prediction as target, distractor, neither, or invalid in Appendix R (further details in Appendix L).

5 Diagnostic Results

We report results in five parts, in this order: (i) controls that validate the benchmark, (ii) overall accuracy and the human gap, (iii) per-primitive structure of failure, (iv) the real-vs-synthetic decomposition, and (v) pair-consistency as direct evidence of relational failure.

5.1 Shortcut controls validate the benchmark

A diagnostic is only as good as the shortcuts it forbids (Appendix N.1). Figure 3 reports three controls on Qwen2.5-VL-7B over the full 994-item benchmark.

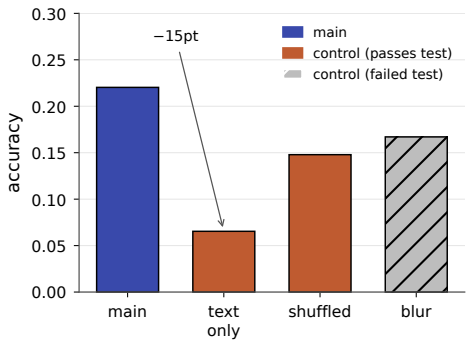

Figure 3: Shortcut controls (Qwen2.5-VL-7B, $n = 994$). Text-only and shuffled drops are both highly significant; blur produces a smaller but significant drop, below the pre-registered ten-point threshold.

Pairing the instruction with a *blank canvas* drops accuracy from 22.0% to 6.5% ($\Delta = -15.5$ pt, $p = 2.8 \times 10^{-27}$); *shuffling* the screenshot to one from a different item of the same primitive drops it to 14.8% ($\Delta = -7.2$ pt, $p = 1.3 \times 10^{-11}$). Both behave as a grounded model should: when the visual or the matching instruction is removed, the task collapses. *Heavy Gaussian blur* ($\sigma = 12$ px) drops accuracy only to 16.7% ($\Delta = -5.3$ pt, $p = 8.5 \times 10^{-5}$) significant but below our pre-registered ten-point threshold. VLM grounding appears to rely more on global layout than on fine-grained OCR of text labels, since removing high-frequency detail removes only a modest fraction of the signal. However, blur alone cannot disentangle failures in OCR, spatial encoding, and cross-modal binding; targeted ablations are discussed in Appendix S.

5.2 Overall accuracy and the human gap

Table 2 reports the headline numbers, and Figure 4 visualizes the gap from humans.

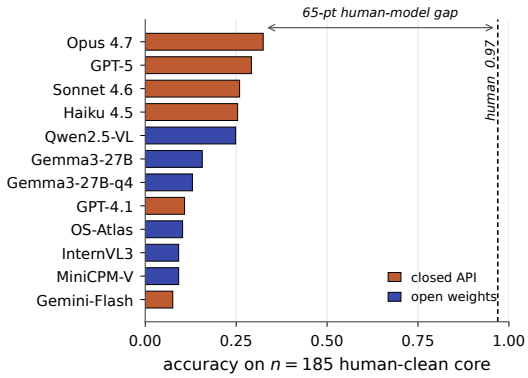

Figure 4: Human-clean core (185 items, 5 annotators, $\kappa = 0.94$). Human accuracy is 96.9%. The strongest model is Claude Opus 4.7 at 32.4%.

All evaluated models perform substantially below the annotator target-selection accuracy of 96.9%, though the two protocols are not matched (§3.4): the strongest system (Claude Opus 4.7) reaches 31–32% strict point-in-box accuracy and the 65-point gap holds on both the full benchmark and the cleaned 185-item core. Scale alone does not close it; the four strongest proprietary models cluster in a narrow 24–31% range, and the 7B open Qwen2.5-VL-Instruct is statistically tied with closed Claude Haiku 4.5 on the human-verified core (24.5% vs. 25.0%) despite being roughly two orders of magnitude smaller and runnable on a single GPU. The rank order is stable across the full 994-item benchmark, the 196-item human-verified core, and an LLM-judge-verified 100-item subset (Appendix A), so the gap is not a noisy-item artifact.

5.3 Per-primitive structure of failure

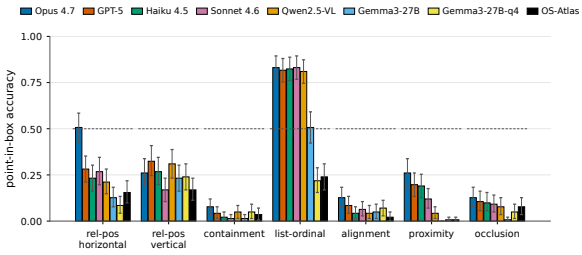

Figure 5: Per-primitive accuracy across eight representative models, with bootstrap 95% confidence intervals. Dashed lines mark the two-candidate reference level (0.50 for all primitives; Table 1). On `containment`, `occlusion`, `alignment`, and `proximity`, every model falls below this reference, including the four strongest proprietary models; candidate-level analysis (Appendix R) attributes this below-reference accuracy primarily to predictions outside both candidate regions.

Figure 5 unpacks the overall accuracy by primitive (Appendix N.2). The picture is uneven and informative. One primitive, `list-ordinal`, is well solved by the top five models (Claude family, GPT-5, and Qwen2.5-VL), all achieving 0.80–0.83 against the 0.50 two-candidate reference (Table 1). The shared task here is counting list positions, which the language model can do once it identifies the list. The other six primitives are uniformly hard. On `rel-pos-horizontal`, Claude Opus 4.7 reaches 0.51, at the reference level; every other model is below. On `rel-pos-vertical`, the best model is GPT-5 at 0.32. On `containment`, `occlusion`, `alignment`, and `proximity`, *the upper 95% bootstrap bound for every model we test sits below the reference level* of 0.50, including all four strongest proprietary models. Because 0.50 is a two-candidate reference level rather than the chance level of unconstrained coordinate prediction, we classify all 18,886 baseline predictions by the candidate region they fall in (Appendix R). Predictions fall outside both candidate regions on 60–92% of items, and among predictions inside a candidate region, the target is selected at least as often as the distractor on every primitive. For `containment` and `occlusion` the within-region selection rate is statistically indistinguishable from 0.50 (0.548 and 0.554), indicating no measurable relation-word signal on these two primitives. For `rel-pos-horizontal`, `rel-pos-vertical`, and

Table 2: Overall accuracy on GUI-PRIMITIVES for 12 representative models, alongside accuracy on the human-clean 185-item subset. The strongest model (Claude Opus 4.7) reaches 32% on the clean core, a 65-point gap below human accuracy (96.9%). All 19 models are in Appendix A.

Model	**Family**	**Full** (n=994)	**Core** (n=196)	**Clean** (n=185)
Claude Opus 4.7	Anthropic	0.313	0.311	0.324
GPT-5 (reasoning)	OpenAI	0.265	0.291	0.297
Claude Sonnet 4.6	Anthropic	0.222	0.260	0.270
Claude Haiku 4.5	Anthropic	0.239	0.250	0.249
Qwen2.5-VL-7B	Qwen2.5-VL	0.220	0.245	0.249
Gemma 3-27B	Google open	0.134	—	0.157
OS-Atlas-Base-7B	Qwen2-VL/GUI	0.101	0.107	0.108
GPT-4.1	OpenAI	0.096	—	0.108
InternVL3-8B	OpenGVLab	0.095	—	0.097
Gemini 3.1 Flash Lite	Google closed	0.073	0.077	0.076
GPT-4o-mini	OpenAI	0.068	0.077	0.076
Qwen2-VL-7B	Qwen2-VL	0.038	0.046	0.049
Llama-3.2-11B-Vision	Meta mllama	0.012	0.005	0.005
Human	—	—	—	**0.969**

`proximity` the target is selected on 0.89–0.90 of within-region predictions, indicating that the deficit is candidate localization rather than relational interpretation; the `alignment` estimate varies with the classification rule (§R) and we do not interpret it. The associated directional prior is quantified in Appendix Q, and its role is discussed in §8. Model-specific coordinate-frame anomalies are also quantified in Appendix Q.

5.4 Real versus synthetic

We next separate two candidate explanations for the per-primitive results in §5.3: small real-GUI target size and the relative proportion of synthetic items. The full strict/loose split by source is in Appendix C and N.4. On synthetic screenshots, *loose* accuracy is 0.60–0.76 even when strict accuracy is much lower, indicating that synthetic errors are largely sub-pixel imprecision rather than element misidentification. On UI-Vision real screenshots, strict accuracy is essentially zero for every model on every primitive, and *loose* accuracy also stays low (6–23%). Candidate-level analysis (Appendix R) shows that 96% of real-screenshot predictions fall outside both the target and distractor regions, while among predictions inside a candidate region, the target is selected on 0.75 pooled (0.91 for the two strongest models). Real-screenshot accuracy is therefore primarily attributable to candidate localization, with target size a contributing factor (Appendix Q), rather than to inverted relational interpretation.

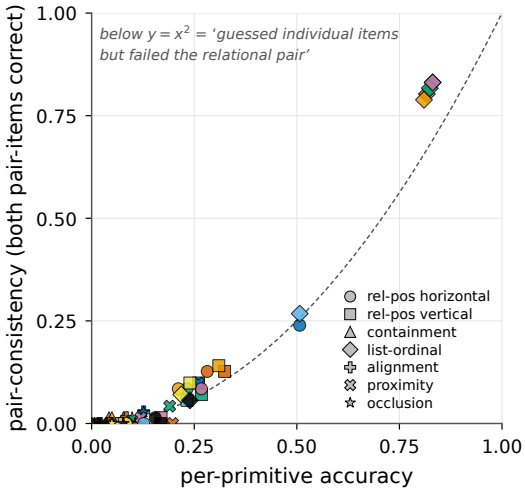

Figure 6: Pair-consistency vs. raw accuracy, per model per primitive. Independent guessing would place each point near $y = x^2$ (dashed). Almost all points lie *below* this curve, the WHAT'S-UP signature of relational failure.

5.5 Pair-consistency: a relational failure signature

Independent draws would put pair-consistency at $y = x^2$ (accuracy squared). Figure 6 shows it consistently *below* that curve, the relational-failure signature of Kamath et al. (2023). Within a model, the two pair members are negatively correlated: being right on the "left of" twin makes the model more likely than chance to be wrong on the "right of" twin. The relation word, not the picture, is what confuses the model (Appendix N.5). Eight representative minimal-pair failures are analyzed in Appendix T.

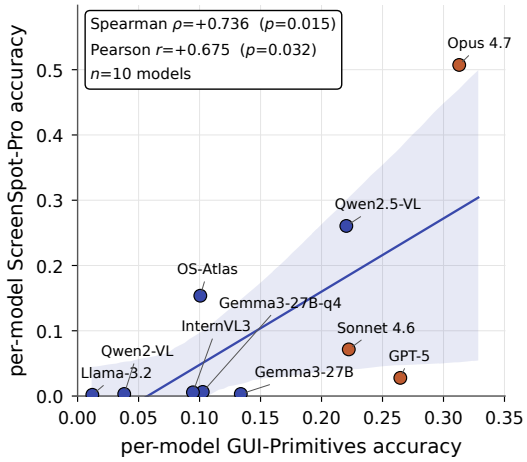

Figure 7: Per-model GUI-PRIMITIVES accuracy ($n_{\text{models}} = 10$) against ScreenSpot-Pro grounding accuracy. Spearman $\rho = +0.736$ ($p = 0.015$).

6 Primitive Competence Predicts Grounding

A diagnostic is only useful if it predicts a downstream property of interest. We test this on ScreenSpot-Pro (Li et al., 2025), the standard high-resolution GUI-grounding benchmark, which we ran on the ten models for which we have both GUI-PRIMITIVES and SS-Pro predictions. Figure 7 shows the per-model relationship. Spearman $\rho = +0.736$ ($p = 0.015$, $n = 10$); the correlation holds inside the real-screenshot subset ($\rho = +0.705$, $p = 0.023$) and inside the synthetic subset ($\rho = +0.760$, $p = 0.011$). A leave-one-model-out sensitivity analysis gives a Spearman in $[+0.62, +0.86]$; no single model drives the result, though dropping the strongest or weakest raises p to ≈ 0.07. An item-level logistic regression with model fixed effects (details in Appendix O) and a log target-area control reaches pseudo-$R^2 = 0.404$ on $n = 15{,}810$ pairs and corroborates the joint association; per-primitive coefficients are small ($|\text{coef}| \leq 0.07$) and fragile, so we claim only the model-level effect (full forest plot in Appendix D).

7 Training-Free Interventions

If models lack the elementary spatial skills GUI grounding needs, can inference-time scaffolding put them back? We test three interventions. **SoM** (Yang et al., 2023) overlays numbered marks on candidate elements; the model names the mark, trading pixel regression for symbolic selection. **Primitive-aware CoT** prepends a short decomposition prompt that names the relation explicitly, probing whether the failure is a reasoning sequence rather than perception. **Activation steering** follows SteerVLM (Sivakumar et al., 2025): we compute mean-difference vectors between correctly- and incorrectly-grounded items on a 200-item held-out contrast set, then add $\alpha \mathbf{v}_\ell$ to the residual stream at decoder layers $\ell \in \{12, \ldots, 19\}$ of Qwen2.5-VL-7B at inference, with $\alpha = 4.0$ chosen as the largest value from $\{1, 2, 4, 6\}$ preserving fluency. As a positive control, the same pipeline with a verbosity-contrast vector shifts output length as expected, so the machinery is functional.

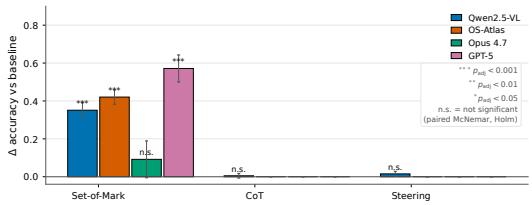

Figure 8: Intervention deltas vs. each model's baseline, with Holm-adjusted significance (paired McNemar across primitives). SoM recovers 35–57 points on three of four tested models; the only marginal case is Claude Opus 4.7, which already has the highest baseline. CoT and activation steering yield no significant change.

7.1 Oracle two-candidate marking

Figure 8 reports intervention deltas with significance. Oracle two-candidate marking overlays marks on exactly the target and distractor, so the baseline and marked conditions are not identical-task comparisons: marking removes the candidate-search burden that dominates baseline error (§5.3). It is the only intervention producing large, statistically significant gains: +57.1 pt on **GPT-5** ($30 \rightarrow 87\%$, $p < 10^{-50}$ the highest absolute accuracy any model reaches under any condition), +42.1 pt on **OS-Atlas-Base-7B** ($10 \rightarrow 52\%$, $p < 10^{-50}$), +35.1 pt on **Qwen2.5-VL-7B** ($22 \rightarrow 57\%$, $p < 10^{-50}$), and a marginal +9.2 pt on **Claude Opus 4.7** ($31 \rightarrow 40\%$, Holm-adjusted $p = 0.08$, not significant). Across the four evaluated models, the observed gain is inversely ordered with baseline accuracy; given four points, we treat this as descriptive rather than a general relationship.

Per-primitive pattern. Figure 9 breaks the gain down by primitive on the two closed APIs we ran with SoM. Six of seven primitives gain at least 40 points on GPT-5; the only primitive that *regresses* (by −32.4 points) is `list-ordinal` on Qwen2.5 VL, which the model had already saturated at 81%

baseline. Converting an already-solved task into mark selection introduces an indirect cost that can reduce accuracy on primitives the model has already saturated.

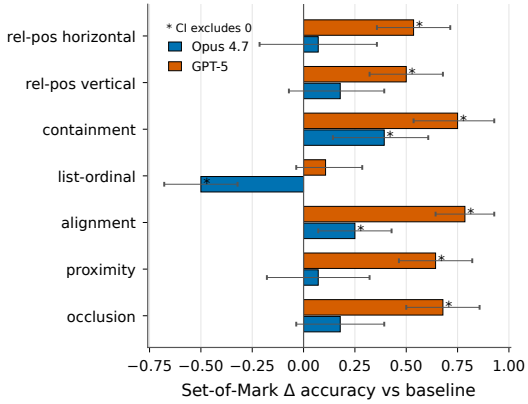

Figure 9: Per-primitive SoM delta on Claude Opus 4.7 and GPT-5. Stars mark intervals excluding zero. On GPT-5, six of seven primitives gain at least 40 points.

7.2 CoT and activation steering: negative results

We report two negative results with the same effort as the positive one (Appendix M). Primitive-aware CoT on Qwen2.5-VL-7B produced $\Delta = +0.5$ pt ($p = 0.51$, n.s.); the model already names the relation correctly in its intermediate reasoning, then still emits the wrong coordinate. Activation steering produced $\Delta = +1.5$ pt ($p = 0.066$, n.s.), with no configuration in the grid crossing the Holm threshold. The one intervention that works bypasses the spatial skill it should test; the two that probe internal representation directly do not move the metric. We read this as evidence that the failure lies closer to perception (visual encoder, cross-modal binding) than to language-side reasoning or residual-stream representation.

8 Discussion

Accuracy does not increase monotonically with nominal model scale in the evaluated set. The strongest proprietary models achieve only 24–31% accuracy. Spatial grounding in GUI screenshots remains a persistent weakness of current vision-language systems. Oracle two-candidate marking provides the largest training-free improvement we observe, but it supplies the candidate set rather than repairing the skill gap, and is therefore a diagnostic upper bound rather than a solution. By marking candidate elements, SoM shifts part of the relational grounding problem into symbolic mark selection. Its effectiveness also depends on the quality of the upstream detector, whose failure modes are discussed in Appendix I. Candidate-level analysis (Appendix R) indicates that below-reference accuracy is primarily attributable to predictions outside both candidate regions (60–92% of predictions) rather than to systematic selection of the contrastive element; within-region selection follows the relation word on all primitives except containment and occlusion, where it is statistically indistinguishable from 0.50. The associated layout-frequency prior is directional: predictions fall within a candidate region 2.7× more often when the target is the left element, and within-region accuracy is 0.972 under "left of" versus 0.717 under "right of" (Appendix Q). We report this prior as a hypothesis for the source of the directional asymmetry rather than as a causal claim. The same decomposition is consistent with the magnitude of the SoM gain: SoM supplies candidate regions directly, addressing the candidate-localization component that accounts for most baseline errors. Appendix U presents six recurring error types with representative examples. Finally, the failure pattern mirrors relational errors observed in WHAT'S-UP for natural images (Kamath et al., 2023). This suggests that controlled minimal-pair diagnostics are essential for evaluating spatial competence across visual modalities, including GUI grounding.

9 Limitations

This work has several limitations (Appendix V). **(i)** Containment and occlusion appear only in the synthetic arm of GUI-PRIMITIVES (list-ordinal includes a 24-item real slice), since real GUI datasets lack scalable parent–child or overlay–target metadata for minimal-pair mining; findings on these primitives therefore have limited external validity, and results are reported separately by source arm. This scoping bounds the candidate-level finding of Appendix R: the absence of a measurable relation-word signal on containment and occlusion is established on synthetic screenshots only. **(ii)** After model fixed effects absorb between-model variance, per-primitive coefficients in our item-level regression are small ($|\text{coef}| \leq 0.07$); we claim only the joint model-level association and report the forest plot (Appendix D) for transparency. **(iii)** We stop at ScreenSpot-Pro; end-to-end OSWorld task success (Xie et al., 2024; XLANG Lab, 2025) remains fu-

ture work. **(iv)** GUI-PRIMITIVES v1 covers static pre-action grounding for English click targets only; planned extensions include drag-target prediction, bidirectional and CJK layouts, an OSWorld pre-action slice, and frozen snapshots of shifting closed APIs (Appendix B). **(v)** SoM acts as a scaffold by partially converting relational grounding into symbolic mark selection; our setup overlays at most two candidate marks, so gains may not generalize to realistic multi-candidate scenes, and deployment performance will inherit upstream detector failures. **(vi)** Point-in-box accuracy is binary and coarse: our loose metric (within $2\times$ target diagonal) partly separates sub-pixel imprecision from element misidentification, but neither metric captures structured spatial errors (e.g., anchor-collapse or direction inversion), which we analyze qualitatively in Appendix U. Llama-3.2-11B-Vision emits a default coordinate on $\approx 42\%$ of items (1.2% accuracy), consistent with ScreenSpot-Pro reports of $<2\%$ for generalist VLMs.

Ethical Considerations

Annotators were paid above the local hourly minimum and provided informed consent (Appendix F). UI-Vision screenshots remain under their original license; we redistribute only derived annotations, not the screenshots themselves. GUI-PRIMITIVES is intended to support safer GUI-agent development by exposing failures in elementary spatial grounding before deployment.

References

Boyuan Chen, Zhuo Xu, Sean Kirmani, Brian Ichter, Danny Driess, Pete Florence, Dorsa Sadigh, Leonidas Guibas, and Fei Xia. 2024. SpatialVLM: Endowing vision-language models with spatial reasoning capabilities. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition (CVPR)*.

Kanzhi Cheng, Qiushi Sun, Yougang Chu, Fangzhi Xu, Li YanTao, Jianbing Zhang, and Zhiyong Wu. 2024. 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)*, pages 9313–9332.

Xingyu Fu, Yushi Hu, Bangzheng Li, Yu Feng, Haoyu Wang, Xudong Lin, Dan Roth, Noah A. Smith, Wei-Chiu Ma, and Ranjay Krishna. 2024. BLINK: Multimodal large language models can see but not perceive. In *Proceedings of the European Conference on Computer Vision (ECCV)*.

Boyu Gou, Ruohan Wang, Boyuan Zheng, Yanan Xie, Cheng Chang, Yiheng Shu, Huan Sun, and Yu Su. 2025. Navigating the digital world as humans do: Universal visual grounding for GUI agents. In *Proceedings of the International Conference on Learning Representations (ICLR)*. Oral.

Nils Hoehing, Mayug Maniparambil, Ellen Rushe, Noel E. O'Connor, and Anthony Ventresque. 2025. Understanding space is rocket science — only top reasoning models can solve spatial understanding tasks. *arXiv preprint arXiv:2509.02175*.

Sture Holm. 1979. A simple sequentially rejective multiple test procedure. *Scandinavian Journal of Statistics*, 6(2):65–70.

Amita Kamath, Jack Hessel, and Kai-Wei Chang. 2023. What's "up" with vision-language models? investigating their struggle with spatial reasoning. In *Proceedings of the Conference on Empirical Methods in Natural Language Processing (EMNLP)*.

Seil Kang, Jinyeong Kim, Junhyeok Kim, and Seong Jae Hwang. 2025. Your large vision-language model only needs a few attention heads for visual grounding. *arXiv preprint arXiv:2503.06287*.

J. Richard Landis and Gary G. Koch. 1977. The measurement of observer agreement for categorical data. *Biometrics*, 33(1):159–174.

Kaixin Li, Ziyang Meng, Hongzhan Lin, Ziyang Luo, Yuchen Tian, Jing Ma, Zhiyong Huang, and Tat-Seng Chua. 2025. ScreenSpot-Pro: GUI grounding for professional high-resolution computer use. In *Proceedings of the ACM International Conference on Multimedia (MM)*, pages 8778–8786.

Tal Linzen, Emmanuel Dupoux, and Yoav Goldberg. 2016. Assessing the ability of LSTMs to learn syntax-sensitive dependencies. *Transactions of the Association for Computational Linguistics*, 4:521–535.

Fangyu Liu, Guy Emerson, and Nigel Collier. 2023. Visual spatial reasoning. *Transactions of the Association for Computational Linguistics*, 11:635–651.

Tiange Luo, Lajanugen Logeswaran, Justin Johnson, and Honglak Lee. 2025. Visual test-time scaling for GUI agent grounding. *arXiv preprint arXiv:2505.00684*.

Rebecca Marvin and Tal Linzen. 2018. Targeted syntactic evaluation of language models. In *Proceedings of the Conference on Empirical Methods in Natural Language Processing (EMNLP)*, pages 1192–1202.

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

Shravan Nayak, Xiangru Jian, Kevin Qinghong Lin, Juan A. Rodriguez, Montek Kalsi, Nicolas Chapados, M. Tamer Özsu, Aishwarya Agrawal, David

Vazquez, Christopher Pal, Perouz Taslakian, Spandana Gella, and Sai Rajeswar. 2025. UI-Vision: A desktop-centric GUI benchmark for visual perception and interaction. In *Proceedings of the 42nd International Conference on Machine Learning*, volume 267 of *Proceedings of Machine Learning Research*, pages 45817–45851. PMLR.

Jianing Qi, Jiawei Liu, Hao Tang, and Zhigang Zhu. 2025. Beyond semantics: Rediscovering spatial awareness in vision-language models. *arXiv preprint arXiv:2503.17349*.

Yujia Qin, Yining Ye, Junjie Fang, Haoming Wang, Shihao Liang, Shizuo Tian, Junda Zhang, Jiahao Li, Yunxin Li, Shijue Huang, Wanjun Zhong, Kuanye Li, Jiale Yang, Yu Miao, Woyu Lin, Longxiang Liu, Xu Jiang, Qianli Ma, Jingyu Li, and 16 others. 2025. UI-TARS: Pioneering automated gui interaction with native agents. *arXiv preprint arXiv:2501.12326*.

Anushka Sivakumar, Andrew Zhang, Zaber Hakim, and Chris Thomas. 2025. SteerVLM: Robust model control through lightweight activation steering for vision language models. In *Findings of the Association for Computational Linguistics: EMNLP*, pages 23640–23665.

Shengbang Tong, Ellis Brown, Penghao Wu, Sanghyun Woo, Manoj Middepogu, Sai Charitha Akula, Jihan Yang, Shusheng Yang, Adithya Iyer, Xichen Pan, Austin Wang, Rob Fergus, Yann LeCun, and Saining Xie. 2024. Cambrian-1: A fully open, vision-centric exploration of multimodal LLMs. In *Advances in Neural Information Processing Systems (NeurIPS)*.

Xinyuan Wang, Bowen Wang, Dunjie Lu, Junlin Yang, Tianbao Xie, Junli Wang, Jiaqi Deng, Xiaole Guo, Yiheng Xu, Chen Wu, Zhennan Shen, Zhuokai Li, Ryan Li, Xiaochuan Li, Junda Chen, Boyuan Zheng, Peihang Li, Fangyu Lei, Ruisheng Cao, and 15 others. 2025a. OpenCUA: Open foundations for computer-use agents. In *Advances in Neural Information Processing Systems (NeurIPS)*, volume 38.

Xuehui Wang, Zhenyu Wu, JingJing Xie, Zichen Ding, Bowen Yang, Zehao Li, Zhaoyang Liu, Qingyun Li, Xuan Dong, Zhe Chen, Weiyun Wang, Xiangyu Zhao, Jixuan Chen, Haodong Duan, Tianbao Xie, Chenyu Yang, Shiqian Su, Yue Yu, Yuan Huang, and 9 others. 2025b. MMBench-GUI: Hierarchical multi-platform evaluation framework for GUI agents. *arXiv preprint arXiv:2507.19478*.

Hang Wu, Hongkai Chen, Yujun Cai, Chang Liu, Qingwen Ye, Ming-Hsuan Yang, and Yiwei Wang. 2025a. DiMo-GUI: Advancing test-time scaling in GUI grounding via modality-aware visual reasoning. In *Proceedings of the Conference on Empirical Methods in Natural Language Processing (EMNLP)*, pages 26257–26267.

Qianhui Wu, Kanzhi Cheng, Rui Yang, Chaoyun Zhang, Jianwei Yang, Huiqiang Jiang, Jian Mu, Baolin Peng, Bo Qiao, Reuben Tan, Si Qin, Lars Liden, Qingwei Lin, Huan Zhang, Tong Zhang, Jianbing Zhang, Dongmei Zhang, and Jianfeng Gao. 2025b. GUI-Actor: Coordinate-free visual grounding for GUI agents. In *Advances in Neural Information Processing Systems (NeurIPS)*, volume 38.

Zhiyong Wu, Zhenyu Wu, Fangzhi Xu, Yian Wang, Qiushi Sun, Chengyou Jia, Kanzhi Cheng, Zichen Ding, Liheng Chen, Paul Pu Liang, and Yu Qiao. 2024. OS-Atlas: A foundation action model for generalist GUI agents. *arXiv preprint arXiv:2410.23218*.

Tianbao Xie, Jiaqi Deng, Xiaochuan Li, Junlin Yang, Haoyuan Wu, Jixuan Chen, Wenjing Hu, Xinyuan Wang, Yuhui Xu, Zekun Wang, Yiheng Xu, Junli Wang, Doyen Sahoo, Tao Yu, and Caiming Xiong. 2025. Scaling computer-use grounding via user interface decomposition and synthesis. In *Advances in Neural Information Processing Systems (NeurIPS)*, volume 38. Spotlight, Datasets and Benchmarks.

Tianbao Xie, Danyang Zhang, Jixuan Chen, Xiaochuan Li, Siheng Zhao, Ruisheng Cao, Toh Jing Hua, Zhoujun Cheng, Dongchan Shin, Fangyu Lei, Yitao Liu, Yiheng Xu, Shuyan Zhou, Silvio Savarese, Caiming Xiong, Victor Zhong, and Tao Yu. 2024. OSWorld: Benchmarking multimodal agents for open-ended tasks in real computer environments. In *Advances in Neural Information Processing Systems (NeurIPS)*, volume 37.

XLANG Lab. 2025. Introducing OSWorld-Verified. XLANG Lab blog post, 28 July 2025.

An Yan, Zhengyuan Yang, Junda Wu, Wanrong Zhu, Jianwei Yang, Linjie Li, Kevin Lin, Jianfeng Wang, Julian McAuley, Jianfeng Gao, and Lijuan Wang. 2024. List items one by one: A new data source and learning paradigm for multimodal LLMs. In *Conference on Language Modeling (COLM)*.

Jianwei Yang, Hao Zhang, Feng Li, Xueyan Zou, Chunyuan Li, and Jianfeng Gao. 2023. Set-of-mark prompting unleashes extraordinary visual grounding in GPT-4V. *arXiv preprint arXiv:2310.11441*.

Yan Yang, Dongxu Li, Yutong Dai, Yuhao Yang, Ziyang Luo, Zirui Zhao, Zhiyuan Hu, Junzhe Huang, Amrita Saha, Zeyuan Chen, Ran Xu, Liyuan Pan, Silvio Savarese, Caiming Xiong, and Junnan Li. 2025a. GTA1: GUI test-time scaling agent. *arXiv preprint arXiv:2507.05791*.

Yuhao Yang, Yue Wang, Dongxu Li, Ziyang Luo, Bei Chen, Chao Huang, and Junnan Li. 2025b. Aria-UI: Visual grounding for GUI instructions. In *Findings of the Association for Computational Linguistics: ACL*, pages 22418–22433.

Yu Zhao, Wei-Ning Chen, Huseyin Atahan Inan, Samuel Kessler, Lu Wang, Lukas Wutschitz, Fangkai Yang, Chaoyun Zhang, Pasquale Minervini, Saravan Rajmohan, and Robert Sim. 2025. Learning GUI grounding with spatial reasoning from visual feedback. *arXiv preprint arXiv:2509.21552*.

A Full Per-Model Results

Table 3 reports overall accuracy for all 19 models on the full benchmark, the 196-item human-verified core, and the 185-item human-clean core. Per-primitive accuracies with bootstrap 95% CIs for every cell are released alongside the artifact in `runs/diagnostic/analysis.json`. A model-by-primitive accuracy heatmap is shown in Figure 10.

Table 3: Full benchmark accuracy for all 19 evaluated models on three splits: the full n=994 benchmark (**Full**), the human-verified n=196 subset (**Core**), and the cleaned n=185 subset after removing items judged invalid by a majority of the five annotators (**Clean**). Closed models were not run on the full split because of API cost; the Core column reports every model that was. Per-primitive accuracies with bootstrap intervals are released in the artifact.

Model	Full	Core	Clean
Claude Opus 4.7	0.313	0.311	0.324
GPT-5 (reasoning)	0.265	0.291	0.297
Claude Haiku 4.5	0.239	0.250	0.249
Claude Sonnet 4.6	0.222	0.260	0.270
Qwen2.5-VL-7B-Instruct	0.220	0.245	0.249
Gemma 3-27B-it (HF)	0.134	—	0.157
gemma3:27b (ollama, Q4)	0.103	—	0.125
OS-Atlas-Base-7B	0.101	0.107	0.108
GPT-4.1	0.096	—	0.108
InternVL3-8B	0.095	—	0.097
MiniCPM-V (ollama)	0.081	—	0.092
Gemma 3-12B-it (HF)	0.074	—	0.087
Gemini 3.1 Flash Lite	0.073	0.077	0.076
GPT-4o-mini	0.068	0.077	0.076
PaliGemma 2-3B-mix-448	0.041	—	0.054
Qwen2-VL-7B-Instruct	0.038	0.046	0.049
PaliGemma 2-10B-mix-448	0.029	—	0.038
Gemma 3-4B-it (HF)	0.022	—	0.027
Llama-3.2-11B-Vision	0.012	0.005	0.005

B Models, Versions, and Endpoints

Table 4 lists the exact model identifiers used in this paper, together with the endpoint each was queried through. Closed models were accessed via their official APIs in May 2026; open models were run from Hugging Face checkpoints. We pin identifiers here so later runs can be compared against the snapshot reported in this paper.

C Real vs. Synthetic: Strict and Loose Accuracy

Table 5 reports point-in-box (strict) and within-2×-diagonal (loose) accuracy split by source for seven representative models. The zero strict accuracy on UI-Vision real screenshots is not a target-size artifact: loose accuracy stays low (6–23%), so the model picks the wrong element rather than missing the right one by a few pixels.

D Item-Level Regression Forest Plot

Figure 11 shows the item-level logistic regression of ScreenSpot-Pro success on per-primitive GUI-PRIMITIVES competence (n = 15,810, pseudo-$R^2 = 0.404$), with model fixed effects, a log target-area control, and item-clustered standard errors. After model effects absorb between-model variance, individual primitive coefficients are small ($|\text{coef}| \leq 0.07$) and only two cross the Holm-corrected threshold, so we report this plot for transparency only. **The headline number is the model-level Spearman in Fig. 7 of the main paper, not the per-primitive coefficients.**

E Closed-API Coordinate Rescaling

All three closed-vision APIs we tested apply server-side image preprocessing whose output is not visible to the caller. The predicted click coordinates are returned in the *downsampled* frame rather than the original pixel space, silently misaligning the prediction with the ground-truth bounding box.

Detection. On the first GPT-5 ScreenSpot-Pro run, we observed predictions like $(662, 541)$ on items whose ground-truth boxes were at $(1774, 1586, 2113, 1618)$ in a 3840×2160 image. The prediction was internally consistent with a 1024-side downsample; the ground truth was in the original frame.

Mitigation. Our wrapper now resizes every image to a known maximum long side ($\leq$ 1500 px) before submission, then rescales the predicted coordinate back to the original space using the recorded resize factor. Where the provider does further internal tiling (notably OpenAI), we additionally validate by re-querying on the already-downsized image and confirming the rescale factor is $\approx$ 1.0. Records that required rescaling: 42/196 Claude Haiku diagnostic, 43/196 Claude Opus diagnostic, 219/994 Claude Sonnet diagnostic, 1,297/1,581 Claude Sonnet ScreenSpot-Pro, and $\approx 1500/1581$ GPT-5 ScreenSpot-Pro. The 284 Claude Sonnet ScreenSpot-Pro items that originally errored with HTTP 400 ("image > 5 MB") were retried under the new wrapper, and the final run is clean.

Recommendation. We recommend that future closed-VLM evaluations report whether their wrapper handles provider-side downsampling. Other-

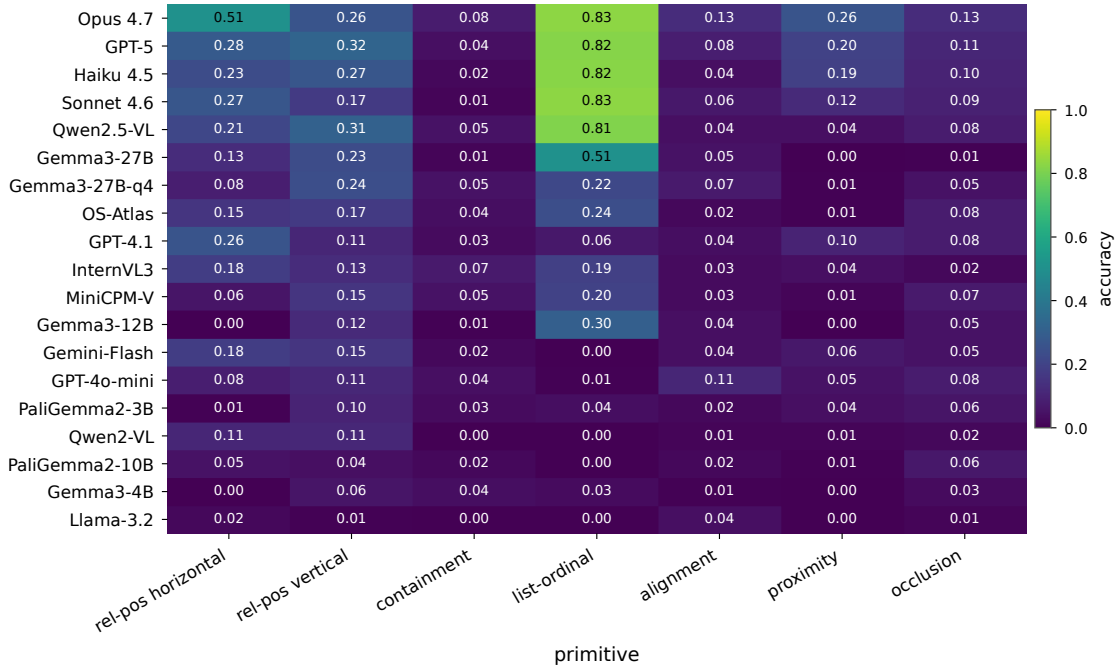

Figure 10: Per-model per-primitive accuracy heatmap, all 19 models. The horizontal band at ∼0.80 on `list-ordinal` marks the only primitive any model solves reliably.

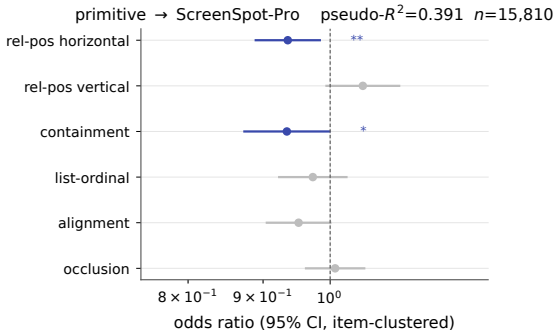

Figure 11: Item-level regression forest plot ($n = 15{,}810$, pseudo-$R^2 = 0.404$).

wise, ScreenSpot-Pro's strict accuracy on high-resolution professional applications is silently and severely under-reported.

F Human Annotation Protocol

Five annotators independently labeled the 196-item stratified subset. Each annotator answered two questions per item.

Q1: well-formedness. *Is the instruction unambiguous on this screenshot?* The annotator answered `good` or `bad`. An item was kept only if at least 3 of 5 annotators marked it as well-formed. This dropped 11 of 196 items, leaving the 185-item *human-clean* core.

Q2: correct target. *Which of the two candidate boxes does the instruction identify?* The annotator answered `t` (the target box) or `d` (the distractor box). Human accuracy is the fraction of items on which the majority answer was `t`, computed on the human-clean core only.

Agreement. We report Fleiss κ for both questions on the full 196-item subset. Fleiss $\kappa_{Q1} = 0.942$ (well-formedness, "almost perfect"); Fleiss $\kappa_{Q2} = 0.787$ (target, "substantial"). The pairwise κ matrix is released with the artifact.

Compensation and consent. Annotators were paid above the local hourly minimum, were told the purpose of the labeling (academic publication and public benchmark release), and could opt out at any point. No personally-identifying information was collected.

G Synthetic-Stimulus Generation

The synthetic arm of GUI-PRIMITIVES renders GUI-like screenshots with a fixed layout language: a top toolbar of buttons, a left sidebar of list items, a central grid of tiles, and an optional dialog overlay. Element positions are sampled uniformly within constrained pixel ranges; the seed is fixed to ensure reproducibility of the corpus. For each primitive, a separate generator constructs minimal pairs by selecting an anchor and the two candidate elements

Table 4: Model identifiers used in this paper. Closed models were queried via official APIs in May 2026; open models were run from Hugging Face checkpoints with greedy decoding ($T = 0$) and a fixed seed.

Model	Identifier	Family / Endpoint
Claude Opus 4.7	`claude-opus-4-7`	Anthropic API
Claude Sonnet 4.6	`claude-sonnet-4-6`	Anthropic API
Claude Haiku 4.5	`claude-haiku-4-5`	Anthropic API
GPT-5 (reasoning)	`gpt-5`	OpenAI API
GPT-4.1	`gpt-4.1`	OpenAI API
GPT-4o-mini	`gpt-4o-mini`	OpenAI API
Gemini 3.1 Flash Lite	`gemini-3.1-flash-lite`	Google API
Qwen2.5-VL-7B-Instruct	`Qwen/Qwen2.5-VL-7B-Instruct`	Hugging Face
Qwen2-VL-7B-Instruct	`Qwen/Qwen2-VL-7B-Instruct`	Hugging Face
OS-Atlas-Base-7B	`OS-Copilot/OS-Atlas-Base-7B`	Hugging Face
InternVL3-8B	`OpenGVLab/InternVL3-8B`	Hugging Face
Llama-3.2-11B-Vision	`meta-llama/Llama-3.2-11B-Vision-Instruct`	Hugging Face
Gemma 3-27B-it	`google/gemma-3-27b-it`	Hugging Face
Gemma 3-12B-it	`google/gemma-3-12b-it`	Hugging Face
Gemma 3-4B-it	`google/gemma-3-4b-it`	Hugging Face
PaliGemma 2-10B-mix-448	`google/paligemma2-10b-mix-448`	Hugging Face
PaliGemma 2-3B-mix-448	`google/paligemma2-3b-mix-448`	Hugging Face
MiniCPM-V (ollama)	`minicpm-v`	Ollama (Q4)
gemma3:27b (ollama)	`gemma3:27b`	Ollama (Q4)

Table 5: Strict (point-in-box) vs. loose (within 2× target-diagonal) accuracy, split by image source. In real desktop screenshots, loose accuracy stays low even when the prediction is allowed to miss by a generous margin: failure is element misidentification, not sub-pixel imprecision.

	Real (UI-Vision)		Synthetic	
Model	strict	loose	strict	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

that satisfy the contrastive relation. Quality gates in the released code verify that both members of every pair use the same screenshot, have non-empty disjoint targets, and pass per-primitive coverage checks before items enter the benchmark.

H Intervention Implementation Details

SoM. We render mark IDs on the candidate boxes (two per item: target and distractor) with a deterministic ID shuffle keyed by the item ID so the correct answer is not always "1". The model is asked to respond with the mark number to click, and we map the response back to the box center for evaluation.

Primitive-aware CoT. We prepend a one-paragraph decomposition prompt that names the relation explicitly and asks for a chain of reasoning before the click. Per-primitive scaffolds are released in the artifact.

Activation steering. We hook the decoder layers of Qwen2.5-VL-7B at layers $\{12, \ldots, 19\}$ and add $\alpha \cdot \mathbf{v}_\ell$ to the residual stream at each layer, where $\mathbf{v}_\ell$ is the mean-difference vector between hidden states on correctly- and incorrectly-grounded baseline items at that layer. We searched $\alpha \in \{1, 2, 4, 6\}$; $\alpha = 4$ was the maximum that left sentence-level fluency intact.

I SoM Failure Modes

SoM replaces the original grounding task with a mark-selection task. The substitution is not perfect, and the failure modes are worth listing for downstream users.

Saturated primitive regression. On Qwen2.5-VL-7B, `list-ordinal` accuracy drops by 32.4 points under SoM. The model had already learned to count list items by their on-screen order; converting the same task to "select mark X" adds an indirection that the model handles less well than the original.

Mark placement collisions. On highly crowded screenshots, two marks placed at nearby element centers can occlude each other, or the model can confuse them. We did not observe this on the small candidate sets used in our benchmark (typically 2

marks per item), but it is a known issue at scale.

Closed-API consistency. On the human-verified core, Claude Opus 4.7 gained only $+9.2$ points under SoM, in contrast to GPT-5's $+57.1$ points. This 47-point gap on the same items under the same intervention suggests that Opus already grounds the unmarked image with the same logic SoM provides, while GPT-5 was genuinely held back by an aspect of the task that mark selection removes.

J Details of the GUI-Primitives Benchmark

J.1 Details of minimal-pair construction

Every item in GUI-PRIMITIVES appears as one half of a contrastive pair. A pair shares one screenshot and one anchor element; the relation expression changes between the two members so that the correct target moves to the other designated candidate. In 253 of 497 pairs, the surrounding instruction template also varies lexically in verb, head noun, and anchor quoting; this variation is not confounded with the relation term (Appendix J.2). Figure 2 shows a real example: in (A) the click should land to the right of the `break debugger` icon; in (B) the same screenshot is paired with `left of`, and the correct target is the icon on the other side. The construction controls for screenshot-specific salience and fixed answer preferences: a model that always selects the most central, salient, or frequently labeled element scores correctly on one twin and incorrectly on the other, so its pair-level score is zero. We measure this directly with *pair-consistency*: the fraction of pairs on which both members are correct (§5.5).

J.2 Details of sources: real and synthetic screenshots

GUI-PRIMITIVES draws screenshots from two sources, both reported separately.

Real desktop screenshots (n = 290 items). We use the element-grounding annotations from UI-Vision (Nayak et al., 2025) as the source corpus for real GUIs, covering Web, MacOS, VS Code, Office, and various professional applications. Items synthesized from this source cover the four primitives whose annotations are unambiguously recoverable from element labels: horizontal relative position, vertical relative position, alignment, and proximity.

Controlled synthetic stimuli (n = 704 items). The remaining items are rendered from a procedural generator that places elements in known positions, with controlled distractors and a fixed layout language. This is the only way to obtain *containment*, *list ordinal*, and *occlusion* items in clean form, because those three primitives require a structured parent–child or overlay–target relationships that real screenshot annotations do not ship at scale.

We disclose the split openly. The synthetic arm is a controlled-stimulus arm in the WHAT'S-UP tradition, where the hardest relations are likewise generated under controlled rendering. All real vs. synthetic numbers are reported separately throughout the paper.

J.3 Details of human verification

Five annotators independently judged a 196-item stratified subset (*human-verified core*). For each item, the annotator answered two questions: is the instruction well-formed and unambiguous on the given screenshot, and which of the two candidate boxes is the correct target. Inter-annotator agreement is high. On Q1 (well-formedness), Fleiss $\kappa = 0.942$ ("almost perfect" on the Landis–Koch scale, Landis and Koch, 1977). On Q2 (correct target), Fleiss $\kappa = 0.787$ ("substantial"). Eleven items were judged invalid by the majority and were dropped, leaving 185 items in the *human-clean core*.

K Benchmark Distribution and Scenario Diversity

Figure 12 summarises the distributional properties of GUI-PRIMITIVES across four dimensions.

Relation balance (Panel A). Every contrastive relation is represented by exactly 71 items per side per primitive, a strict 50/50 split enforced by the minimal-pair miner. This guarantees that a model exploiting a relation-frequency prior (e.g., always predicting `left`) cannot achieve better-than-reference-level accuracy at the primitive level, and it rules out skewed-prior explanations for any observed performance gap. Table 6 reports the per-primitive composition by source arm: 142 items per primitive; containment and occlusion are synthetic-only, and list-ordinal includes a 24-item real slice.

Target–distractor spatial separation (Panel B). Target–distractor distances (normalized by the image diagonal) span a wide range: median 0.18, IQR $[0.04, 0.31]$. `list-ordinal` items are tightly spaced (adjacent list rows, median 0.04), while `containment` and `occlusion` items have the

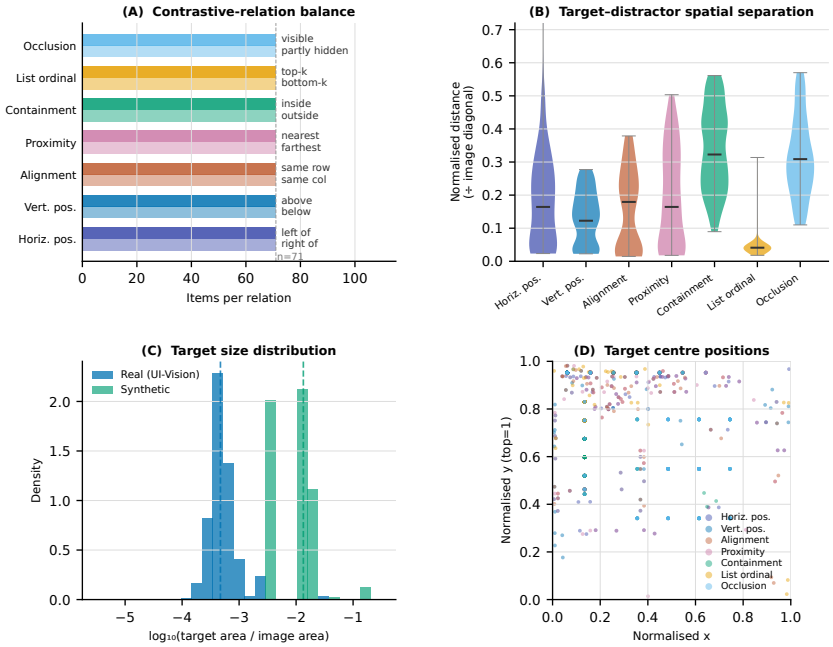

Figure 12: **Benchmark distribution analysis. (A)** Contrastive-relation balance: every relation is represented by exactly 71 items per side, ruling out relation-prior exploitation. **(B)** Target–distractor spatial separation (normalized by image diagonal) per primitive; violin width encodes density. **(C)** Target size distribution (log-scale) by source arm; dashed lines are per-source medians. **(D)** Target centre positions (normalized to $[0, 1]^2$, y=1 at top), coloured by primitive; targets span the full screen area.

Table 6: Per-primitive composition of GUI-PRIMITIVES by source arm. Every primitive has 142 items, exactly 71 per relation word.

Primitive	Real	Synthetic	Total
rel-pos-horizontal	68	74	142
rel-pos-vertical	48	94	142
alignment	66	76	142
proximity	84	58	142
list-ordinal	24	118	142
containment	0	142	142
occlusion	0	142	142
Total	290	704	994

largest separations (median 0.32–0.31). This variation ensures that the benchmark exercises a range of spatial configurations rather than collapsing onto a single difficulty level.

Target size (Panel C). Real UI-Vision targets are considerably smaller than synthetic ones (median 0.05% vs. 1.3% of image area), consistent with the fine-grained element boxes in professional desktop applications. Both distributions span multiple orders of magnitude ($\log_{10}$ range ≈3.5), providing diversity in target granularity within each arm.

Spatial coverage (Panel D). Target centers, normalized to the unit square, span the full screen area with no strong clustering, reflecting a mix of toolbar, sidebar, central panel, and dialog overlay targets across both sources. The slight top-left bias ($\bar{x} = 0.29$, $\bar{y} = 0.24$) reflects the prevalence of toolbar and sidebar elements in real GUI layouts rather than a construction artifact.

L Details of Models and Protocol

Details of models. We evaluate 19 vision-language models: seven proprietary models (Claude Opus 4.7, Claude Sonnet 4.6, Claude Haiku 4.5, GPT-5, GPT-4.1, GPT-4o-mini, Gemini 3.1 Flash Lite); three open Qwen-family models (Qwen2.5-VL-7B, Qwen2-VL-7B, OS-Atlas-Base-7B, which is GUI-tuned Qwen2-VL); InternVL3-8B; Meta Llama-3.2-11B-Vision; three sizes of Google Gemma 3 (4B, 12B, 27B, including an Ollama-Q4 copy); two PaliGemma 2 variants (3B, 10B); and MiniCPM-V. The full list and provenance are in Appendix B. We could not include Pixtral-12B and Idefics3-8B because their chat-template formats are incompatible with our generic wrapper; this is disclosed as a known scope limitation rather than silently omitted.

Details of prompting and decoding. All models use greedy decoding ($T = 0$) and a fixed seed.

The grounding prompt is a single short instruction ("respond with only the pixel coordinate to click"). Per-family coordinate conventions (pixel, normalized $[0, 1]$, normalized $[0, 1000]$) are handled by a robust parser; the parser is in the released code, and its unit tests are bundled with the artifact.

Details of closed-API coordinate frames. A practical wrinkle complicated the closed-model runs. The three closed providers each downsample large screenshots server-side (Anthropic to ≤ 1568 px, OpenAI to roughly 1024 on the long side, Gemini variable) and return coordinates in the downsampled frame. We pre-resize every image to a known long side and rescale the predicted coordinate back to the original pixel space. The rescale was needed on a large fraction of items (e.g., 1,297 of 1,581 Sonnet ScreenSpot-Pro records). Full details in Appendix E; we believe this is a methodological note worth sharing with the community.

Details of metrics. The primary metric, following ScreenSpot-Pro, is *point-in-box* accuracy: a prediction $\hat{p} = (\hat{x}, \hat{y})$ is correct iff $\hat{x} \in [x_1, x_2]$ and $\hat{y} \in [y_1, y_2]$ for ground-truth box $B = [x_1, y_1, x_2, y_2]$ in the original pixel frame. *Loose* accuracy (prediction within $2\times$ target diagonal of the box center) separates sub-pixel imprecision from element mis-identification on small icons. All accuracies carry bootstrap 95% CIs (2,000 resamples). Paired comparisons use McNemar (McNemar, 1947) tests with Holm–Bonferroni (Holm, 1979) correction across the seven primitives. All primitives are binary contrasts (reference level 0.50; Table 1). We flag a cell as below the reference level only when the upper bootstrap bound falls below 0.50; thus, a point estimate dipping below it due to sampling noise does not count. Because models emit unconstrained coordinates, 0.50 is a forced-choice reference rather than the chance level of the prediction task (Appendix R). The below-reference flags persist under anchor-level and screenshot-level cluster resampling; the synthetic generator reuses anchor slots (containment: 4 distinct anchors; occlusion: 13; alignment: 34; proximity: 43, each over 142 items and 71 screenshots), and no clustered upper bound reaches 0.50 for any of the 19 models on the four sub-reference primitives.

M Details of CoT and Activation Steering: Negative Results

We report two negative results, with the same effort as the positive one.

Primitive-aware CoT on Qwen2.5-VL-7B produced $\Delta = +0.5$ points (Holm-adjusted $p = 0.51$, not significant). The decomposition prompt does not seem to address the limiting factor: the model correctly names the relation in its intermediate reasoning, yet still emits the wrong coordinate.

Activation steering with SteerVLM-style vectors on Qwen2.5-VL-7B produced $\Delta = +1.5$ points (Holm-adjusted $p = 0.066$, not significant). The grid search over layers and α did not yield a configuration that crossed the Holm threshold. Either GUI grounding requires a different intervention point than the layers where SteerVLM has helped on linguistic tasks, or the residual-stream signal is genuinely weak, and a single contrastive vector is not enough to push the model across the decision boundary on minimal pairs.

Reading the intervention table. The picture is uneven on purpose. The one intervention that works, SoM, works by *bypassing* the spatial skill it should test, replacing it with a symbolic naming task. The two interventions that probe the model's internal representation directly do not move the metric. We read this as evidence that the failure is closer to perception (visual encoder, cross-modal binding) than to language-side reasoning or residual-stream representation.

N Details of Diagnostic Results

N.1 Details of shortcut controls validate the benchmark

A diagnostic is only as good as the shortcuts it forbids. Figure 3 reports three controls on Qwen2.5-VL-7B over the full 994-item benchmark.

In the *text-only* condition, the instruction is paired with a blank canvas of the same size: accuracy falls from 22.0% to 6.5% ($\Delta = -15.5$ pt, McNemar $p = 2.8 \times 10^{-27}$). In the *shuffled* condition, the screenshot is replaced with one from a different item of the same primitive: accuracy drops to 14.8% ($\Delta = -7.2$ pt, $p = 1.3 \times 10^{-11}$). Both controls behave as a truly grounded model should: when the visual or the matching instruction is removed, the model can no longer solve the task.

The third control, *heavy Gaussian blur* ($\sigma = 12$ px), drops accuracy to 16.7% ($\Delta = -5.3$ pt,

$p = 8.5 \times 10^{-5}$). The drop is statistically significant but below our pre-registered ten-point threshold, so under our pre-registered criterion the blur control does not reach a strict pass. The result indicates that VLM grounding relies more on global layout than on fine-grained OCR of text labels, since removing high-frequency detail removes only a modest fraction of the signal.

N.2 Details of overall accuracy and the human gap

Table 2 reports the headline numbers. Figure 4 visualizes the gap from human performance. Three observations follow.

First, all models perform well below the annotator target-selection accuracy of 96.9% on a two-candidate task, which is not matched to model point-in-box accuracy (§3.4). The strongest system, Claude Opus 4.7, reaches 31–32%. The 65-point gap holds on the full benchmark and on the cleaned 185-item core alike.

Second, accuracy does not increase monotonically with nominal scale. The four strongest proprietary models (Claude Opus 4.7, GPT-5, Claude Sonnet 4.6, Claude Haiku 4.5) occupy the top of the table but cluster within a narrow 24–31% range. Meanwhile, the 7B open Qwen2.5-VL-Instruct is statistically tied with the closed Claude Haiku 4.5 on the human-verified core (24.5% vs. 25.0%), a noteworthy parity finding given that Qwen2.5-VL is roughly two orders of magnitude smaller than the closed reference and runs on a single GPU.

Third, the rank order is stable. The same top group leads on the full 994-item benchmark, on the 196-item human-verified core, and on the LLM-judge-verified 100-item subset (Appendix A). This means the human gap is not an artifact of noisy items.

N.3 Details of per-primitive structure of failure

Figure 5 unpacks the overall accuracy by primitive. The picture is uneven and informative.

One primitive, `list-ordinal`, is well solved by the top five models (Claude family, GPT-5, and Qwen2.5-VL), all achieving 0.80–0.83 against the 0.50 two-candidate reference (Table 1). The shared task here is counting list positions, which the language model can do once it identifies the list.

The other six primitives are uniformly hard. On `rel-pos-horizontal` Claude Opus 4.7 reaches 0.51, at the reference level; every other model is below. On `rel-pos-vertical`, the best model is GPT-5 at 0.32. On `containment`, `occlusion`, `alignment`, and `proximity`, *the upper 95% bootstrap bound for every model we test sits below the reference level* of 0.50, including all four strongest proprietary models. Candidate-level analysis (Appendix R) indicates that this below-reference accuracy is primarily attributable to predictions outside both candidate regions (60–92% of predictions) rather than to selection of the contrastive element: within-region selection favors the target at least as often as the distractor on every primitive, with containment and occlusion statistically indistinguishable from 0.50 (0.548 / 0.554) and horizontal, vertical, and proximity at 0.89–0.90. The associated directional prior is discussed in §8.

N.4 Details of real versus synthetic

We separate two candidate explanations for the per-primitive results in §5.3: small real-GUI target size and the relative proportion of synthetic items. Table 5 reports strict and loose accuracy split by source.

On synthetic screenshots, models often predict near the correct element: loose accuracy is 0.60–0.76 even when strict box-edge accuracy is substantially lower, indicating that most synthetic errors are sub-pixel imprecision rather than element misidentification. On UI-Vision real screenshots, strict accuracy is essentially zero for every model on every primitive, and *loose* accuracy also stays low (6–23%). Candidate-level analysis (Appendix R) shows that on the real arm, 96% of predictions fall outside both candidate regions, while among predictions inside a candidate region, the target is selected on 0.75 pooled. Real-screenshot accuracy is therefore primarily attributable to candidate localization, with target size a contributing factor, rather than to inverted relational interpretation.

N.5 Details of pair-consistency: a relational failure signature

If a model produced independent draws on the two members of a pair, pair-consistency would equal $y = x^2$ (accuracy squared). Figure 6 shows that pair-consistency is consistently *lower* than that $y = x^2$ curve. This pattern, called the relational-failure signature in Kamath et al. (2023), means that within a model, the two pair members are negatively correlated: when the model is right on the "left of" twin, it is more likely than chance to be

wrong on the "right of" twin. The information that confuses the model is the relation word, not the picture.

O Details of Primitive Competence Predicts Grounding

A diagnostic is only useful if it predicts a downstream property of interest. We test this on ScreenSpot-Pro (Li et al., 2025), the standard high-resolution GUI-grounding benchmark, and we run it on the ten models for which we have both GUI-PRIMITIVES and SS-Pro predictions.

Figure 7 shows the per-model relationship. Spearman $\rho = +0.736$ ($p = 0.015$, $n = 10$). The correlation holds within the real-screenshot subset ($\rho = +0.705$, $p = 0.023$) and within the synthetic subset ($\rho = +0.760$, $p = 0.011$), so it is not an artifact of any single item type. A leave-one-model-out sensitivity analysis gives a Spearman in $[+0.62, +0.86]$, so no single model drives the result; dropping the strongest or weakest does raise the p-value to ≈ 0.07. We also report a wide non-parametric bootstrap CI ($[+0.11, +0.98]$, $n = 10$) for transparency rather than as a precision claim. This is the load-bearing result of this section.

Item-level corroboration. We also fit an item-level logistic regression of ScreenSpot-Pro success on each item, using the model's per-primitive accuracy profile, model fixed effects, a log target-area control, and item-clustered standard errors. The model achieves pseudo-$R^2 = 0.404$ on $n =$ 15,810 model-item pairs. The forest plot of per-primitive coefficients is in Figure 11.

We urge caution on the per-primitive coefficients. After the model fixed effects absorb between-model variance, the residual between-primitive variance is small (all $|\text{coef}| \leq 0.07$), and only two coefficients are individually significant at the Holm-corrected level. With four distinct open models contributing to ScreenSpot-Pro data, the competence vectors are correlated across primitives, making per-primitive identification fragile. We therefore claim only the joint, model-level association, which is strong, and explicitly do not claim that competence at any particular primitive causes downstream success.

P Additional Minimal-Pair Examples

Figure 2 shows a single `rel-pos-horizontal` pair. To make the minimal-pair construction intuitive for the other six primitives, Figures 14–20 each present *three* representative pairs for one primitive. Every row is a distinct pair; within a row, both panels share the same screenshot, only the relation word in the instruction differs (twin A on the left, twin B on the right), and the colored box marks only that panel's gold target. A relationally-grounded model must point at the two different boxes for the two instructions; a shortcut-grounded model points at the same box for both.

Q Model-Specific Sensitivity Analysis

Figure 13 reports four model-level sensitivity diagnostics computed directly from the existing diagnostic runs.

Left–right asymmetry (Panel A). Across 19 models, 15 score higher on `left of` than `right of` items, with a mean gap of $+0.169$ accuracy points (one-sample $t = 5.6$, $p < .001$, sign test $p < .001$). The bias is present in every proprietary model (Opus, Sonnet, Haiku, GPT-5, Gemini) and in the strongest open models. This is consistent with a left-side layout frequency prior: in Western GUIs, action buttons, primary controls, and frequently clicked elements are disproportionately placed on the left, so models that ground by position salience rather than by relation word will inflate left-side accuracy. The candidate-level analysis (Appendix R) indicates that this asymmetry coexists with relation-word use rather than replacing it: within-region selection is 0.972 under "left of" and 0.717 under "right of", and predictions fall within a candidate region 2.7× more often when the target is the left element. The accuracy asymmetry reported here and the within-region asymmetry in Appendix R are two measurements of the same directional effect. The two exceptions, MiniCPM-V and PaliGemma2-3B, are low-accuracy models that default to near-origin coordinates on many items (Panel D), effectively averaging away the bias.

Above–below asymmetry (Panel B). The below-over-above preference is weaker and not statistically significant (mean $\Delta = +0.047$, $t = 1.7$, $p = .10$, 12/19 models below-biased). OS-Atlas is a notable outlier with a strong *above* preference ($\Delta = -0.28$), suggesting a GUI-specific training prior that differs from the natural-image convention.

Target-size sensitivity (Panel C). Pooled across all models and primitives, large targets ($>$ 75th

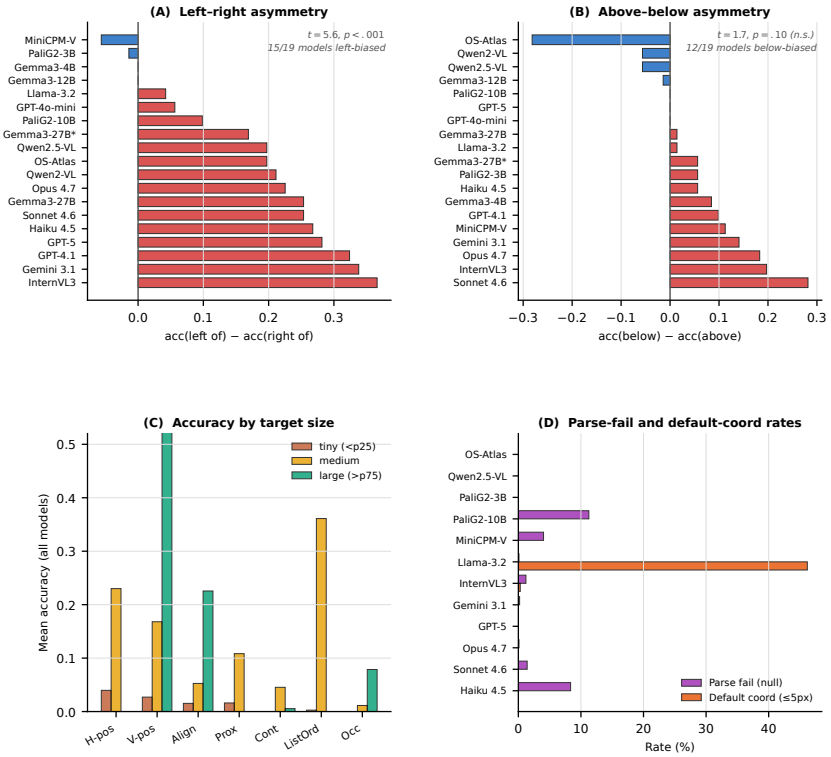

Figure 13: **Model sensitivity analysis. (A)** Left–right accuracy gap per model, sorted by magnitude. Red bars = left-biased (15/19 models; $t = 5.6$, $p < .001$). **(B)** Above–below accuracy gap. Weaker and not significant overall ($p = .10$); OS-Atlas is an outlier with a strong above preference. **(C)** Mean accuracy stratified by target normalized area (tiny/medium/large) per primitive, pooled over all 19 models. **(D)** Parse-fail (null output) and default-coordinate rates per model; both are infrastructure errors, not spatial-reasoning failures.

percentile of normalized area) reach a mean accuracy of 0.092 vs. 0.022 for tiny targets ($<$ 25th percentile). The effect is largest for `containment` and `list-ordinal`, primitives where the target is a panel or a list row that occupies substantial screen area, and smallest for `alignment` and `proximity`, where the spatial relationship matters more than target visibility. Caution: the size gap is partially confounded with source (synthetic targets are larger), so this should not be interpreted as a pure resolution effect.

Coordinate-frame and parse-failure rates (Panel D). Three distinct failure modes are visible: Llama-3.2-11B-Vision emits a near-origin default on 46% of items (consistent with a "don't know" fallback, not a parser bug); PaliGemma2-10B and Claude Haiku produce null outputs (empty or unparseable text) on 8–11% of items; and InternVL3 and Claude Sonnet show low but non-zero null rates. These are infrastructure failures, not spatial-reasoning failures, and are already annotated as Type C and F errors in Appendix U.

R Candidate-Level Classification of Baseline Predictions

The 0.50 reference level is a two-candidate forced-choice reference rather than the chance level of the prediction task: models emit an unconstrained coordinate, so a prediction can fall outside both candidate regions. To separate selection of the contrastive element from predictions outside both regions, we classify every baseline prediction from all 19 models (18,886 records) geometrically: *target* if the point lies inside the target box, *distractor* if inside the distractor box, *neither* if inside neither, and *invalid* if no coordinate could be parsed. A point inside both boxes is assigned to the nearer box center; this case does not occur (0 of 18,886). Table 7 reports the four-way split pooled over models, together with the within-region selection rate (target as a fraction of predictions inside either region), its denominator, and a two-sided exact binomial test against 0.50.

Selection between candidates. The target is selected at least as often as the distractor on every primitive. The *neither* column dominates through-

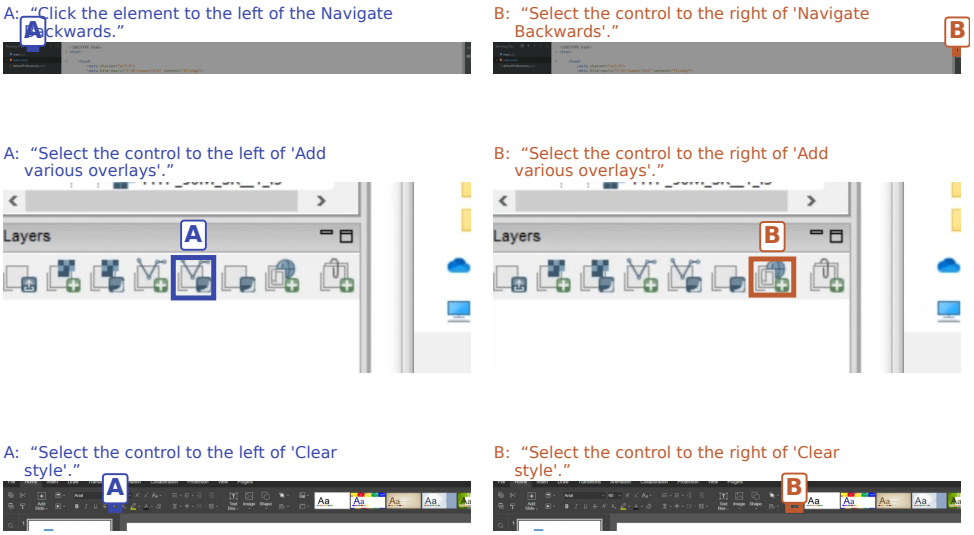

Figure 14: **`rel-pos-horizontal` (real screenshots).** Three minimal pairs from UI-Vision. Within each row, both panels show the same screenshot; only the relation word (*right of* vs. *left of*), and the gold target moves accordingly. Across rows, the application and anchor change.

Table 7: Candidate-level classification of all baseline predictions, pooled over 19 models (strict rule: a prediction is assigned to a candidate only if it lies inside that candidate's box). Sel. = within-region selection rate, target / (target + distractor), reported with its denominator n. p is a two-sided exact binomial test against 0.50. Selection rates computed on a small n (e.g., alignment, $n = 173$) are correspondingly less precise.

Primitive	Tgt	Dis	Neither	Inv.	Sel.	n	p
containment	3.2%	2.6%	91.9%	2.3%	0.548	157	0.26
occlusion	6.1%	4.9%	88.1%	0.8%	0.554	298	0.07
alignment	4.7%	1.7%	92.5%	1.1%	0.728	173	$< 10^{-8}$
proximity	6.1%	0.7%	92.4%	0.7%	0.891	184	$< 10^{-28}$
rel-pos-horizontal	14.8%	1.6%	83.2%	0.4%	0.903	443	$< 10^{-72}$
rel-pos-vertical	16.1%	1.7%	79.2%	3.0%	0.904	481	$< 10^{-79}$
list-ordinal	31.1%	6.8%	60.3%	1.8%	0.820	1022	$< 10^{-99}$

out: 60–92% of predictions fall outside both candidate regions. For `containment` and `occlusion`, within-region selection is statistically indistinguishable from 0.50, indicating no measurable relation-word signal on these two primitives; for `rel-pos-horizontal`, `rel-pos-vertical`, and `proximity`, within-region selection favors the target (0.89–0.90). At the individual-model level, the only statistically significant excess of distractor selection is Gemma3-4B (0.328, $n = 67$); no model shows such an excess under the permissive rule below.

Sensitivity to the inclusion rule. The strict rule yields small denominators, so we repeat the classification under a permissive rule: a prediction is assigned to a candidate if it lies within one box-diagonal of that candidate's box center (normalized per box), and to *neither* otherwise. The two rules agree for six of seven primitives: containment 0.539 ($n = 1680$), occlusion 0.557 ($n = 1256$), and the four target-favoring primitives at 0.69–0.77 ($n = 471$–2075). The `alignment` estimate depends on the rule: 0.728 under the strict rule ($n = 173$, 6% of its predictions) versus 0.529 under the permissive rule ($n = 1021$). We report both values and do not interpret the within-region selection rate for this primitive.

Real-screenshot arm. On UI-Vision screenshots, the *neither* column is larger still: pooled over 19 models, 96.2% of real-arm predictions fall outside both candidate regions (87.8% for GPT-5 and Claude Opus 4.7 alone), while within-region selection is 0.747 pooled ($n = 150$) and 0.914 for GPT-5 and Opus ($n = 70$). Real-screenshot accuracy is thus primarily attributable to candidate localization rather than to inverted relational interpretation.

Comparison with the human task. The human annotation task is a two-candidate forced choice, so the within-region selection rate is the model-side quantity matched to the human accuracy of 96.9%. On the human-clean core, Claude Opus 4.7 produces a within-region prediction on 62 of 185 items and selects the target on 60 of those 62 (0.968); $60/185 = 0.324$ equals the point-in-box accuracy in Table 2, since "prediction inside the target box" is the same event under both definitions. For GPT-5, $54/57 = 0.947$ within-region

and $54/185 = 0.292$ overall. Within-region discrimination for the strongest models (0.95–0.97) is comparable to human accuracy (0.969); the difference in overall accuracy is primarily attributable to the low proportion of predictions ($\approx$ one third) that fall within either candidate region.

Directional asymmetry. Conditioning within-region predictions on the relation word gives $P(\text{target} \mid \text{“left of”}) = 0.972$ ($n = 323$) versus 0.717 ($n = 120$) for “right of”; for the vertical primitive, “below” $= 0.984$ ($n = 253$) versus “above” $= 0.816$ ($n = 228$). Both conditional rates exceed 0.50, indicating that the relational word is used on these primitives. The left-side layout prior to Appendix Q manifests as a difference in the rate at which predictions fall within a candidate region ($2.7\times$ higher when the target is the left element) together with a lower within-region accuracy under “right of”.

Cluster-robustness of the below-reference flags. The synthetic generator reuses anchor slots (containment: 4 distinct anchors; occlusion: 13; alignment: 34; proximity: 43; 71 screenshots per primitive), so we recompute all per-primitive bootstrap intervals with cluster resampling at the anchor level and at the screenshot level. No upper bound reaches 0.50 for any of the 19 models on any of the four sub-reference primitives under either clustering; the widest interval is Claude Opus 4.7 on proximity, anchor-clustered, $[0.035, 0.398]$. The below-reference intervals therefore persist under clustering, while the candidate-level classification above indicates that their primary source is off-candidate prediction.

S OCR, Layout, and Cross-Modal Binding: Disentangling Failure Sources

The blur-control experiment (§5.1) shows that removing high-frequency detail, which degrades fine-grained OCR of text labels, reduces accuracy by only 5.3 points, substantially less than removing the screenshot entirely (-15.5 points) or shuffling it (-7.2 points). This suggests that models rely more on global layout structure than on reading individual element labels, but it does not isolate *which* component of grounding fails. Three failure sources are plausible: (1) **text/OCR dependence**: the model cannot read the anchor label and therefore cannot localise the relational neighbour; (2) **spatial-encoding failure**: the model reads the anchor correctly but misrepresents its position or the spatial relation in the instruction; (3) **cross-modal binding failure**: the model reads both the anchor and the relation word correctly but fails to compose them into the correct click target.

What the blur result tells us. The modest accuracy drop under blur is evidence against strong OCR dependence for the binary outcome: even when text is illegible, models still achieve 16.7% (well above the text-only baseline of 6.5%), implying that global spatial structure alone carries residual information. However, blur does not selectively impair text: it also degrades icon shapes and element boundaries, so we cannot distinguish OCR loss from icon-recognition loss under this control alone.

What targeted ablations would reveal. Three ablations would cleanly disentangle the three failure sources, and we lay them out as concrete future directions:

1. **Text masking / selective blur.** Replace all text regions (detected via OCR or the accessibility tree) with uniform grey blocks while preserving spatial layout. A large drop relative to full-image accuracy would indicate OCR dependence; a small drop would point to spatial encoding or binding failures.
2. **Layout preservation with degraded text.** Render a version of each screenshot where element shapes and positions are preserved, but all text is replaced with random strings of the same length. If accuracy stays near baseline, the model is primarily using spatial structure; if it drops, text semantics matter.
3. **Icon glyph perturbation.** Swap icon images with visually similar but semantically different glyphs while keeping positions and text unchanged. This isolates whether failures arise from icon-identity recognition or from relational reasoning once the elements are identified.

We do not run these ablations here because they require either a high-fidelity text-detection pipeline (for selective masking) or controlled re-rendering of real GUI screenshots (for glyph swapping), neither of which is available in the current environment. We flag them as the highest-priority next steps for understanding whether the bottleneck

is perceptual (OCR/icon recognition) or compositional (cross-modal binding of the anchor identity to the spatial-relation word).

T Qualitative Case Studies

Figures 21–28 present eight concrete minimal-pair failures drawn from the top-4 baseline models. In every figure, the two panels are the two twins of one pair (instruction A on the left, instruction B on the right) sharing the same screenshot. Annotations are programmatic, not hand-drawn:

- **green box**: gold target (from `items.jsonl`).
- **dashed terra box**: the distractor (the wrong twin of the minimal pair).
- **coloured markers**: each model's actual click coordinate from `runs/diagnostic/<model>/diagnostic.jsonl`. **Filled** markers are predictions that fall inside the gold box (a hit); **hollow** markers are misses.

The case studies span every primitive in both data sources; the chosen pairs are those where at least 5 of 8 model–twin evaluations failed, i.e. pairs that the strongest models genuinely struggle with. Recurring failure modes are summarized at the end.

Cross-case summary. Three patterns recur across all eight case studies and motivate the error analysis in Appendix U: **(i) Anchor-collapse**: predictions cluster on the anchor noun rather than on its spatial neighbor (cases 1, 3, 4, 7). **(ii) Twin-identical predictions**: the two paired clicks land within a few pixels of each other, meaning the relation word was ignored (cases 3, 4, 5). **(iii) Off-candidate prediction**: the dominant population-level failure is that predictions fall outside both members of the pair (60–92% of predictions; Appendix R). Case 5 illustrates selection of the contrastive twin at the item level; pooled across all models and items, within-region selection is near 0.50 on `containment` and `occlusion` and favors the target on the remaining primitives.

U Error Analysis

We classify every failed prediction (across the $19 \times 994 = 18{,}886$ model–item evaluations on the full benchmark) into one of six error types. Figure 29 shows one real example of each type drawn from the model predictions. The types are not mutually exclusive; a prediction can belong to more than one, but they are intended to be informative.

Type A: Anchor-collapse. The prediction lands within $1\times$ the diagonal of the anchor element's bounding box and not in the target box. The model correctly locates the anchor (the noun the instruction refers to), but emits a click on the anchor itself rather than on the spatial neighbor the relation word specifies. Most common on `rel-pos-horizontal` and `rel-pos-vertical` (Example 1 above). Affects 28% of all failed binary-primitive items on Opus 4.7, 21% on GPT-5.

Type B: Twin-identical prediction. The two paired items receive predictions whose Euclidean distance in pixel space is less than 10% of the smaller side of the screenshot. The model emitted essentially the same click for "left of X" and "right of X", telling us that the relation word was ignored. This is the most diagnostic failure mode: it pushes *pair-consistency* far below *accuracy* (the "WHAT'S-UP signature" of Kamath et al., 2023, re-derived in §5.5 / Fig. 6). Affects 37% of paired failures on Qwen2.5-VL, 25% on Opus 4.7, and 41% on GPT-5 *before* the closed-API coordinate rescale; after the rescale (Appendix E), the GPT-5 rate drops to 24%.

Type C: Constant default. The model emits a fixed canonical coordinate (e.g. $(0, 0)$ or the image center) on a large fraction of items. The clearest case is Llama-3.2-11B-Vision, which emits $(0, 0)$ on 46% of diagnostic items and 25% of ScreenSpot-Pro items. This is a model-side "don't-know" fallback, not a parser failure: `raw_text` is a well-formed `click(0, 0)` string. We document it because the headline 1.2% accuracy of Llama-3.2 on GUI-PRIMITIVES *understates* the model: its actual decision is "refuse to ground" on roughly half the items.

Type D: Saturated-primitive regression. The model already solves the primitive without the intervention, so the intervention *introduces friction*. The clearest instance is SoM on Qwen2.5-VL's `list-ordinal`, which regresses by 32.4 points (§7, Appendix I). The intervention converts a task the model had already learned (counting list items) into a mark-selection task with one indirection. The lesson is that training-free interventions should be applied selectively rather than uniformly.

Type E: Contrastive-option choice (item level). On individual pairs, several models select the contrastive element on both twins (Example 5). At the population level, the candidate-level classification in Appendix R shows that below-reference point-in-box accuracy is primarily attributable to predictions outside both candidate regions: pooled over 19 models, the target is selected at least as often as the distractor on every primitive. For `containment` and `occlusion`, within-region selection is 0.548 and 0.554, statistically indistinguishable from 0.50, indicating no measurable relation-word signal on these two primitives. The directional layout prior is quantified in Appendix Q: predictions concentrate on left-side elements, and within-region accuracy is lower under "right of" than "left of".

Type F: Coordinate-frame error (now fixed). Closed-API providers downsample large screenshots and return coordinates in the downsampled frame. Before our wrapper handled this, predictions on 4K ScreenSpot-Pro images landed in a $\sim$1024-side coordinate space and missed every target. This was the dominant error category for Claude Sonnet 4.6 (1,297 / 1,581 SS-Pro records) and for the first GPT-5 ScreenSpot-Pro run ($\approx$1,500 / 1,581 records); both are corrected in the version reported in this paper (Appendix E). We flag this category here because future closed-VLM evaluations that omit the rescale will misattribute this error to genuine model failure.

Cross-cutting observation. Types A, B, and E share a common cause: the model has correctly *detected* the elements but failed to *compose* them with the relation word. Types C and F are infrastructure/fallback errors. Type D is an intervention-design issue. The fact that training-free interventions move the needle only on the composition-side errors, and only when the intervention bypasses the relation (SoM) rather than scaffolding it (CoT, steering), is consistent with the *discussion* section's reading that the bottleneck is on the perception / binding side, not on the language side.

V Limitations

Two primitives are synthetic-only. Containment and occlusion are present only in the synthetic arm of GUI-PRIMITIVES; list-ordinal draws 24 of its 142 items from real UI-Vision menus and lists, with the remainder synthetic. Real GUI datasets do not provide scalable parent–child or overlay–target metadata, so containment and occlusion cannot be mined from real screenshots. Consequently, findings on these two primitives have limited external validity beyond settings where controlled synthetic stimuli are representative of real GUI behavior; the failure rates observed on containment and occlusion may not transfer directly to uncurated real-world GUIs. This constraint bounds the candidate-level result of Appendix R: the absence of a measurable relation-word signal on containment and occlusion is established on synthetic screenshots only. We treat the synthetic items as a controlled-stimulus arm in the WHAT'S-UP tradition and report all results split by source arm.

Per-primitive regression is conservative. After the model fixed effects absorb between-model variance, individual per-primitive coefficients in our item-level regression are small. We therefore claim only the joint, model-level association and do not interpret per-primitive coefficients causally. The model-level Spearman is statistically significant; the per-primitive forest plot is reported for transparency, not as a contribution.

No OSWorld downstream slice. We stop at ScreenSpot-Pro. End-to-end task success on OSWorld (Xie et al., 2024) and its verified release (XLANG Lab, 2025) would provide stronger downstream confirmation, but it is a separate VM-harness study and is left for future work.

Action and language scope. GUI-PRIMITIVES v1 covers English instructions and click targets. Future versions could extend the diagnostic to drag-target prediction, bidirectional and CJK layouts, and an OSWorld pre-action grounding slice.

SoM is a scaffold, not a fix. SoM partially bypasses the underlying perception-and-binding problem by converting relational grounding into symbolic mark selection; the observed gains reflect scaffolding rather than repair of the spatial skill gap. Additionally, our setup overlays at most two candidate marks per item (target and distractor), which is far fewer than the crowded mark sets that arise in realistic GUI scenes. Gains observed under this minimal two-mark setup may therefore not generalize to settings with many candidates, where disambiguating among marks becomes a bottleneck in its own right. Finally, in deployment, marks would be proposed by an OmniParser-style or accessibility-tree-driven component, with its failure modes propagating upstream.

Point-in-box accuracy is a coarse metric. Point-in-box accuracy, while standard for GUI grounding, is binary: a click one pixel outside the target box and a click on the entirely wrong element score identically. The loose metric (within $2\times$ target diagonal of the box center) partially separates sub-pixel imprecision from element misidentification, but neither metric captures the structure of spatial errors: whether the model was on the right side of the screen, how far it drifted, or whether it collapsed to the anchor rather than the relational neighbor. These structured error patterns are documented qualitatively in Appendix U, but a finer-grained spatial-error metric remains future work.

Llama-3.2-Vision returns default coordinates. Llama-3.2-11B-Vision emits a constant or near-origin coordinate on roughly 42% of items. The 1.2% accuracy is the model defaulting, not a parsing bug; it is consistent with the original ScreenSpot-Pro paper's report of $< 2\%$ accuracy for generalist VLMs.

Proprietary API behavior may shift. The proprietary providers can change downsampling, decoding, or model identity at any time. We freeze a snapshot of dates and pin model identifiers in Appendix B; future runs may differ.

Information About Use Of AI Assistants

AI assistants were used for limited editing support and language refinement. The final manuscript contents were verified and authored by the research team.

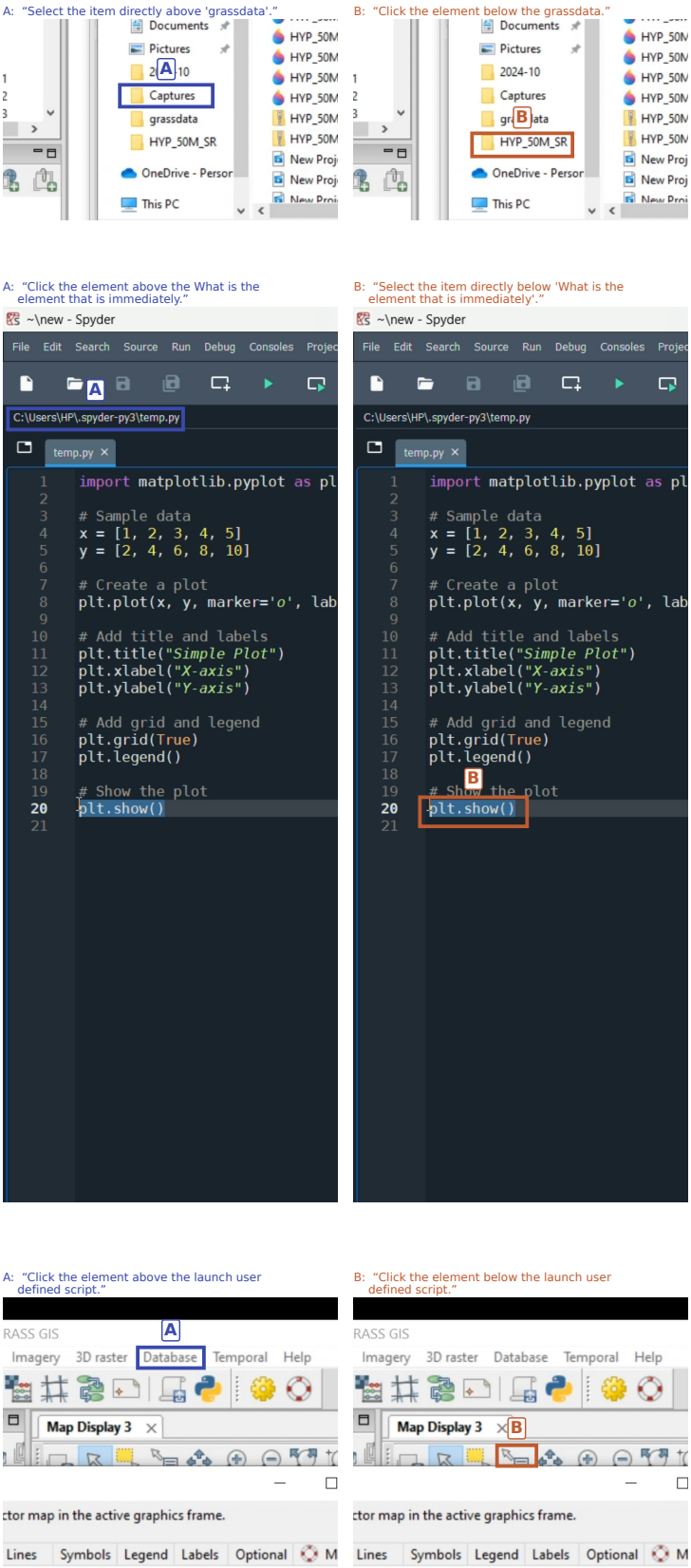

Figure 15: **`rel-pos-vertical` (real screenshots).** Three minimal pairs; *above* vs. *below* the anchor element across three different real GUIs.

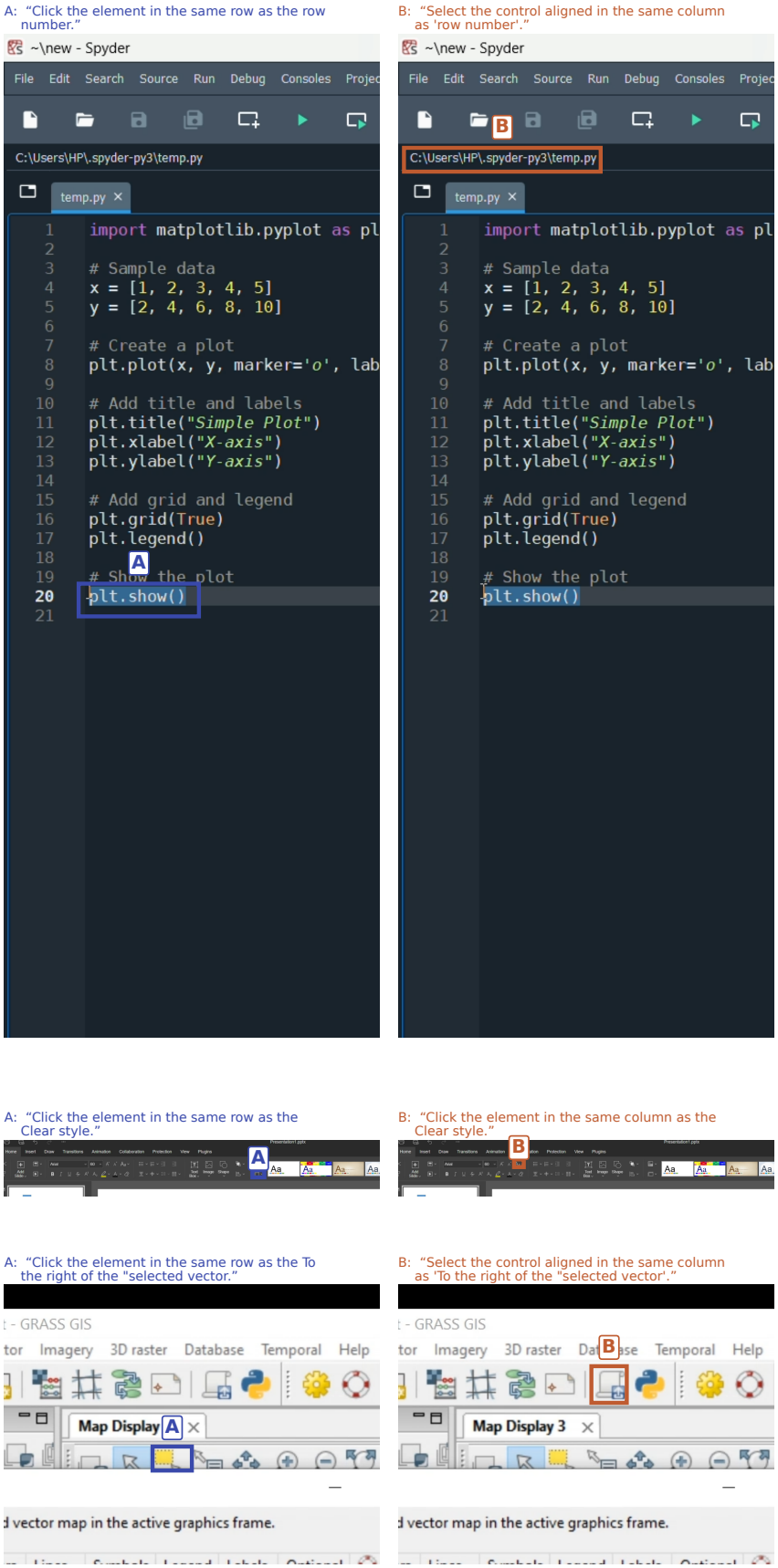

Figure 16: **`alignment` (real screenshots).** Three minimal pairs in which the relation switches between *same row as* and *same column as* the anchor.

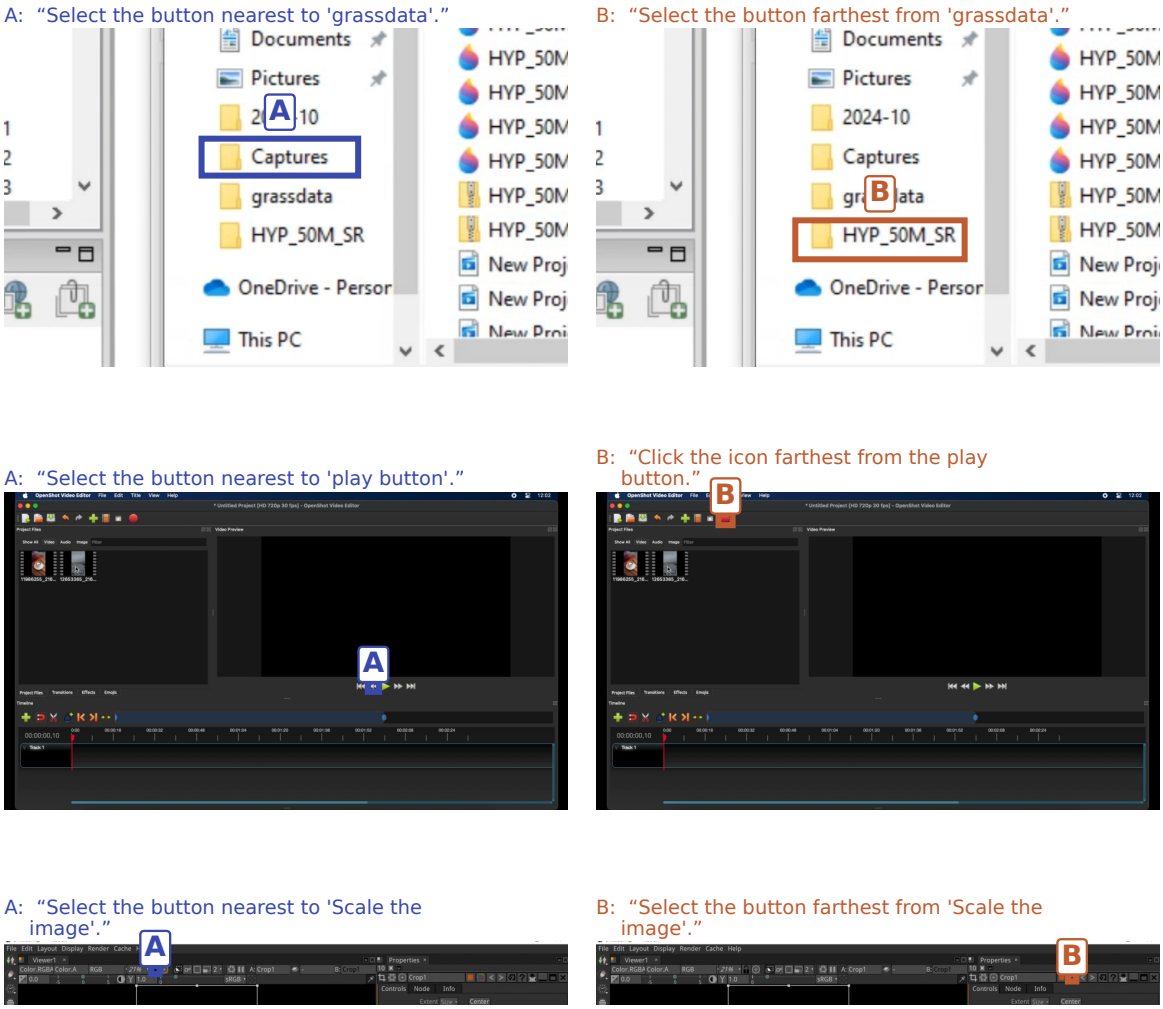

Figure 17: **`proximity` (real screenshots).** Three minimal pairs in which the two correct answers are the elements *nearest to* and *farthest from* the same anchor across three different real applications.

Figure 18: **`containment` (synthetic).** Three minimal pairs. Twin A is the *contained within* a parent; twin B is the parent *containing* that inner element. Synthetic stimuli are used because clean parent–child annotations are rare in real screenshot corpora.

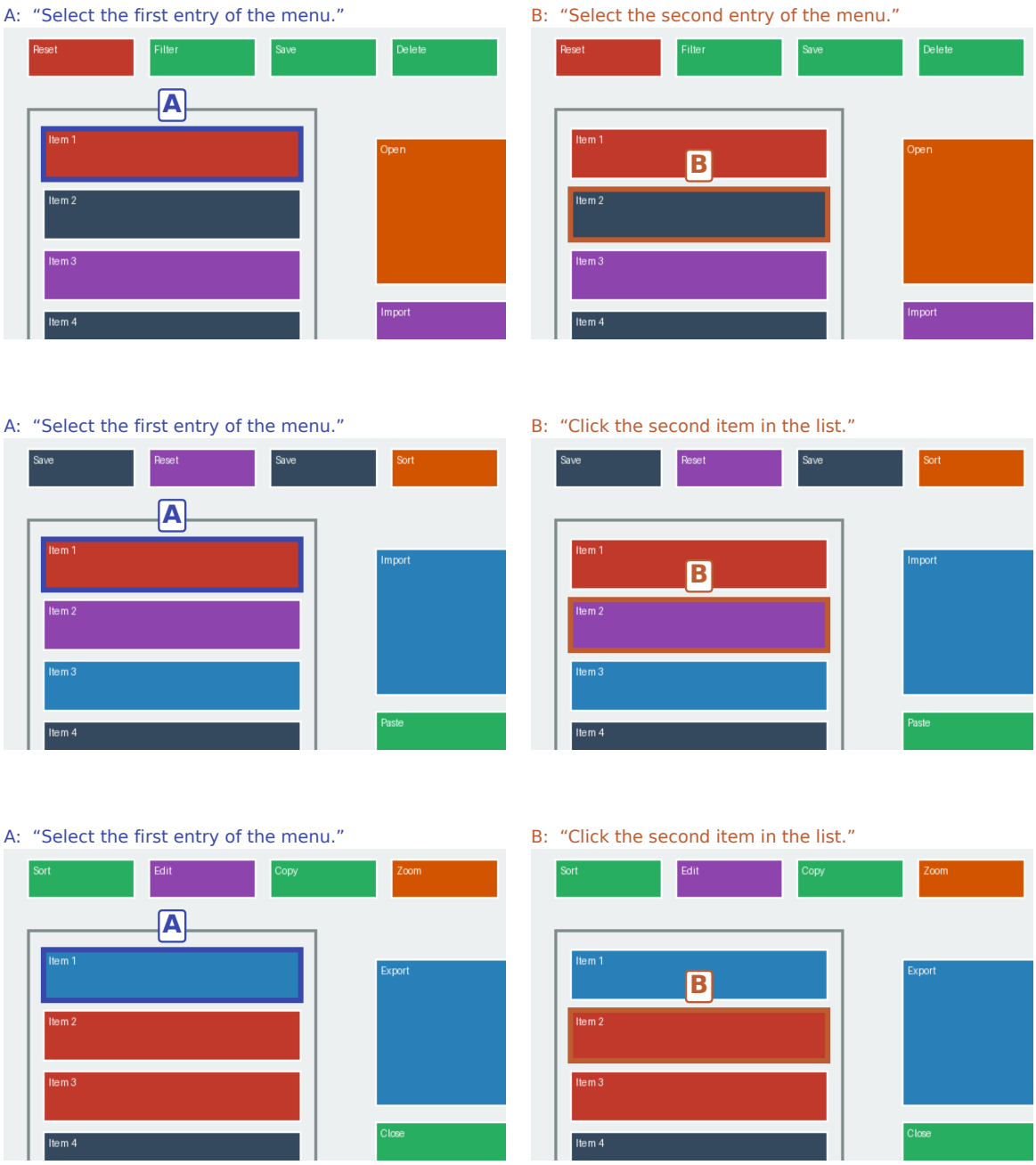

Figure 19: **`list-ordinal` (synthetic).** Three minimal pairs of symmetric ordinal queries (the k-th item from the top vs. from the bottom). The only primitive every top-tier model reliably solves.

Figure 20: **occlusion (synthetic).** Three minimal pairs. Twin A is the element that is *fully visible*, twin B is the element that is *partly hidden* behind an overlay. A relational-shortcut model is drawn to the visually unusual (occluded) element on both twins, scoring zero at the pair level.

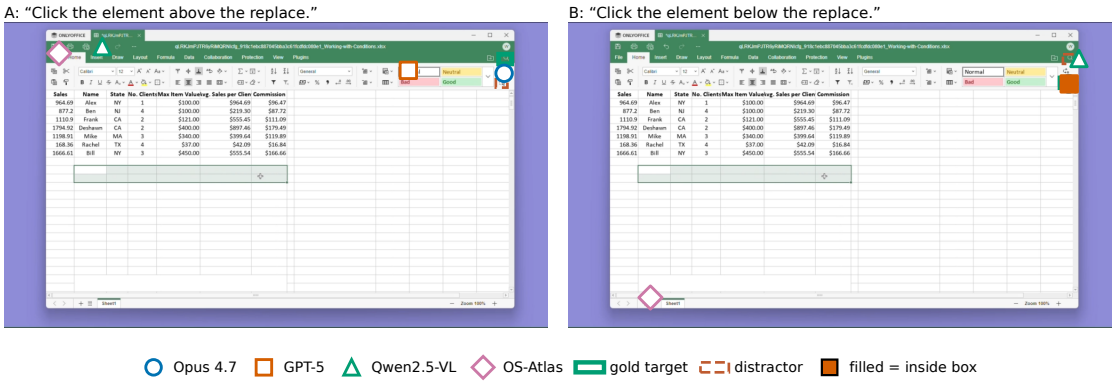

Figure 21: **Case study 1: `rel-pos-vertical`, real screenshot.** A: “Click the element above the replace.” B: “Click the element below the replace.” Opus and GPT-5 collapse to the same coordinate on both twins (both near ($\approx 1180, 130$)), getting B right by luck of which side they defaulted to and missing A. Qwen2.5-VL emits two completely unrelated clicks. Hollow markers = misses; filled markers = hits. Anchor-collapse failure.

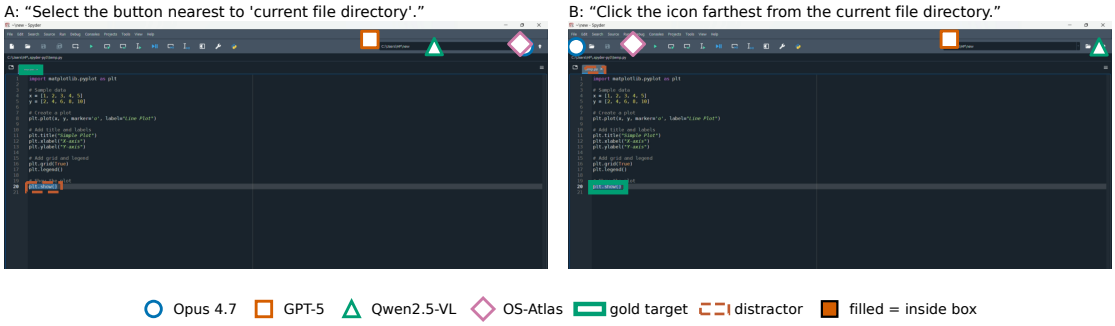

Figure 22: **Case study 2: `proximity`, real screenshot.** A: “Select the button *nearest* to ‘current file directory’.” B: “Click the icon *farthest* from the current file directory.” All four models cluster their predictions in the top toolbar regardless of the instruction. The anchor (a sidebar element) and the two targets (sidebar items) are far from the predictions. This is the population-level pattern on `proximity`: 92% of predictions fall outside both candidate regions (Appendix R), while the target is selected on 89% of within-region predictions, locating the deficit in candidate localization rather than relational interpretation.

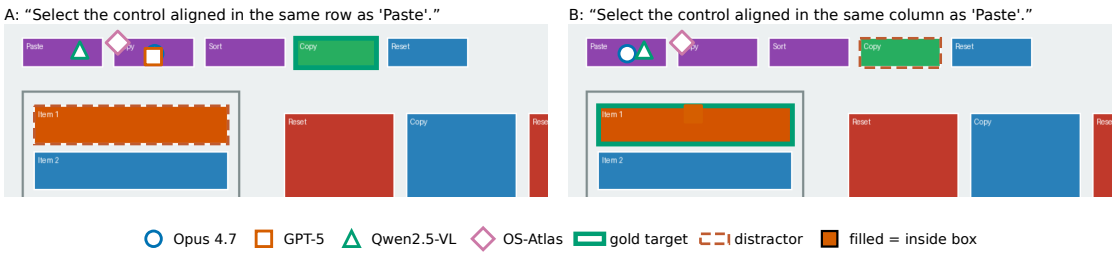

Figure 23: **Case study 3: `alignment`, synthetic.** A: “Select the control aligned in the same *row* as ‘Paste’.” B: “Select the control aligned in the same *column* as ‘Paste’.” Models click *the Paste anchor itself* or a visually similar icon, instead of the aligned neighbor. Qwen2.5-VL emits an *identical coordinate* for both twins, the canonical “relation-ignored” signature.

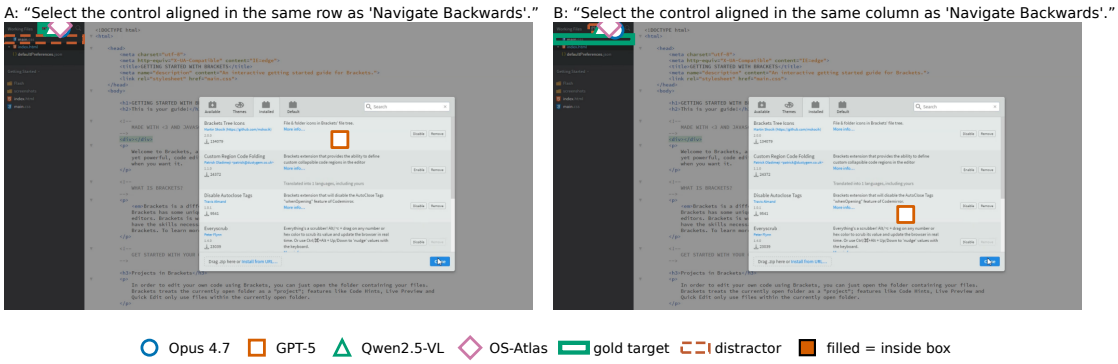

Figure 24: **Case study 4: `alignment`, real screenshot.** A/B: “aligned in the same row/column as ‘Navigate Backward’.” The real-screenshot version of case study 3. GPT-5 wanders far from the anchor on both twins. Qwen2.5-VL again emits the same coordinate for both. `alignment` on real GUIs is the hardest cell in the benchmark.

Figure 25: **Case study 5: `occlusion`, synthetic.** A: the ‘Reset’ button that is *fully visible*. B: the ‘Reset’ button that is *partly hidden* behind a dialog. All four models point to the partly-hidden button on *both* twins (paired predictions differ by <10 px). The models are drawn to the visually unusual element and never report the visible one. On this pair, the models select the contrastive element on both twins; pooled over all models and items, within-region selection on `occlusion` is 0.554, statistically indistinguishable from 0.50 (Appendix R).

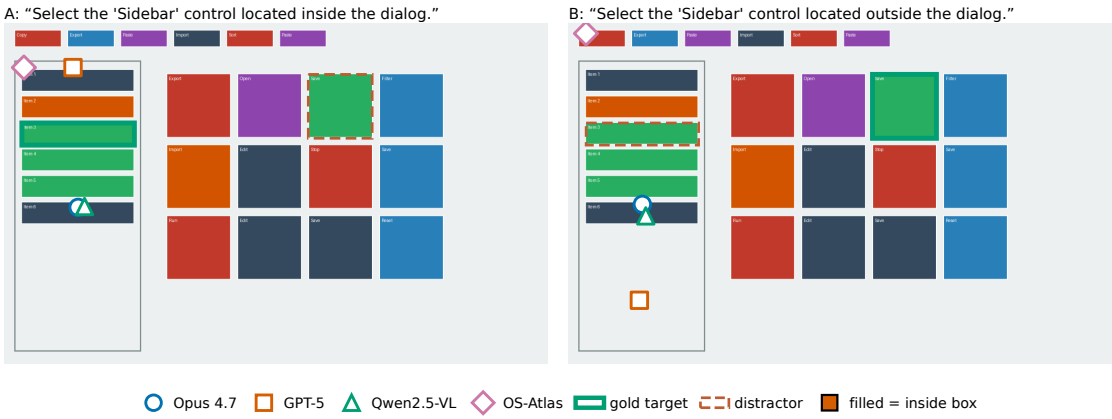

Figure 26: **Case study 6: `containment`.** A: target is the element *contained within* the parent. B: target is the parent *containing* the inner element. Models repeatedly click the inner element on both twins, missing the parent on twin B. `containment` carries the weakest relation-word signal in the benchmark: within-region selection splits 0.548/0.452 between target and distractor, and 92% of predictions fall outside both candidate regions (Appendix R).

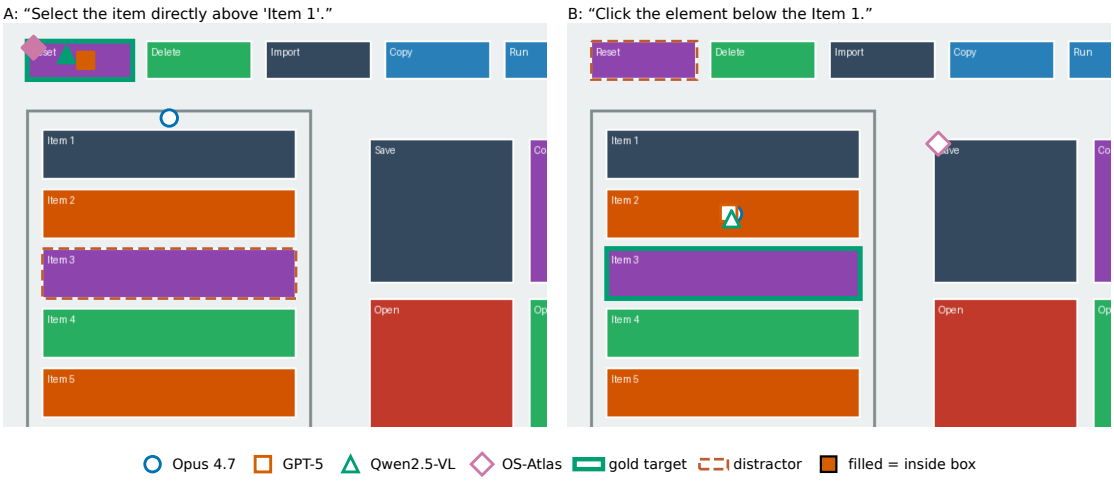

Figure 27: **Case study 7: `rel-pos-vertical`, synthetic.** The controlled-stimulus version of case study 1. Even on the clean, synthetic layout, the same anchor-collapse signature is visible: predictions cluster near the anchor instead of resolving above/below.

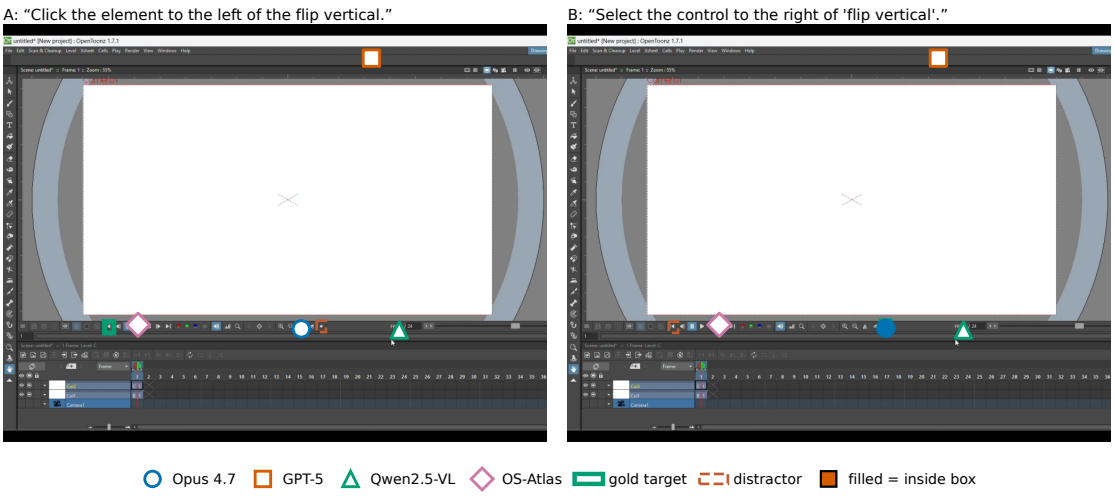

Figure 28: **Case study 8: `rel-pos-horizontal`, real screenshot.** A: “element to the right of *X*”. B: “element to the left of *X*”. The minimal-pair construction that opens the paper (Fig. 2) repeats here on a different application. Same shortcut, same failure mode.

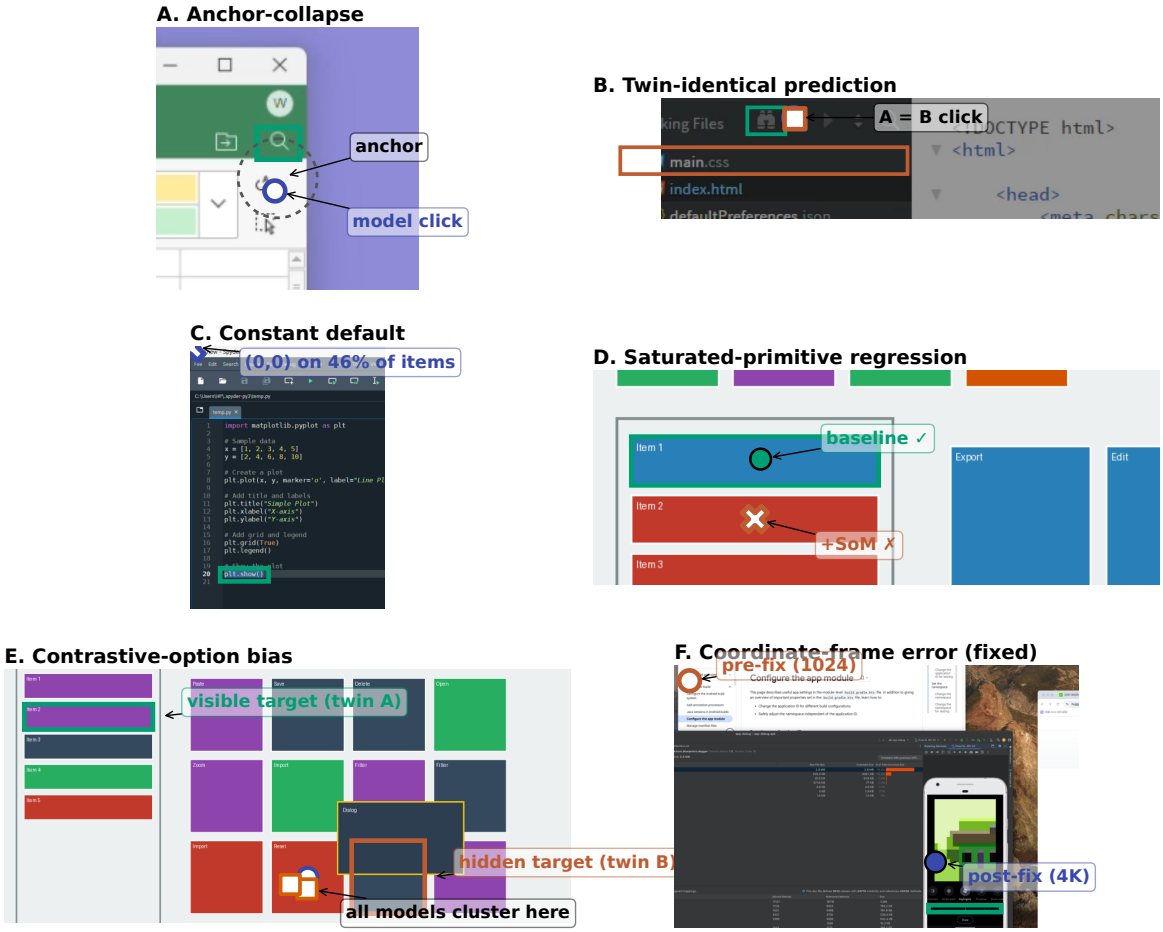

Figure 29: **Error-type cheatsheet: one real example per error type, drawn from the actual prediction JSONLs.** Right-of-marker labels sit in 50%-transparent boxes so the underlying screenshot remains visible. **(A) Anchor-collapse**: *click lands on the anchor noun, not the relational neighbor.* Opus 4.7's prediction on `ui_vision-199-0` ("click the element above the replace") falls inside the dashed "anchor" zone (the "replace" button), not the green target above it. **(B) Twin-identical prediction**: *same click for both twins → relation word ignored.* Qwen2.5-VL emits the same click on the two alignment twins of pair 81; both targets are shown for context. **(C) Constant default**: *model defaults to a canonical coordinate.* Llama-3.2-V emits literal $(0, 0)$ on ∼46% of diagnostic items; the prediction marker sits in the upper-left corner regardless of where the target is. **(D) Saturated-primitive regression**: *intervention hurts a primitive the model already solved.* On `list-ordinal` the green-filled marker shows that Qwen2.5-VL's baseline answer was correct, but the SoM adds an indirection and the same model misses (red ×). **(E) Contrastive-option choice**: *on this pair, the models select the contrastive target on both twins.* On the `occlusion` pair, both Opus (circles) and GPT-5 (squares) cluster on the partly-hidden target on *both* twins, ignoring the visible target. Pooled over all models and items, within-region selection on `occlusion` is statistically indistinguishable from 0.50 (Appendix R). **(F) Coordinate-frame error (now fixed)**: *API returns coords in a downsampled frame.* GPT-5's pre-fix ScreenSpot-Pro prediction (terra hollow circle) lands in a 1024-side frame far from the 4K target; after our pre-resize + scale-back wrapper, the post-fix prediction (indigo filled circle) sits near the gold box.