SCOPE: SPATIAL CONFIDENCE AND OFFLOADING WITH PER-ELEMENT GUARANTEES FOR DEVICE-CLOUD GUI GROUNDING

Anonymous authors
Paper under double-blind review

ABSTRACT

GUI grounding turns a natural language instruction into a coordinate on the screen. Since cloud inference is costly, practical deployments run lightweight edge agents on the device. The edge agent closes the capability gap by offloading the tasks it would get wrong, which calls for uncertainty estimation and a risk-controlled offloading decision. Existing methods typically estimate the uncertainty from logit statistics or repeated sampling, and offload by thresholding the score. However, these methods measure hesitation implicitly without quantifying the deviation of the prediction from the target element, which decides the correctness of grounding. We instead expand the logits of the grounding coordinate digits into a *click distribution* over the screen, and observe that the hesitation separating correct from wrong predictions lies at the high-order digits that decide the element rather than at the low-order digits, and that wrong groundings often escape to the neighborhood of the target. Building on this, we propose SCOPE, a framework for **S**patial **C**onfidence and **O**ffloading with **P**er-**E**lement Guarantees. SCOPE first reads the separation of hesitation across coordinate scales through an entropy gate, cross-checks the prediction by re-grounding on a local view, and calibrates the combined spatial confidence into a conformal click radius in units of UI elements. The edge offloads with a per-element guarantee whenever the radius exceeds one element. Experiments demonstrate that the spatial confidence recovers on average 7.7 percentage points more of the device-cloud capability gap than the strongest baseline, and the calibrated radius offloads 58% of inputs at 90% nominal coverage.

1 INTRODUCTION

Graphical user interface (GUI) automation requires an agent to fulfill user intents through GUI actions. A fundamental capability of such agents is GUI grounding (Cheng et al., 2024; Gou et al., 2025; Wu et al., 2025; Li et al., 2025), which locates the target element given a screenshot and a natural language instruction. Despite the remarkable performance of cloud-scale agents (OpenAI, 2025; Anthropic, 2024; Google DeepMind, 2025), the cost of grounding every step in the cloud pushes practical deployments toward on-device edge agents (Christianos et al., 2025; Yang et al., 2025b; Huang et al., 2026). Yet edge agents still lag behind the cloud in grounding accuracy. They need to offload to the cloud the samples they would get wrong and thus close the capability gap. Since an error kept on the device turns into a wrong click, this calls for estimating the uncertainty of each edge prediction and designing a risk-controlled offloading policy.

However, existing methods answer neither call well. Logit-based methods average confidence over all decoded tokens (Malinin & Gales, 2021; Kadavath et al., 2022), diluting the sparse signal of inter-element ambiguity that surfaces only at coordinates. Sampling-based methods promise sharper signals but multiply the inference cost on the device (Wang et al., 2023; Manakul et al., 2023), and the samples never rise from coordinates to UI elements. Beyond estimation, existing offloading policies typically calibrate a threshold or train a module (Gupta et al., 2024; Ding et al., 2024; Ong et al., 2025). Yet clicking the wrong element often leads to high-latency recovery or is even irreversible (Wang et al., 2026), and none of these policies bounds where a grounding may land relative to the target. Fundamentally, these methods are misaligned with the geometry of GUI grounding, where a grounding is correct only when it falls inside the target UI element. Uncertainty should

Reviewers: please read the Reviewer Guidelines (iclr.cc/Conferences/2027/ReviewerGuidelines) and the AI Policy for Reviewers (iclr.cc/Conferences/2027/AIPolicyForReviewers). If you used AI to expand, edit, or polish your review, please provide the input text to the LLM. Better still, consider skipping the LLM and submitting your original text: we, and the authors, are much more interested in your unedited thoughts than in what an LLM has to say. AI-assisted or not, you are putting your name and reputation behind your review: LLM-generated falsehoods, hallucinations or misrepresentations are subject to disciplinary action, which may include desk-rejecting all papers you have authored.

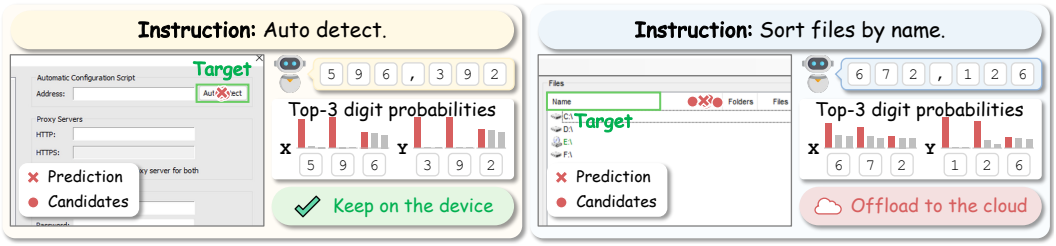


Figure 1: **Click distributions of correct and wrong tasks.** Left: the candidates cluster inside the target when grounding is correct. Right: they escape to a neighboring element when it is wrong.

therefore be measured as a spatial distribution over the screen, and a sample should be offloaded whenever this distribution cannot be narrowed down to a single element.

When a GUI agent hesitates during grounding, the hesitation is largely about where on the screen to click, and this spatial uncertainty is already recorded in the agent's decoding. Specifically, the agent decodes coordinates digit by digit, and each digit token carries a full softmax. Expanding and combining these per-digit alternatives from the decode yields a set of weighted candidate coordinates on the screen, a free byproduct that we call the *click distribution*. As Figure 1 shows, the candidates cluster tightly within the target element when grounding is correct, but escape outside the target element when it is wrong. We observe two phenomena in the click distribution: (1) Uncertainty at low-order digits such as the units moves the coordinate by only a few pixels, and it looks nearly the same whether the grounding is correct or wrong. The uncertainty signal instead concentrates at high-order digits such as the hundreds, which determine the element the prediction grounds on. *The hesitation thus separates by scale*. (2) When grounding fails, the target element often retains no candidate. The candidates shift instead to competing elements that are visually or semantically similar to the target and typically share its local region. *The errors thus escape locally*.

Building on these observations, we treat grounding uncertainty as the geometry of the click distribution. Once placed on the screen, uncertainty inherits the structure of the screen, and each phenomenon turns into a readout. Spatial scale separates pixel-level jitter from element-level ambiguity, telling the edge at which scale to read its own hesitation. Local escape bounds where the missed target can lie, telling the edge where to take a second view. These readouts together define the spatial confidence. The edge then offloads to the cloud whenever the distribution cannot be resolved into a single element. Because this criterion is a measurable geometric event, it admits conformal calibration with a finite-sample coverage guarantee (Vovk et al., 2005).

Technically, we implement this perspective as SCOPE, a framework for inference-time uncertainty estimation and autonomous offloading. From the per-digit coordinate logits, we derive the *scale-profile score*, which measures how strongly the hesitation separates across coordinate scales and gates it by the entropy of the whole decode. From the candidate coordinates, we introduce the *view-consistency score*, which re-grounds on a local view cropped around the candidates with one extra edge decode and takes the spatial distance between the two answers as an external consistency signal. The internal scale-profile score and the external view-consistency score are aggregated by normalization into the final spatial confidence. We further propose the *conformal click radius*: the confidence enters split conformal calibration as a per-sample error scale, yielding a click radius denominated in UI elements, and the edge offloads to the cloud whenever the radius exceeds one element. With marginal extra edge inference, the spatial confidence recovers 78.5% of the device-cloud capability gap on average, versus 70.8% for the strongest baseline. The conformal click radius offloads 58% of inputs on average at a nominal coverage of 90%.

Our contributions are summarized as follows:

- We introduce the click distribution by expanding the coordinate logits from the decode into a distribution over the screen, and reveal two phenomena: hesitation separates by scale, and errors escape locally.
- We propose SCOPE, a framework for uncertainty estimation and autonomous offloading. The scale-profile and view-consistency scores aggregate into the spatial confidence, calibrated into the conformal click radius for offloading with per-element guarantees.

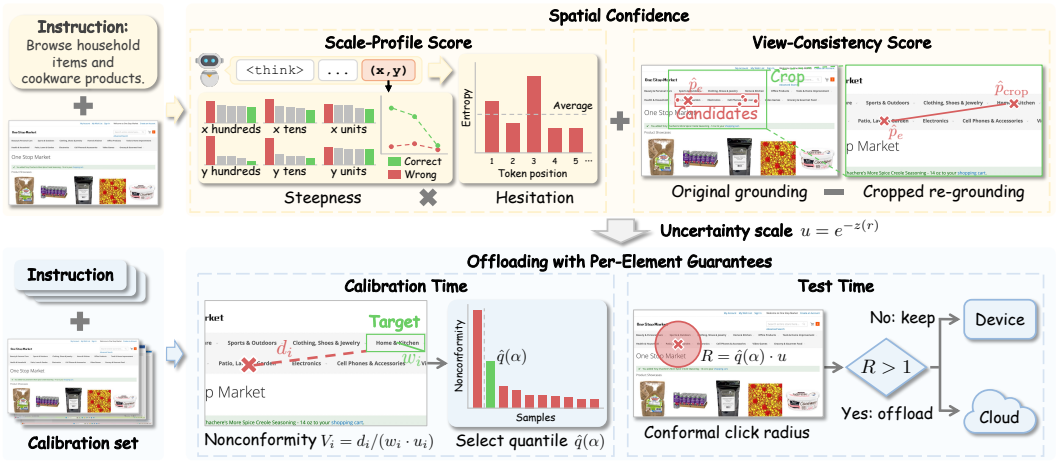


Figure 2: **Overview of SCOPE.** Spatial confidence combines the scale-profile score and the view-consistency score. Offloading with per-element guarantees calibrates it into a conformal click radius that decides between edge and cloud.

- Extensive experiments validate the effectiveness of our method. The spatial confidence recovers 78.5% of the device-cloud capability gap on average, and the conformal click radius offloads 58% of inputs on average at a nominal coverage of 90%.

2 Related Work

GUI Agents and Grounding. Driven by multimodal large language models, GUI agents fulfill user intents through screenshot perception and action execution, with planning, execution, and reflection stages that emulate human interaction (Yao et al., 2023; Hu et al., 2025; Agashe et al., 2025). Among these stages, GUI grounding is the bottleneck capability of action execution, which locates the target element given a screenshot and a natural language instruction (Zheng et al., 2024; Gou et al., 2025; Wu et al., 2025). SeeClick popularized this task with the first realistic grounding benchmark (Cheng et al., 2024). Subsequent work strengthens this capability by pre-training (Gou et al., 2025; Wu et al., 2025; Qin et al., 2025) or fine-tuning (Yang et al., 2025a; Company, 2025; Liu et al., 2026) specialized grounding models on large-scale GUI corpora (Liu et al., 2025; Chen et al., 2025). While cloud-scale agents set the performance frontier (OpenAI, 2025; Anthropic, 2024; Google DeepMind, 2025), the cost of serving each interaction from the cloud has driven the deployment of lightweight 2–4B agents on edge devices (Yang et al., 2025a; Company, 2025; Qwen Team, 2026). Yet due to their limited scale, edge agents still lag behind the cloud in capability. When the edge should act autonomously and when it should offload to the cloud remains an open prerequisite for reliable deployment.

Uncertainty Estimation and Offloading. General uncertainty estimation for LLMs reads length-normalized confidence or entropy from the logits of the decode (Malinin & Gales, 2021; Kadavath et al., 2022), and self-consistency methods pursue sharper signals through multiple samples (Wang et al., 2023; Manakul et al., 2023; Farquhar et al., 2024). Estimation tailored to GUI grounding is just emerging (Kumar et al., 2026). PSS extracts confidence signals from the coordinate logits (Tao et al., 2025). HyperClick calibrates verbalized confidence through reinforcement learning (Zhang et al., 2025). SafeGround builds click density maps from multiple samples and estimates uncertainty at the region level (Wang et al., 2026).

On the offloading side, model cascades trigger requests by thresholding a confidence or quality score (Gupta et al., 2024; Yue et al., 2024). Learned routers instead dispatch queries before generation (Lu et al., 2024; Ong et al., 2025). Both rest on empirical operating points without statistical guarantees. Conformal prediction and distribution-free risk control provide finite-sample guarantees for model predictions (Vovk et al., 2005; Quach et al., 2024; Angelopoulos et al., 2024). KnowNo lets a robot ask for help when more than one candidate action remains (Ren et al., 2023). SafeGround calibrates its threshold to control the error rate of the kept subset. However, the screen-space guaran-

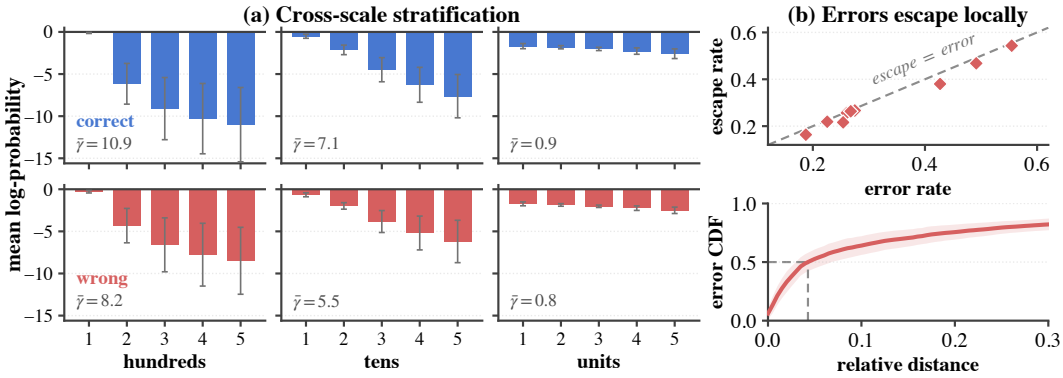


Figure 3: **Mechanisms behind spatial confidence.** (a) Mean and standard deviation of top-5 digit tokens at each place. (b) Top: correlation between error and escape. Bottom: relative distance between the target box and the nearest candidate for wrong predictions.

tee for GUI grounding is never addressed. We calibrate a click radius denominated in UI elements, aligning the guarantee with the unit in which grounding correctness is decided.

3 METHOD

3.1 PROBLEM FORMULATION

Formally, we define the GUI grounding task as follows: given a GUI screenshot $\mathcal{V}$ and a natural language instruction $\mathcal{T}$, the task requires an agent $\mathcal{A}$ to locate the target UI element that $\mathcal{T}$ refers to and to output a coordinate $\hat{p} = (x, y)$ representing the predicted location on the screen. The grounding process can be formulated as:

$$\hat{p} = \mathcal{A}(\mathcal{V}, \mathcal{T}), \tag{1}$$

where the prediction is correct if $\hat{p}$ falls inside the ground-truth bounding box $\mathcal{B} = [x_1, x_2] \times [y_1, y_2]$ of the target element.

In device-cloud collaboration, each input can be served by two heterogeneous agents: a lightweight edge agent $\mathcal{A}_e$ with low cost but limited capability, and a stronger cloud agent $\mathcal{A}_c$ with superior accuracy but higher cost. For each input $(\mathcal{V}, \mathcal{T})$, the device side makes a binary offloading decision $\delta \in \{0, 1\}$, where $\delta = 1$ requests help from the cloud. The final collaborative prediction $\hat{p}^*$ is

$$\hat{p}^* = (1 - \delta) \cdot \mathcal{A}_e(\mathcal{V}, \mathcal{T}) + \delta \cdot \mathcal{A}_c(\mathcal{V}, \mathcal{T}). \tag{2}$$

The goal is a decision rule that offloads as rarely as possible while keeping autonomous edge execution reliable:

$$\min_{\delta} \mathbb{E}[\delta] \;\; \text{s.t.} \;\; \mathbb{P}\big(\delta = 0 \wedge \text{dist}(\hat{p}_e, \mathcal{B}) > w\big) \leq \alpha, \tag{3}$$

where $\hat{p}_e = \mathcal{A}_e(\mathcal{V}, \mathcal{T})$ is the edge prediction, α is the tolerated risk level, and $\text{dist}(p, \mathcal{B}) = \min_{q \in \mathcal{B}} \|p - q\|_2$ is the minimum Euclidean distance from a coordinate to the target element, which is zero inside it. The radius of the target element is defined as $w = \frac{1}{2}\sqrt{(x_2 - x_1)^2 + (y_2 - y_1)^2}$.

3.2 SCOPE DESIGN

The design goal of SCOPE is to let the edge agent decide for each input when to ask the cloud for help. We build on the observation that the decode already carries a distribution of candidate predictions over the screen, whose geometry exposes whether the agent has grounded correctly. As illustrated in Figure 2, SCOPE reads this distribution at two levels to estimate spatial confidence, then calibrates the signal into a conformal click radius that drives the offloading decision. The whole pipeline is training-free and sampling-free, adding only marginal extra edge inference.

3.2.1 SPATIAL CONFIDENCE

Scale-Profile Score. Coordinate digits form a natural multi-resolution partition of the screen. Ambiguity at the high-order digits signals hesitation across different elements, while at the low-order digits it signals jitter within one element. Figure 3(a) confirms this structure. The gap between the top-5 digit tokens narrows gradually from the high-order digits to the low-order digits, and the difference between correct and wrong predictions also concentrates at the high-order digits. We therefore measure the gap of each digit and analyze how steeply it separates across scales. The steepness is then gated by the global hesitation of the decode, which may arise outside the coordinate tokens.

Specifically, let $\mathcal{C}$ denote the set of coordinate digit positions and $p_j^{(k)}$ the k-th largest token probability at position j. Each digit contributes its top-1 to top-5 log-probability gap, whose standard deviation across $\mathcal{C}$ gives the steepness:

$$\gamma_j = \log p_j^{(1)} - \log p_j^{(5)}, \quad g = \sqrt{\frac{1}{|\mathcal{C}|}\sum\nolimits_{j\in\mathcal{C}}(\gamma_j - \bar{\gamma})^2}, \tag{4}$$

where $\bar{\gamma}$ is the mean of γ_j over $\mathcal{C}$. The global hesitation is the average token entropy over all L decoded positions:

$$\bar{H} = -\frac{1}{L}\sum\nolimits_{t=1}^{L}\sum\nolimits_{k=1}^{K} p_t^{(k)} \log p_t^{(k)}, \tag{5}$$

where K is the vocabulary size. The scale-profile score gates the steepness by the global hesitation of the decode:

$$s_{\text{scale}} = g \cdot e^{-\bar{H}}. \tag{6}$$

A correct grounding is confident at the high-order digits and jitters at the low-order ones, so γ_j varies across $\mathcal{C}$ and g is large, and $e^{-\bar{H}} \in (0, 1]$ discounts the steepness by how much the decode hesitates.

View-Consistency Score. We first extract a set of candidate coordinates from the click distribution. At each coordinate digit the decode retains a few of the most probable alternatives. Composing them across digits and weighting each composition by the product of its per-digit probabilities, we keep the m highest-weighted coordinates as the candidate set $\mathcal{P}$, whose construction is detailed in Appendix A.1. We define an *escape* as the event that no candidate falls inside the target element, namely $\mathcal{P} \cap \mathcal{B} = \emptyset$. When the agent grounds correctly, the candidates concentrate on the target. When the prediction fails, they escape as a whole. As shown in Figure 3(b), wrong predictions rarely retain any candidate on the target, and the escape correlates strongly with grounding failure. Yet grounding errors tend to arise among elements with similar appearance or semantics, so the true target usually stays within the candidate neighborhood. A finer look at this neighborhood thus offers a second opinion on the prediction.

The edge agent therefore re-grounds within the neighborhood. We crop a local view centered on the bounding box of $\mathcal{P}$ with view fraction β, whose width and height are at least β times those of the original view, as detailed in Appendix A.2. The agent then re-grounds on the cropped view and maps the answer back to the screenshot as $\hat{p}_{\text{crop}}$. The view-consistency score is the negative displacement between the two answers:

$$s_{\text{view}} = -\frac{\|\hat{p}_e - \hat{p}_{\text{crop}}\|_2}{D}, \tag{7}$$

where $D = \sqrt{W^2 + H^2}$ is the screen diagonal and W, H are the screen width and height. The two decodes answer the same instruction under different visual conditioning. Agreement corroborates the prediction, while discrepancy indicates that at least one view is unreliable. Since the crop reduces the visual tokens that dominate grounding inference, the re-grounding adds only marginal computational overhead.

The two scores read the click distribution at complementary levels. To obtain a unified score, we first normalize them individually:

$$z(s_*) = \frac{s_* - \text{median}(s_*)}{\text{MAD}(s_*)}, \tag{8}$$

where s_* represents the respective set of scores on the calibration set, and $\text{MAD}(s_*) = \text{median}(|s_* - \text{median}(s_*)|)$ is the median absolute deviation. The normalized scores are then aggregated as the spatial confidence:

$$r = z(s_{\text{scale}}) + z(s_{\text{view}}). \tag{9}$$

Table 1: **Uncertainty estimation for edge agents.** We report AUCG for three edge agents on three benchmarks. The best and the second best results are highlighted in bold and underlined.

Methods	ScreenSpot-Pro			UI-I2E-Bench			MMBench-GUI			Mean
	InfiGUI	Holo	Qwen	InfiGUI	Holo	Qwen	InfiGUI	Holo	Qwen	
Random	0.484	0.486	0.407	0.504	0.455	0.530	0.505	0.499	0.483	0.484
LN-Confidence	0.512	0.669	0.737	0.524	0.601	0.659	0.568	0.602	0.795	0.630
LN-Entropy	0.532	<u>0.724</u>	0.764	0.547	0.740	0.726	0.592	0.668	0.806	0.678
Self-Consist.	0.558	0.694	<u>0.808</u>	0.650	0.749	0.710	0.587	0.644	0.807	0.690
PSS	0.518	0.649	0.718	0.405	0.668	0.615	0.507	0.620	<u>0.809</u>	0.612
SafeGround	0.588	0.695	**0.810**	<u>0.669</u>	0.747	**0.789**	0.618	0.663	0.797	0.708
Ours (s_{scale})	<u>0.602</u>	0.689	0.697	0.647	<u>0.881</u>	0.733	<u>0.704</u>	<u>0.730</u>	0.794	<u>0.720</u>
Ours (+ s_{view})	**0.665**	**0.773**	0.774	**0.742**	**0.980**	<u>0.753</u>	**0.743**	**0.795**	**0.835**	**0.785**

3.2.2 Offloading with Per-Element Guarantees

Calibration Time. The correctness of GUI grounding is judged at the granularity of elements. When assessing how far a prediction deviates, the target element itself therefore provides the natural unit of measure. Conformal prediction turns point predictions into prediction sets that cover the ground truth with probability at least $1-\alpha$. For coordinate outputs, the prediction set is a region around the predicted point. On the screen, this region becomes an actionable decision: execute when it is smaller than the target element, and offload when it is larger. The error mode of grounding makes the regions of correct and wrong predictions separate cleanly: deviations of correct predictions are exactly zero, while those of wrong ones are typically large.

We instantiate the region with split conformal prediction. All statistics are fitted on a labeled calibration set of size n. Test time requires no labels. The spatial confidence r is first mapped to a per-sample error scale $u_i = e^{-z(r_i)}$. A confident sample declares a small u_i and a hesitant one a large u_i. For each sample i, the grounding deviation and the element radius are normalized by the screen diagonal:

$$d_i = \frac{\text{dist}(\hat{p}_{e,i}, \mathcal{B}_i)}{D_i}, \qquad w_i = \frac{\sqrt{b_{w,i}^2 + b_{h,i}^2}}{2D_i}, \tag{10}$$

where $b_{w,i}$ and $b_{h,i}$ are the width and height of the target element's bounding box $\mathcal{B}_i$, and D_i is the screen diagonal of sample i. The nonconformity score measures how many element radii the prediction deviates, relative to the scale the agent itself declares:

$$V_i = \frac{d_i}{w_i \cdot u_i}. \tag{11}$$

Correct predictions have $V_i = 0$ regardless of confidence. A wrong prediction that claimed confidence yields a huge V_i, whereas one that declared hesitation in advance is discounted by its u_i. On the calibration set we take the finite-sample quantile

$$\hat{q}(\alpha) = \text{Quantile}_{\lceil (n+1)(1-\alpha) \rceil / n}(\{V_i\}). \tag{12}$$

Test Time. The edge agent obtains u_i based on spatial confidence. Each prediction receives its conformal click radius, denominated in element radii:

$$R_i = \hat{q}(\alpha) \cdot u_i. \tag{13}$$

When calibration and test data are exchangeable, the radius carries a marginal coverage guarantee

$$\mathbb{P}(d_i \le R_i \cdot w_i) \ge 1 - \alpha, \tag{14}$$

that is, for at least a $1-\alpha$ fraction of predictions, the target lies within R_i element radii of the prediction. The offloading rule is that the edge agent keeps a prediction only when its radius fits within one element and asks the cloud for help otherwise:

$$\delta_i = \mathbb{I}(R_i > 1), \tag{15}$$

where $\mathbb{I}(\cdot)$ denotes the indicator function. Together, the offloading rule and the coverage guarantee satisfy the reliability constraint in Eq. (3).

Table 2: **Comprehensive uncertainty estimation performance.** Extra denotes the additional inference cost relative to a single grounding decode. The best and the second best results are highlighted in bold and underlined.

Methods	AUROC	AURC↓	bgt@.9↓	acc@.3	Extra
Random	0.496	0.327	0.903	0.704	0
LN-Confidence	0.650	0.241	0.754	0.724	0
LN-Entropy	0.700	0.212	0.684	0.730	0
Self-Consist.	0.675	0.244	0.723	<u>0.737</u>	5×
PSS	0.660	0.234	0.737	0.722	0
SafeGround	0.706	0.214	0.624	0.737	10×
Ours (s_{scale})	<u>0.763</u>	<u>0.174</u>	<u>0.614</u>	0.737	0
Ours (+ s_{view})	**0.775**	**0.160**	**0.495**	**0.747**	0.20×

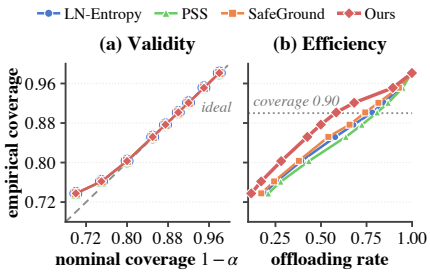


Figure 4: **Coverage validity and offloading efficiency across risk levels.** (a) Empirical versus nominal coverage. (b) Coverage–offloading frontier.

4 Experiments

4.1 Experimental Settings

Benchmarks. We evaluate on three public GUI grounding benchmarks: (1) **ScreenSpot-Pro** (Li et al., 2025) demands precise grounding of tiny targets among dense distractors on professional high-resolution screens. (2) **UI-I2E-Bench** (Liu et al., 2025) focuses on implicit semantic understanding, where instructions mostly describe the target implicitly by function or position rather than visible text. (3) **MMBench-GUI** (Wang et al., 2025) spans element grounding tasks on six desktop, mobile, and web platforms, validating generalization across diverse GUI styles.

Evaluation Metrics. (1) *Estimation Metrics.* Our primary metric is the Area Under the Collaborative Gain (**AUCG**). Given an offloading budget ρ, the least confident ρ fraction of samples are offloaded to the cloud while the rest keep their edge predictions, yielding the collaborative accuracy $\mathrm{Acc}_{\mathrm{col}}(\rho)$. AUCG integrates the recovered fraction of the capability gap over the budget range:

$$\mathrm{AUCG} = \int_0^1 \frac{\mathrm{Acc}_{\mathrm{col}}(\rho) - \mathrm{Acc}_{\mathrm{e}}}{\mathrm{Acc}_{\mathrm{c}} - \mathrm{Acc}_{\mathrm{e}}}\, d\rho, \tag{16}$$

where $\mathrm{Acc}_{\mathrm{e}}$ and $\mathrm{Acc}_{\mathrm{c}}$ denote the edge and cloud accuracy. We further report the minimum offloading budget that recovers a fraction η of the capability gap as **bgt@**η, and the collaborative accuracy at a fixed budget ρ as **acc@**ρ, as well as the Area Under the Receiver Operating Characteristic (**AUROC**) (Bradley, 1997) and the Area Under the Risk-Coverage curve (**AURC**) (Geifman et al., 2019). (2) *Offloading Metrics.* We report the offloading rate at the risk level α as **off@**α, and the grounding accuracy on the samples kept on the device as **acc@kept**.

Compared Baselines. We compare our framework with representative baselines categorized into three groups: (1) **Random**, which assigns random scores to serve as a performance lower bound. (2) Generic uncertainty estimation, including **LN-Confidence** and **LN-Entropy** (Malinin & Gales, 2021). (3) GUI-specific estimation, including **Self-Consistency** (Wang et al., 2023), Peak Sharpness Score (**PSS**) (Tao et al., 2025), and **SafeGround** (Wang et al., 2026).

Table 3: **Conformal offloading within and across benchmarks.** In-bench results are averaged over random splits, while cross-bench calibration uses the other two benchmarks. † denotes use of our conformal click-radius rule.

Methods	In-bench		Cross-bench	
	off@.1↓	acc@kept	off@.1↓	acc@kept
SafeGround	0.942	0.911	0.871	0.890
LN-Entropy†	0.782	0.868	0.833	0.869
PSS†	0.808	0.858	0.839	0.815
SafeGround†	<u>0.743</u>	0.856	0.840	0.838
Ours (s_{scale})	0.743	0.925	<u>0.832</u>	0.912
Ours (+ s_{view})	**0.584**	0.891	**0.609**	0.860

Implementation Details. We employ InfiGUI-G1-3B (Liu et al., 2026), Holo1.5-3B (Company, 2025), and Qwen3.5-4B (Qwen Team, 2026) as the edge agents $\mathcal{A}_e$, and Qwen3.5-35B-A3B as the cloud agent $\mathcal{A}_c$. The three edge agents and the three benchmarks combine into nine configurations. All edge predictions come from greedy decoding, and the

Table 4: **Practical deployment costs.** Off. and Up. denote the offloading rate and the expected uplink payload (KiB). TFLOPs are reported as edge/cloud. Latency is the expected end-to-end latency.

System	Acc	Off.	Up.	TFLOPs	Latency
Edge-only	0.673	–	–	38.9 / –	2.17 s
Cloud-only	0.782	1.000	1082	– / 27.9	3.99 s
Ours	0.773	0.584	801	46.5 / 18.3	5.98 s

Table 5: **Ablation of the spatial confidence.** Results are averaged over nine configurations.

Variant	AUCG	off@.1↓	acc@kept
Steepness	0.707	0.746	0.917
Global Hesitation	0.678	0.784	0.869
Scale-Profile	0.720	0.743	0.925
View-Consistency	0.753	0.576	0.848
Ours	**0.785**	**0.584**	0.891

view-consistency decode is greedy as well. For the view-consistency score, we set $m = 16$ and $\beta = 0.5$. Each configuration is randomly split into equal calibration and test halves, on which the normalization statistics and the conformal quantile are fitted and evaluated respectively. We report the average over 100 random splits at the risk level $\alpha = 0.1$. All experiments run on NVIDIA A100 GPUs, except that the edge agent of Table 4 is measured on an RTX 4090 D. More details are provided in Appendix C.

4.2 Uncertainty Estimation Performance

Table 1 reports the collaborative gains. The scale-profile score alone averages 0.720 and already surpasses all baselines, showing that the cross scale separation of a single decode carries a strong uncertainty signal by itself. Adding the view-consistency score further raises the mean to 0.785, which is 7.7 points above the strongest baseline. While SafeGround achieves competitive AUCG on Qwen3.5, sampling based estimation largely depends on the agent retaining a well calibrated sampling distribution (Kirk et al., 2024), and such baselines degrade sharply on the fine-tuned InfiGUI-G1 and Holo1.5. This suggests that task specific fine-tuning sharpens the output distribution so that multiple sampling can hardly produce effective disagreement. Our method instead stays strong across all configurations, since it reads the distribution embedded in the decode itself rather than reconstructing it from sampling, which grows unreliable as edge agents specialize.

Table 2 extends the comparison beyond AUCG, where our method leads on every metric for both edge decisions and collaboration. Notably, the scale-profile score reuses the logits of the decode and already delivers leading performance at zero extra cost, and the full method adds only one lightweight decode on the reduced local view for a further considerable gain.

4.3 Collaborative Performance

We further examine whether the conformal click radius converts the spatial confidence into autonomous offloading with per-element guarantees, as reported in Table 3 and Figure 4. Figure 4(a) shows that, regardless of how the uncertainty is measured, the empirical coverage tracks the nominal level from above across the whole risk range. Under the guarantee at a risk level of 0.1, our method offloads only 58.4% of inputs while maintaining high accuracy on the kept predictions, whereas native SafeGround reaches a similar kept accuracy but offloads as much as 94.2%, securing reliability at the cost of nearly abandoning autonomy. Applying our rule to the baseline scores lowers their offloading rates, yet a clear gap to our method remains. The rule guarantees coverage for whatever score it wraps, while the autonomy it preserves depends on the estimation quality, so the collaborative efficiency comes from combining them.

The advantage remains robust across benchmarks and risk levels. When calibrated across benchmarks, our offloading rate rises by only 2.5 points, so an agent facing a new scenario can reuse its calibration and still retain most of its autonomy. Figure 4(b) extends the comparison across risk levels. Our frontier lies leftmost, keeping the most decisions on the device under matched coverage.

4.4 Practical Deployment Costs

Table 4 presents the practical deployment costs, where the edge agents run on an RTX 4090 D and the cloud agent on an A100, connected through a link with a bandwidth of 20 Mbps and a round trip delay of 40 ms. While offloading 58.4% of inputs, our method achieves an accuracy close to that of cloud-only, indicating that the offloaded inputs largely concentrate on the cases where the cloud

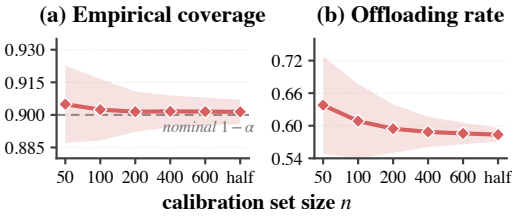


Figure 5: **Robustness to calibration set size.** Coverage stays near the nominal 0.90 and offloading stabilizes as n grows. Mean and one standard deviation across random splits.

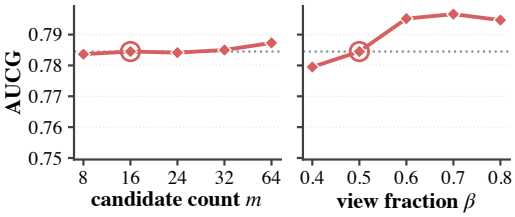


Figure 6: **Sensitivity to view-construction hyperparameters.** We vary m at $\beta = 0.5$ (left) and β at $m = 16$ (right). Open circles and the dotted line mark the selected setting.

brings an advantage, which reduces the expected uplink payload and the cloud compute. The extra inference modestly increases edge compute and lengthens the end-to-end latency, since the edge runs its inference sequentially before the cloud is invoked. Deployments sensitive to latency can fall back to the scale-profile score alone, which removes the extra view at the price of offloading more.

4.5 Robustness and Sensitivity Analysis

Figure 5 shows how much calibration data our method needs. The empirical coverage stays slightly above the nominal level at each calibration size, and as few as 50 samples already meet it. Small calibration sets instead cost efficiency. With few samples, the rule selects a more conservative order statistic, which inflates the radius and offloads more inputs for safety. This inflation fades as the calibration set grows, and with about 200 samples the offloading rate already approaches its level on the full calibration half. Our method therefore secures validity with little calibration data, while additional data mainly brings more autonomy to the edge.

Figure 6 varies the two hyperparameters of view construction. The candidate count m has almost no effect, since the view size is set by the view fraction β, and additional candidates cluster near the existing ones and barely move the view center. As β increases, AUCG first rises and then levels off, suggesting that the second prediction benefits from a compact context, while a view approaching the full screenshot makes its answer redundant with the first prediction. We keep $\beta = 0.5$, which trades a little collaborative gain for a smaller view and thus lower inference cost on the device.

4.6 Ablation Study

We ablate the components of the spatial confidence, as shown in Table 5. Within the scale-profile score, steepness alone already performs on par with the strongest baseline, which verifies the effectiveness of the cross scale separation. Global hesitation alone has the same AUCG as LN-Entropy, since they rank samples identically. Gating steepness with it further improves the scale-profile score, as they carry relatively independent information and both contribute positively to uncertainty estimation. The view-consistency score is the stronger single signal and offloads the fewest samples, but the predictions it keeps are also the least accurate, because two decodes of the same agent may agree on the same wrong element. Combining the internal cross scale separation with the external inconsistency check under a changed view achieves the most competitive results.

5 Conclusion

We propose SCOPE, a framework for reliable device-cloud GUI grounding through uncertainty estimation and autonomous offloading. Our central insight is that correctness in GUI grounding is determined at UI elements. We therefore expand the decode of the edge agent into a click distribution and uncover two patterns of grounding uncertainty, where hesitation separates by scale and errors escape locally. SCOPE turns the patterns into the scale-profile and view-consistency scores, and calibrates the combined spatial confidence into a conformal click radius in UI element units, offloading whenever the radius exceeds one element. Extensive experiments demonstrate that SCOPE recovers 78.5% of the device-cloud capability gap on average and offloads 58% of inputs at 90% nominal coverage. We believe that aligning uncertainty estimation and calibration with the element-level geometry of GUI grounding offers a promising direction for reliable device-cloud collaboration.

AI USE STATEMENT

In this work, we used generative AI tools to assist in implementing the code for our experiments. We have not used generative AI tools to develop the conceptual framework, to propose or refine hypotheses, to design the methodology or experiments, or to interpret results, and the rest of the required disclosure tasks are not applicable to this work. Additionally, we used generative AI tools to identify relevant literature and to edit the paper for readability. We have reviewed all AI-assisted work. The code was verified and tested for correctness by the authors. The retrieved literature and the edited text were checked by the authors. We take responsibility for the final content of this work, including text, claims or artifacts produced with the aid of generative AI.

REFERENCES

Saaket Agashe, Jiuzhou Han, Shuyu Gan, Jiachen Yang, Ang Li, and Xin Eric Wang. Agent S: an open agentic framework that uses computers like a human. In *The Thirteenth International Conference on Learning Representations, ICLR 2025, Singapore, April 24-28, 2025*. OpenReview.net, 2025. URL https://openreview.net/forum?id=lIVRgt4nLv.

Anastasios Nikolas Angelopoulos, Stephen Bates, Adam Fisch, Lihua Lei, and Tal Schuster. Conformal risk control. In *The Twelfth International Conference on Learning Representations, ICLR 2024, Vienna, Austria, May 7-11, 2024*. OpenReview.net, 2024. URL https://openreview.net/forum?id=33XGfHLtZg.

Anthropic. Introducing computer use, a new Claude 3.5 Sonnet, and Claude 3.5 Haiku, October 2024. URL https://www.anthropic.com/news/3-5-models-and-computer-use.

Andrew P. Bradley. The use of the area under the ROC curve in the evaluation of machine learning algorithms. *Pattern Recognit.*, 30(7):1145–1159, 1997. doi: 10.1016/S0031-3203(96)00142-2. URL https://doi.org/10.1016/S0031-3203(96)00142-2.

Wentong Chen, Junbo Cui, Jinyi Hu, Yujia Qin, Junjie Fang, Yue Zhao, Chongyi Wang, Jun Liu, Guirong Chen, Yupeng Huo, Yuan Yao, Yankai Lin, Zhiyuan Liu, and Maosong Sun. Guicourse: From general vision language model to versatile GUI agent. In Wanxiang Che, Joyce Nabende, Ekaterina Shutova, and Mohammad Taher Pilehvar (eds.), *Proceedings of the 63rd Annual Meeting of the Association for Computational Linguistics (Volume 1: Long Papers), ACL 2025, Vienna, Austria, July 27 - August 1, 2025*, pp. 21936–21959. Association for Computational Linguistics, 2025. doi: 10.18653/V1/2025.ACL-LONG.1065. URL https://doi.org/10.18653/v1/2025.acl-long.1065.

Kanzhi Cheng, Qiushi Sun, Yougang Chu, Fangzhi Xu, Yantao Li, Jianbing Zhang, and Zhiyong Wu. Seeclick: Harnessing GUI grounding for advanced visual GUI agents. In Lun-Wei Ku, Andre Martins, and Vivek Srikumar (eds.), *Proceedings of the 62nd Annual Meeting of the Association for Computational Linguistics (Volume 1: Long Papers), ACL 2024, Bangkok, Thailand, August 11-16, 2024*, pp. 9313–9332. Association for Computational Linguistics, 2024. doi: 10.18653/V1/2024.ACL-LONG.505. URL https://doi.org/10.18653/v1/2024.acl-long.505.

Filippos Christianos, Georgios Papoudakis, Thomas Coste, Jianye Hao, Jun Wang, and Kun Shao. Lightweight neural app control. In *The Thirteenth International Conference on Learning Representations, ICLR 2025, Singapore, April 24-28, 2025*. OpenReview.net, 2025. URL https://openreview.net/forum?id=BL4WBIfyrz.

H Company. Holo1.5 - open foundation models for computer use agents, 2025. URL https://huggingface.co/collections/Hcompany/holo15-68c1a5736e8583a309d23d9b.

Dujian Ding, Ankur Mallick, Chi Wang, Robert Sim, Subhabrata Mukherjee, Victor Rühle, Laks V. S. Lakshmanan, and Ahmed Hassan Awadallah. Hybrid LLM: cost-efficient and quality-aware query routing. In *The Twelfth International Conference on Learning Representations, ICLR 2024, Vienna, Austria, May 7-11, 2024*. OpenReview.net, 2024. URL https://openreview.net/forum?id=02f3mUtqnM.

Sebastian Farquhar, Jannik Kossen, Lorenz Kuhn, and Yarin Gal. Detecting hallucinations in large language models using semantic entropy. *Nat.*, 630(8017):625–630, 2024. doi: 10.1038/S41586-024-07421-0. URL https://doi.org/10.1038/s41586-024-07421-0.

Yonatan Geifman, Guy Uziel, and Ran El-Yaniv. Bias-reduced uncertainty estimation for deep neural classifiers. In *7th International Conference on Learning Representations, ICLR 2019, New Orleans, LA, USA, May 6-9, 2019*. OpenReview.net, 2019. URL https://openreview.net/forum?id=SJfb5jCqKm.

Google DeepMind. Introducing the Gemini 2.5 computer use model, October 2025. URL https://blog.google/technology/google-deepmind/gemini-computer-use-model/.

Boyu Gou, Ruohan Wang, Boyuan Zheng, Yanan Xie, Cheng Chang, Yiheng Shu, Huan Sun, and Yu Su. Navigating the digital world as humans do: Universal visual grounding for GUI agents. In *The Thirteenth International Conference on Learning Representations, ICLR 2025, Singapore, April 24-28, 2025*. OpenReview.net, 2025. URL https://openreview.net/forum?id=kxnoqaisCT.

Neha Gupta, Harikrishna Narasimhan, Wittawat Jitkrittum, Ankit Singh Rawat, Aditya Krishna Menon, and Sanjiv Kumar. Language model cascades: Token-level uncertainty and beyond. In *The Twelfth International Conference on Learning Representations, ICLR 2024, Vienna, Austria, May 7-11, 2024*. OpenReview.net, 2024. URL https://openreview.net/forum?id=KgaBScZ4VI.

Xueyu Hu, Tao Xiong, Biao Yi, Zishu Wei, Ruixuan Xiao, Yurun Chen, Jiasheng Ye, Meiling Tao, Xiangxin Zhou, Ziyu Zhao, Yuhuai Li, Shengze Xu, Shenzhi Wang, Xinchen Xu, Shuofei Qiao, Zhaokai Wang, Kun Kuang, Tieyong Zeng, Liang Wang, Jiwei Li, Yuchen Eleanor Jiang, Wangchunshu Zhou, Guoyin Wang, Keting Yin, Zhou Zhao, Hongxia Yang, Fan Wu, Shengyu Zhang, and Fei Wu. OS agents: A survey on mllm-based agents for computer, phone and browser use. In Wanxiang Che, Joyce Nabende, Ekaterina Shutova, and Mohammad Taher Pilehvar (eds.), *Proceedings of the 63rd Annual Meeting of the Association for Computational Linguistics (Volume 1: Long Papers), ACL 2025, Vienna, Austria, July 27 - August 1, 2025*, pp. 7436–7465. Association for Computational Linguistics, 2025. doi: 10.18653/V1/2025.ACL-LONG.369. URL https://doi.org/10.18653/v1/2025.acl-long.369.

Runxi Huang, Liyu Zhang, Shengzhong Liu, and Xiaomin Ouyang. Mobileexplorer: Accelerating on-device inference for mobile GUI agents via online exploration. *CoRR*, abs/2605.26546, 2026. doi: 10.48550/ARXIV.2605.26546. URL https://doi.org/10.48550/arXiv.2605.26546.

Saurav Kadavath, Tom Conerly, Amanda Askell, Tom Henighan, Dawn Drain, Ethan Perez, Nicholas Schiefer, Zac Hatfield-Dodds, Nova DasSarma, Eli Tran-Johnson, Scott Johnston, Sheer El Showk, Andy Jones, Nelson Elhage, Tristan Hume, Anna Chen, Yuntao Bai, Sam Bowman, Stanislav Fort, Deep Ganguli, Danny Hernandez, Josh Jacobson, Jackson Kernion, Shauna Kravec, Liane Lovitt, Kamal Ndousse, Catherine Olsson, Sam Ringer, Dario Amodei, Tom Brown, Jack Clark, Nicholas Joseph, Ben Mann, Sam McCandlish, Chris Olah, and Jared Kaplan. Language models (mostly) know what they know. *CoRR*, abs/2207.05221, 2022. doi: 10.48550/ARXIV.2207.05221. URL https://doi.org/10.48550/arXiv.2207.05221.

Robert Kirk, Ishita Mediratta, Christoforos Nalmpantis, Jelena Luketina, Eric Hambro, Edward Grefenstette, and Roberta Raileanu. Understanding the effects of RLHF on LLM generalisation and diversity. In *The Twelfth International Conference on Learning Representations, ICLR 2024, Vienna, Austria, May 7-11, 2024*. OpenReview.net, 2024. URL https://openreview.net/forum?id=PXD3FAVHJT.

Divake Kumar, Sina Tayebati, Devashri Naik, Amanda Sofie Rios, Nilesh A. Ahuja, Omesh Tickoo, Ranganath Krishnan, and Amit Ranjan Trivedi. Uncertainty quantification for computer-use agents: A benchmark across vision-language models and GUI grounding datasets. *CoRR*, abs/2606.25760, 2026. doi: 10.48550/ARXIV.2606.25760. URL https://doi.org/10.48550/arXiv.2606.25760.

Kaixin Li, Ziyang Meng, Hongzhan Lin, Ziyang Luo, Yuchen Tian, Jing Ma, Zhiyong Huang, and Tat-Seng Chua. Screenspot-pro: GUI grounding for professional high-resolution computer use. In Cathal Gurrin, Klaus Schoeffmann, Min Zhang, Luca Rossetto, Stevan Rudinac, Duc-Tien Dang-Nguyen, Wen-Huang Cheng, Phoebe Chen, and Jenny Benois-Pineau (eds.), *Proceedings of the 33rd ACM International Conference on Multimedia, MM 2025, Dublin, Ireland, October 27-31, 2025*, pp. 8778–8786. ACM, 2025. doi: 10.1145/3746027.3755688. URL https://doi.org/10.1145/3746027.3755688.

Xinyi Liu, Xiaoyi Zhang, Ziyun Zhang, and Yan Lu. Ui-e2i-synth: Advancing GUI grounding with large-scale instruction synthesis. In Wanxiang Che, Joyce Nabende, Ekaterina Shutova, and Mohammad Taher Pilehvar (eds.), *Findings of the Association for Computational Linguistics, ACL 2025, Vienna, Austria, July 27 - August 1, 2025*, volume ACL 2025 of *Findings of ACL*, pp. 15668–15684. Association for Computational Linguistics, 2025. doi: 10.18653/V1/2025.FINDINGS-ACL.809. URL https://doi.org/10.18653/v1/2025.findings-acl.809.

Yuhang Liu, Zeyu Liu, Shuanghe Zhu, Pengxiang Li, Congkai Xie, Jiasheng Wang, Xueyu Hu, Xiaotian Han, Jianbo Yuan, Xinyao Wang, Shengyu Zhang, Hongxia Yang, and Fei Wu. Infigui-g1: Advancing GUI grounding with adaptive exploration policy optimization. In Sven Koenig, Chad Jenkins, and Matthew E. Taylor (eds.), *Fortieth AAAI Conference on Artificial Intelligence, Thirty-Eighth Conference on Innovative Applications of Artificial Intelligence, Sixteenth Symposium on Educational Advances in Artificial Intelligence, AAAI 2026, Singapore, January 20-27, 2026*, pp. 32267–32275. AAAI Press, 2026. doi: 10.1609/AAAI.V40I38.40500. URL https://doi.org/10.1609/aaai.v40i38.40500.

Keming Lu, Hongyi Yuan, Runji Lin, Junyang Lin, Zheng Yuan, Chang Zhou, and Jingren Zhou. Routing to the expert: Efficient reward-guided ensemble of large language models. In Kevin Duh, Helena Gómez-Adorno, and Steven Bethard (eds.), *Proceedings of the 2024 Conference of the North American Chapter of the Association for Computational Linguistics: Human Language Technologies (Volume 1: Long Papers), NAACL 2024, Mexico City, Mexico, June 16-21, 2024*, pp. 1964–1974. Association for Computational Linguistics, 2024. doi: 10.18653/V1/2024.NAACL-LONG.109. URL https://doi.org/10.18653/v1/2024.naacl-long.109.

Andrey Malinin and Mark J. F. Gales. Uncertainty estimation in autoregressive structured prediction. In *9th International Conference on Learning Representations, ICLR 2021, Virtual Event, Austria, May 3-7, 2021*. OpenReview.net, 2021. URL https://openreview.net/forum?id=jN5y-zb5Q7m.

Potsawee Manakul, Adian Liusie, and Mark J. F. Gales. Selfcheckgpt: Zero-resource black-box hallucination detection for generative large language models. In Houda Bouamor, Juan Pino, and Kalika Bali (eds.), *Proceedings of the 2023 Conference on Empirical Methods in Natural Language Processing, EMNLP 2023, Singapore, December 6-10, 2023*, pp. 9004–9017. Association for Computational Linguistics, 2023. doi: 10.18653/V1/2023.EMNLP-MAIN.557. URL https://doi.org/10.18653/v1/2023.emnlp-main.557.

Isaac Ong, Amjad Almahairi, Vincent Wu, Wei-Lin Chiang, Tianhao Wu, Joseph E. Gonzalez, M. Waleed Kadous, and Ion Stoica. Routellm: Learning to route llms from preference data. In *The Thirteenth International Conference on Learning Representations, ICLR 2025, Singapore, April 24-28, 2025*. OpenReview.net, 2025. URL https://openreview.net/forum?id=8sSqNntaMr.

OpenAI. Computer-Using Agent: Introducing a universal interface for AI to interact with the digital world, January 2025. URL https://openai.com/index/computer-using-agent.

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, Xiaojun Xiao, Kai Cai, Chuang Li, Yaowei Zheng, Chaolin Jin, Chen Li, Xiao Zhou, Minchao Wang, Haoli Chen, Zhaojian Li, Haihua Yang, Haifeng Liu, Feng Lin, Tao Peng, Xin Liu, and Guang Shi. UI-TARS: pioneering automated GUI interaction with native agents. *CoRR*, abs/2501.12326, 2025. doi: 10.48550/ARXIV.2501.12326. URL https://doi.org/10.48550/arXiv.2501.12326.

Victor Quach, Adam Fisch, Tal Schuster, Adam Yala, Jae Ho Sohn, Tommi S. Jaakkola, and Regina Barzilay. Conformal language modeling. In *The Twelfth International Conference on Learning Representations, ICLR 2024, Vienna, Austria, May 7-11, 2024*. OpenReview.net, 2024. URL https://openreview.net/forum?id=pzUhfQ74c5.

Qwen Team. Qwen3.5: Towards native multimodal agents, February 2026. URL https://qwen.ai/blog?id=qwen3.5.

Allen Z. Ren, Anushri Dixit, Alexandra Bodrova, Sumeet Singh, Stephen Tu, Noah Brown, Peng Xu, Leila Takayama, Fei Xia, Jake Varley, Zhenjia Xu, Dorsa Sadigh, Andy Zeng, and Anirudha Majumdar. Robots that ask for help: Uncertainty alignment for large language model planners. In Jie Tan, Marc Toussaint, and Kourosh Darvish (eds.), *Conference on Robot Learning, CoRL 2023, 6-9 November 2023, Atlanta, GA, USA*, volume 229 of *Proceedings of Machine Learning Research*, pp. 661–682. PMLR, 2023. URL https://proceedings.mlr.press/v229/ren23a.html.

Xingjian Tao, Yiwei Wang, Yujun Cai, Zhicheng Yang, and Jing Tang. Understanding GUI agent localization biases through logit sharpness. In Christos Christodoulopoulos, Tanmoy Chakraborty, Carolyn Rose, and Violet Peng (eds.), *Findings of the Association for Computational Linguistics: EMNLP 2025, Suzhou, China, November 4-9, 2025*, pp. 23361–23374. Association for Computational Linguistics, 2025. doi: 10.18653/V1/2025.FINDINGS-EMNLP.1268. URL https://doi.org/10.18653/v1/2025.findings-emnlp.1268.

Vladimir Vovk, Alexander Gammerman, and Glenn Shafer. *Algorithmic learning in a random world*. Springer, 2005.

Qingni Wang, Yue Fan, and Xin Eric Wang. Safeground: Know when to trust GUI grounding models via uncertainty calibration. *CoRR*, abs/2602.02419, 2026. doi: 10.48550/ARXIV.2602.02419. URL https://doi.org/10.48550/arXiv.2602.02419.

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, Yiqian Liu, Xiao Zhang, Yanting Zhang, Xiangyu Yue, Weijie Su, Xizhou Zhu, Wei Shen, Jifeng Dai, and Wenhai Wang. Mmbench-gui: Hierarchical multi-platform evaluation framework for GUI agents. *CoRR*, abs/2507.19478, 2025. doi: 10.48550/ARXIV.2507.19478. URL https://doi.org/10.48550/arXiv.2507.19478.

Xuezhi Wang, Jason Wei, Dale Schuurmans, Quoc V. Le, Ed H. Chi, Sharan Narang, Aakanksha Chowdhery, and Denny Zhou. Self-consistency improves chain of thought reasoning in language models. In *The Eleventh International Conference on Learning Representations, ICLR 2023, Kigali, Rwanda, May 1-5, 2023*. OpenReview.net, 2023. URL https://openreview.net/forum?id=1PL1NIMMrw.

Zhiyong Wu, Zhenyu Wu, Fangzhi Xu, Yian Wang, Qiushi Sun, Chengyou Jia, Kanzhi Cheng, Zichen Ding, Liheng Chen, Paul Pu Liang, and Yu Qiao. OS-ATLAS: foundation action model for generalist GUI agents. In *The Thirteenth International Conference on Learning Representations, ICLR 2025, Singapore, April 24-28, 2025*. OpenReview.net, 2025. URL https://openreview.net/forum?id=n9PDaFNi8t.

Yan Yang, Dongxu Li, Yutong Dai, Yuhao Yang, Ziyang Luo, Zirui Zhao, Zhiyuan Hu, Junzhe Huang, Amrita Saha, Zeyuan Chen, Ran Xu, Liyuan Pan, Caiming Xiong, and Junnan Li. GTA1: GUI test-time scaling agent. *CoRR*, abs/2507.05791, 2025a. doi: 10.48550/ARXIV.2507.05791. URL https://doi.org/10.48550/arXiv.2507.05791.

Zhen Yang, Zi-Yi Dou, Di Feng, Forrest Huang, Anh Nguyen, Keen You, Omar Attia, Yuhao Yang, Michael Feng, Haotian Zhang, Ram Ramrakhya, Chao Jia, Jeffrey Nichols, Alexander Toshev, Yinfei Yang, and Zhe Gan. Ferret-ui lite: Lessons from building small on-device GUI agents. *CoRR*, abs/2509.26539, 2025b. doi: 10.48550/ARXIV.2509.26539. URL https://doi.org/10.48550/arXiv.2509.26539.

Shunyu Yao, Jeffrey Zhao, Dian Yu, Nan Du, Izhak Shafran, Karthik R. Narasimhan, and Yuan Cao. React: Synergizing reasoning and acting in language models. In *The Eleventh International Conference on Learning Representations, ICLR 2023, Kigali, Rwanda, May 1-5, 2023*. OpenReview.net, 2023. URL https://openreview.net/forum?id=WE_vluYUL-X.

Murong Yue, Jie Zhao, Min Zhang, Liang Du, and Ziyu Yao. Large language model cascades with mixture of thought representations for cost-efficient reasoning. In *The Twelfth International Conference on Learning Representations, ICLR 2024, Vienna, Austria, May 7-11, 2024*. OpenReview.net, 2024. URL https://openreview.net/forum?id=6okaSfANzh.

Shaojie Zhang, Pei Fu, Ruoceng Zhang, Jiahui Yang, Anan Du, Xiuwen Xi, Shaokang Wang, Ying Huang, Bin Qin, Zhenbo Luo, and Jian Luan. Hyperclick: Advancing reliable GUI grounding via uncertainty calibration. *CoRR*, abs/2510.27266, 2025. doi: 10.48550/ARXIV.2510.27266. URL https://doi.org/10.48550/arXiv.2510.27266.

Boyuan Zheng, Boyu Gou, Jihyung Kil, Huan Sun, and Yu Su. Gpt-4v(ision) is a generalist web agent, if grounded. In Ruslan Salakhutdinov, Zico Kolter, Katherine A. Heller, Adrian Weller, Nuria Oliver, Jonathan Scarlett, and Felix Berkenkamp (eds.), *Forty-first International Conference on Machine Learning, ICML 2024, Vienna, Austria, July 21-27, 2024*, volume 235 of *Proceedings of Machine Learning Research*, pp. 61349–61385. PMLR / OpenReview.net, 2024. URL https://proceedings.mlr.press/v235/zheng24e.html.

APPENDIX

A Method Details **15**

A.1 Candidate Set Construction 15

A.2 Local View Construction 16

B Algorithms **17**

C Experimental Details **17**

C.1 Benchmarks 17

C.2 Inference Settings 18

C.3 Baselines 18

D Additional Results **20**

D.1 Detailed Results on Each Benchmark 20

D.2 Full Results of Uncertainty Estimation 21

E Case Studies **21**

E.1 Examples 22

E.2 Digit Distributions 22

E.3 Failure Mode Analysis 22

A METHOD DETAILS

This section details the method of SCOPE, including the construction of the candidate set and local view. The complete algorithm pseudocode is given in Appendix B.

A.1 CANDIDATE SET CONSTRUCTION

Let $\mathcal{D} = \{0, \ldots, 9\}$ be the digit set and $p_j(d)$ the probability of digit $d \in \mathcal{D}$ at position $j \in \mathcal{C}$. At each position we keep the top-4 digits and renormalize their probabilities:

$$\mathcal{D}_j = \{d \mid p_j(d) \in \mathrm{Top}_4(\{p_j(d') \mid d' \in \mathcal{D}\})\}, \tag{17}$$

$$\pi_j(d) = \frac{p_j(d)}{\sum_{d' \in \mathcal{D}_j} p_j(d')}, \tag{18}$$

where $\mathrm{Top}_k(\cdot)$ returns the k largest elements of a set, $\mathcal{D}_j$ is the set of the most probable digits at position j, and $\pi_j(d)$ is $p_j(d)$ renormalized within $\mathcal{D}_j$. Let $\mathcal{C}_x, \mathcal{C}_y \subseteq \mathcal{C}$ be the digit positions of the x and y coordinates. For each axis $a \in \{x, y\}$, we take the product of the per-digit probabilities of a coordinate $\mathbf{d} = (d_j)_{j \in \mathcal{C}_a}$ as its candidate probability and keep the most probable coordinates:

$$\pi_a(\mathbf{d}) = \prod\nolimits_{j \in \mathcal{C}_a} \pi_j(d_j), \tag{19}$$

$$\mathcal{Q}_a = \{\mathbf{d} \mid \pi_a(\mathbf{d}) \in \mathrm{Top}_{10}(\{\pi_a(\mathbf{d}') \mid \mathbf{d}' \in \prod\nolimits_{j \in \mathcal{C}_a} \mathcal{D}_j\})\}, \tag{20}$$

where $\prod_{j \in \mathcal{C}_a} \mathcal{D}_j$ is the set of all candidate coordinates on axis a, and $\mathcal{Q}_a$ is the set of the most probable coordinates on axis a. We then weight a candidate coordinate $p = (\mathbf{d}_x, \mathbf{d}_y)$ by the product of its axis probabilities and keep the m most probable coordinates as the candidate set:

$$\pi(p) = \pi_x(\mathbf{d}_x)\, \pi_y(\mathbf{d}_y), \tag{21}$$

$$\mathcal{P} = \{p \mid \pi(p) \in \mathrm{Top}_m(\{\pi(p') \mid p' \in \mathcal{Q}_x \times \mathcal{Q}_y\})\}, \tag{22}$$

Algorithm 1 Spatial confidence

Input: screenshot $\mathcal{V}$, instruction $\mathcal{T}$, edge agent $\mathcal{A}_e$, candidate count m, view fraction β
Output: edge prediction $\hat{p}_e$, scale-profile score s_{scale}, view-consistency score s_{view}

$\hat{p}_e \leftarrow \mathcal{A}_e(\mathcal{V}, \mathcal{T})$ ▷ recording $p_t^{(k)}$ for $t = 1, \ldots, L$
$\mathcal{C} \leftarrow$ coordinate digit positions of $\hat{p}_e$
$\gamma_j \leftarrow \log p_j^{(1)} - \log p_j^{(5)}; \quad \bar{\gamma} \leftarrow \frac{1}{|\mathcal{C}|} \sum_{j \in \mathcal{C}} \gamma_j$
$g \leftarrow \sqrt{\frac{1}{|\mathcal{C}|} \sum_{j \in \mathcal{C}} (\gamma_j - \bar{\gamma})^2}$
$\bar{H} \leftarrow -\frac{1}{L} \sum_{t=1}^{L} \sum_{k=1}^{K} p_t^{(k)} \log p_t^{(k)}$
$s_{\text{scale}} \leftarrow g \cdot e^{-\bar{H}}$ ▷ scale-profile score
$\mathcal{P} \leftarrow \{p \mid \pi(p) \in \text{Top}_m(\{\pi(p') \mid p' \in \mathcal{Q}_x \times \mathcal{Q}_y\})\}$ ▷ Appendix A.1
$[x_{\min}, x_{\max}] \times [y_{\min}, y_{\max}] \leftarrow$ bounding box of $\mathcal{P}$
$[x_1, x_2] \leftarrow [\frac{x_{\min}+x_{\max}}{2} \mp \frac{1}{2} \max(x_{\max} - x_{\min}, \beta W)] \cap [0, W]$
$[y_1, y_2] \leftarrow [\frac{y_{\min}+y_{\max}}{2} \mp \frac{1}{2} \max(y_{\max} - y_{\min}, \beta H)] \cap [0, H]$
$\hat{p}_{\text{crop}} \leftarrow (x_1, y_1) + \mathcal{A}_e(\mathcal{V}|_{[x_1,x_2]\times[y_1,y_2]}, \mathcal{T})$
$s_{\text{view}} \leftarrow -\|\hat{p}_e - \hat{p}_{\text{crop}}\|_2 / D$ ▷ view-consistency score
return $\hat{p}_e, s_{\text{scale}}, s_{\text{view}}$

Algorithm 2 Offloading with per-element guarantees

Input: calibration set $\{(\hat{p}_{e,i}, s_{\text{scale},i}, s_{\text{view},i}, \mathcal{B}_i)\}_{i=1}^{n}$ from Algorithm 1, risk level α
Input: test input $(\mathcal{V}, \mathcal{T})$ with $(\hat{p}_e, s_{\text{scale}}, s_{\text{view}})$ from Algorithm 1, cloud agent $\mathcal{A}_c$
Output: offloading decision δ, final collaborative prediction $\hat{p}^*$

$\mu_{\text{s}} \leftarrow \text{median}(\{s_{\text{scale},i}\}); \quad M_{\text{s}} \leftarrow \text{MAD}(\{s_{\text{scale},i}\})$ ▷ calibration time
$\mu_{\text{v}} \leftarrow \text{median}(\{s_{\text{view},i}\}); \quad M_{\text{v}} \leftarrow \text{MAD}(\{s_{\text{view},i}\})$
$r_i \leftarrow (s_{\text{scale},i} - \mu_{\text{s}})/M_{\text{s}} + (s_{\text{view},i} - \mu_{\text{v}})/M_{\text{v}}$
$\mu_r \leftarrow \text{median}(\{r_i\}); \quad M_r \leftarrow \text{MAD}(\{r_i\})$
$u_i \leftarrow e^{-(r_i - \mu_r)/M_r}$
$d_i \leftarrow \text{dist}(\hat{p}_{e,i}, \mathcal{B}_i)/D_i; \quad w_i \leftarrow \sqrt{b_{w,i}^2 + b_{h,i}^2}/(2D_i)$
$V_i \leftarrow d_i/(w_i \cdot u_i)$
$\hat{q}(\alpha) \leftarrow \text{Quantile}_{\lceil (n+1)(1-\alpha) \rceil / n}(\{V_i\})$
$r \leftarrow (s_{\text{scale}} - \mu_{\text{s}})/M_{\text{s}} + (s_{\text{view}} - \mu_{\text{v}})/M_{\text{v}}$ ▷ test time
$u \leftarrow e^{-(r - \mu_r)/M_r}$
$R \leftarrow \hat{q}(\alpha) \cdot u$
$\delta \leftarrow \mathbb{I}(R > 1)$
$\hat{p}^* \leftarrow (1 - \delta)\, \hat{p}_e + \delta\, \mathcal{A}_c(\mathcal{V}, \mathcal{T})$
return $\delta, \hat{p}^*$

where $\mathcal{Q}_x \times \mathcal{Q}_y$ is the set of all candidates, and $\mathcal{P}$ is the set of the most probable coordinates. By construction, $\hat{p}_e \in \mathcal{P}$ always holds, as it is composed of the most probable digit at each position.

A.2 Local View Construction

Let $x_{\min}, x_{\max}$ and $y_{\min}, y_{\max}$ be the extreme coordinates of $\mathcal{P}$, $W \times H$ the size of the screenshot $\mathcal{V}$, and β the view fraction. The local view widens the bounding box of $\mathcal{P}$ about its center to at least β times the width and height of the screenshot and clips it to the screenshot:

$$[x_1, x_2] = \left[\frac{x_{\min}+x_{\max}}{2} \mp \frac{1}{2} \max(x_{\max} - x_{\min}, \beta W)\right] \cap [0, W], \tag{23}$$

$$[y_1, y_2] = \left[\frac{y_{\min}+y_{\max}}{2} \mp \frac{1}{2} \max(y_{\max} - y_{\min}, \beta H)\right] \cap [0, H]. \tag{24}$$

The agent grounds the same instruction on the cropped view with the same preprocessing, and the prediction is mapped back to the screenshot by adding the crop offset:

$$\hat{p}_{\text{crop}} = (x_1, y_1) + \mathcal{A}_e(\mathcal{V}|_{[x_1,x_2]\times[y_1,y_2]}, \mathcal{T}). \tag{25}$$

Table 6: **Prompts of the edge and cloud agents.** {schema} in Holo1.5-3B stands for the JSON schema of its click action, whose fields are `action`, `x`, and `y`.

InfiGUI-G1-3B (system)

```
You FIRST think about the reasoning process as an internal monologue and
then provide the final answer.
The reasoning process MUST BE enclosed within <think> </think> tags.
```

InfiGUI-G1-3B (user)

```
<image>The screen's resolution is {width}x{height}.
Locate the UI element(s) for "{instruction}", output the coordinates
using JSON format: [{"point_2d": [x, y]}, ...]
```

Holo1.5-3B (user)

```
Localize an element on the GUI image according to the provided target and
output a click position.
* You must output a valid JSON following the format: {schema}
Your target is:
{instruction}
```

Qwen3.5-4B and Qwen3.5-35B-A3B (user)

```
Locate the GUI element matching the target description below and report a
single click point.

Target: {instruction}

Output a JSON object in the exact form (no extra text, no bbox):
{"point_2d": [x, y], "label": "<short element name>"}
Coordinates x and y are integers in [0, 1000], normalized to the
screenshot's width and height respectively.
```

B ALGORITHMS

In this section, we present the algorithms of SCOPE. Algorithm 1 computes the spatial confidence on the edge, and Algorithm 2 calibrates it into the conformal click radius and makes the offloading decision. For more details, please refer to Section 3 and Appendix A.

C EXPERIMENTAL DETAILS

This section presents the benchmarks, inference settings, and baselines summarized in Section 4.1.

C.1 BENCHMARKS

We evaluate on three public GUI grounding benchmarks, whose details are as follows.

- **ScreenSpot-Pro** covers professional applications whose screenshots are high-resolution and whose targets are tiny and surrounded by dense distractors. It contains 1,581 samples across six domains, including creative (341), development (299), CAD (261), scientific (254), office (230), and operating system (196), on Windows (927), macOS (604), and Linux (50), where the target elements are categorized into text (977) and icons (604).
- **UI-I2E-Bench** focuses on instruction understanding, since most instructions refer to the target implicitly by its function or position rather than by visible text. It contains 1,477 samples on mobile (705), desktop (519), and web (253) screenshots, where the instructions are implicit (917) or explicit (560).
- **MMBench-GUI** spans six platforms across desktop, mobile, and web. We use its element grounding level, which categorizes instructions into basic ones describing the appearance and position of the element and advanced ones referring to its function. It contains 3,594

Table 7: **Detailed results on ScreenSpot-Pro.** Accuracy (%) of the cloud agent (Cloud), of each edge agent (Edge), and of the device-cloud system (Ours), and the offloading rate (%, Off.) at $\alpha = 0.1$. Text and Icon denote the type of the target element.

		Development		Creative		CAD		Scientific		Office		OS		Overall		
		Text	Icon	Text	Icon	Text	Icon	Text	Icon	Text	Icon	Text	Icon	Text	Icon	Avg
Cloud		83.8	52.4	73.2	37.8	70.1	32.8	76.4	38.2	85.3	62.3	78.5	57.3	77.5	45.9	65.4
InfiGUI	Edge	66.2	17.9	51.5	14.7	49.2	25.0	67.4	31.8	71.8	28.3	47.7	16.9	59.0	21.2	44.5
	Ours	83.8	52.3	73.1	37.9	69.7	32.3	76.2	37.7	84.7	63.1	78.5	56.7	77.2	45.7	65.3
	Off.	100.0	100.0	100.0	100.0	100.0	100.0	100.0	100.0	100.0	100.0	100.0	100.0	100.0	100.0	100.0
Holo	Edge	70.1	28.3	61.1	22.4	50.3	29.7	70.8	38.2	79.1	39.6	50.5	28.1	63.9	29.8	50.9
	Ours	83.8	52.3	73.1	37.9	69.7	32.3	76.2	37.7	84.7	63.1	78.5	56.7	77.2	45.7	65.3
	Off.	100.0	100.0	100.0	100.0	100.0	100.0	100.0	100.0	100.0	100.0	100.0	100.0	100.0	100.0	100.0
Qwen	Edge	72.7	40.0	65.2	22.4	65.0	29.7	79.2	36.4	82.5	45.3	66.4	37.1	71.6	34.1	57.3
	Ours	83.7	52.4	73.6	37.9	69.7	32.3	75.8	37.7	84.7	63.1	79.3	56.8	77.4	45.8	65.3
	Off.	91.6	99.0	99.3	99.9	99.2	99.6	96.1	97.2	95.1	99.1	95.3	97.5	96.4	98.7	97.3

Table 8: **Detailed results on UI-I2E-Bench.** Accuracy (%) of the cloud agent (Cloud), of each edge agent (Edge), and of the device-cloud system (Ours), and the offloading rate (%, Off.) at $\alpha = 0.1$. Columns are grouped by platform, instruction type, and element type.

		Platform			Instruction		Element							Avg
		Web	Desktop	Mobile	Explicit	Implicit	Button	Icon	Drop.	Input	Toggle	Check.	Radio	
Cloud		90.1	72.1	85.0	87.3	77.6	82.9	82.9	85.4	83.3	68.7	88.0	96.0	81.3
InfiGUI	Edge	80.6	60.1	78.7	80.9	67.4	77.7	69.2	85.4	71.3	64.4	74.7	68.0	72.5
	Ours	88.1	70.7	84.9	87.2	76.3	78.7	78.6	87.1	82.5	75.5	85.0	82.2	80.4
	Off.	40.8	64.9	37.1	44.5	49.3	43.1	35.1	53.0	73.5	38.4	23.9	30.5	47.5
Holo	Edge	79.1	68.0	76.5	85.4	67.0	74.7	67.1	86.0	70.7	77.7	81.3	56.0	73.9
	Ours	90.1	77.2	83.1	90.8	77.0	79.5	80.3	88.7	81.1	80.6	91.9	87.0	82.2
	Off.	36.3	39.4	39.2	31.2	43.4	34.6	27.5	42.2	59.9	26.7	24.6	56.1	38.8
Qwen	Edge	89.7	63.0	77.7	83.4	69.2	72.0	73.1	89.3	83.3	52.5	92.0	82.0	74.6
	Ours	91.9	71.8	84.6	87.6	77.5	79.7	81.5	89.9	85.6	67.3	93.9	87.6	81.4
	Off.	37.4	67.3	49.5	47.9	57.2	43.6	35.7	60.6	82.8	52.2	23.3	40.8	53.7

samples on Android (711), macOS (691), iOS (644), web (618), Windows (543), and Linux (387), where the instructions are basic (1,787) or advanced (1,807).

C.2 Inference Settings

Inference. All agents run with vLLM 0.19.1 in bfloat16. The edge agents decode in batches of 256 requests on one GPU, and the cloud agent in batches of 32 requests over two GPUs. Every prediction comes from greedy decoding. The first decode of the full view retains the top-20 log probabilities of each generated token, which provide the coordinate digit distributions for the scale-profile score, the candidate set of Appendix A.1, and the logit-based baselines. The second decode of the local view is greedy without log probabilities.

Prompts. Table 6 lists the prompts, which follow the official prompt templates. InfiGUI-G1 reasons inside `<think>` tags before its answer as its system prompt requires, while the thinking mode of Qwen3.5 is disabled and Holo1.5 answers directly. Every agent ends its answer with a short JSON object, from which the x and y coordinates can be parsed. The same prompt is used for the full view and the local view.

C.3 Baselines

All baselines use the same prompts as SCOPE. Every score is oriented so that a larger score indicates higher confidence.

Random. Each sample is randomly assigned an independent score between 0 and 1. The expected AUCG of this baseline is 0.5, which serves as the lower bound.

Table 9: **Detailed results on MMBench-GUI.** Accuracy (%) of the cloud agent (Cloud), of each edge agent (Edge), and of the device-cloud system (Ours), and the offloading rate (%, Off.) at $\alpha = 0.1$. Basic and Adv. denote basic and advanced instructions.

		Windows		macOS		Linux		iOS		Android		Web		Overall		
		Basic	Adv.	Basic	Adv.	Basic	Adv.	Basic	Adv.	Basic	Adv.	Basic	Adv.	Basic	Adv.	Avg
Cloud		90.4	75.7	85.5	78.6	83.8	72.4	96.2	92.1	97.5	91.3	96.5	86.4	92.2	83.8	88.0
InfiGUI	Edge	73.1	47.4	80.3	54.9	64.4	42.3	95.2	78.5	91.6	76.3	90.0	64.6	84.1	62.6	73.3
	Ours	86.6	69.6	86.8	72.5	78.7	69.4	96.2	86.1	95.8	86.8	94.5	80.7	90.7	78.4	84.5
	Off.	36.3	56.4	30.2	49.0	45.4	61.4	26.4	39.9	21.5	31.6	23.9	41.8	29.3	45.2	37.3
Holo	Edge	83.8	54.8	81.7	63.6	71.7	50.5	94.3	78.5	92.7	81.1	90.3	70.1	86.8	68.1	77.4
	Ours	89.2	68.8	90.6	77.9	79.6	66.2	95.5	85.7	96.6	87.8	95.5	81.3	92.1	79.2	85.6
	Off.	21.1	42.9	23.7	36.6	26.7	40.7	22.2	33.1	21.9	28.8	22.1	38.9	22.7	36.2	29.5
Qwen	Edge	81.5	59.2	84.9	67.9	75.9	59.7	95.9	83.6	94.9	87.0	93.2	76.3	88.8	73.8	81.2
	Ours	86.0	68.8	89.9	75.6	79.1	65.8	96.5	90.6	97.6	89.9	94.7	82.1	91.7	80.1	85.9
	Off.	13.0	29.4	17.8	32.7	11.2	27.2	13.2	25.8	14.7	24.8	15.0	26.7	14.5	27.8	21.2

LN-Confidence. LN-Confidence averages the log probability of the generated token over the L decoded positions. Under greedy decoding the generated token is the most probable one, so the score averages the log probability of the top-1 token at each position:

$$s_{\mathrm{LNC}} = \frac{1}{L} \sum_{t=1}^{L} \log p_t^{(1)}. \tag{26}$$

LN-Entropy. LN-Entropy averages the negative entropy over the same positions, which is the same as the negative global hesitation of the scale-profile score:

$$s_{\mathrm{LNE}} = -\bar{H}. \tag{27}$$

Self-Consistency. Self-Consistency samples $N = 5$ additional answers at temperature 0.4 with top-p 0.95 and measures their spatial dispersion. Let $\mathcal{N}$ be the set of sampled answers. The score is calculated as the negative standard deviation of the x and y coordinate components:

$$s_{\mathrm{SC}} = -\sqrt{\frac{1}{|\mathcal{N}|} \sum_{\ell \in \mathcal{N}} (x^{(\ell)} - \bar{x})^2 + \frac{1}{|\mathcal{N}|} \sum_{\ell \in \mathcal{N}} (y^{(\ell)} - \bar{y})^2}, \tag{28}$$

where $x^{(\ell)}$ and $y^{(\ell)}$ are the coordinates of answer ℓ, and $\bar{x}$ and $\bar{y}$ are their means over $\mathcal{N}$.

PSS. PSS measures the sharpness of the digit distribution at each coordinate position around its most likely digit. Let $p_j(d)$ be the probability of digit $d \in \mathcal{D}$ at position $j \in \mathcal{C}$, and let $d_j^\star = \arg\max p_j(d)$ be the peak. The sharpness is the height of the peak above the two ends of the digit range, or twice the difference between the two ends when the peak itself lies at an end:

$$\omega_j = \begin{cases} \big(2p_j(d_j^\star) - p_j(0) - p_j(9)\big)/9, & 0 < d_j^\star < 9, \\ 2\,|p_j(9) - p_j(0)|/9, & \text{otherwise.} \end{cases} \tag{29}$$

The score averages the sharpness times the peak probability over the coordinate digits, scaled by the constant 4.5 of the original score:

$$s_{\mathrm{PSS}} = \frac{4.5}{|\mathcal{C}|} \sum_{j \in \mathcal{C}} \omega_j \cdot p_j(d_j^\star). \tag{30}$$

SafeGround. SafeGround samples $N = 10$ additional answers at temperature 1.0 and measures how concentrated they are on the screen. The sampled answers are counted on a grid of patches, the patches whose density exceeds 30% of the maximum are grouped into connected regions, and the regions are ranked by their mean density and normalized. The official uncertainty combines three terms, where U_{TA} decreases with the margin between the two densest regions, U_{IE} is the normalized entropy of the region densities, and U_{CD} decreases with their concentration. The score is the negative of the weighted sum:

$$s_{\mathrm{SG}} = -(0.2\,U_{\mathrm{TA}} + 0.2\,U_{\mathrm{IE}} + 0.6\,U_{\mathrm{CD}}). \tag{31}$$

Table 10: **Full results of uncertainty estimation.** We report the four metrics of Table 2 for all nine configurations. The best and the second best results are highlighted in bold and underlined.

Methods	ScreenSpot-Pro			UI-I2E-Bench			MMBench-GUI			Mean
	InfiGUI	Holo	Qwen	InfiGUI	Holo	Qwen	InfiGUI	Holo	Qwen	
AUROC										
Random	0.497	0.502	0.480	0.491	0.470	0.501	0.512	0.510	0.504	0.496
LN-Confidence	0.625	0.756	0.801	0.524	0.524	0.696	0.593	0.561	0.771	0.650
LN-Entropy	0.653	0.824	**0.836**	0.537	0.624	0.734	0.626	0.661	0.803	0.700
Self-Consist.	0.678	0.741	0.791	0.589	0.595	0.698	0.592	0.622	0.769	0.675
PSS	0.647	0.755	0.810	0.498	0.599	0.690	0.515	0.633	0.797	0.660
SafeGround	0.711	0.791	0.827	0.589	0.621	**0.744**	0.626	0.650	0.792	0.706
Ours (s_{scale})	0.793	0.814	0.808	0.655	**0.740**	0.671	**0.782**	**0.791**	**0.816**	0.763
Ours (+ s_{view})	**0.842**	**0.838**	0.821	**0.691**	**0.740**	0.684	0.778	0.781	0.798	**0.775**
AURC↓										
Random	0.558	0.494	0.438	0.276	0.270	0.243	0.258	0.220	0.184	0.327
LN-Confidence	0.474	0.319	0.227	0.268	0.223	0.150	0.229	0.192	0.084	0.241
LN-Entropy	0.440	0.266	0.200	0.261	0.180	0.134	0.206	0.148	0.072	0.212
Self-Consist.	0.458	0.347	0.247	0.246	0.211	0.166	0.245	0.188	0.090	0.244
PSS	0.466	0.305	0.206	0.285	0.192	0.151	0.273	0.159	0.074	0.234
SafeGround	0.407	0.300	0.214	0.235	0.188	**0.129**	0.205	0.168	0.078	0.214
Ours (s_{scale})	0.332	0.255	0.207	0.193	0.133	0.170	0.118	0.093	**0.065**	0.174
Ours (+ s_{view})	**0.292**	**0.236**	**0.196**	**0.159**	**0.126**	0.160	**0.117**	**0.092**	0.066	**0.160**
bgt@.9↓										
Random	0.920	0.930	0.920	0.880	0.880	0.900	0.900	0.900	0.900	0.903
LN-Confidence	0.900	0.670	0.520	0.950	0.690	0.780	0.870	0.860	0.550	0.754
LN-Entropy	0.880	0.540	0.460	0.940	0.620	0.650	0.840	0.770	0.460	0.684
Self-Consist.	0.900	0.720	**0.350**	0.940	0.620	0.800	0.930	0.840	0.410	0.723
PSS	0.900	0.700	0.470	0.960	0.660	0.790	0.920	0.810	0.420	0.737
SafeGround	0.810	0.640	0.410	0.740	0.550	**0.400**	0.810	0.810	0.450	0.624
Ours (s_{scale})	0.740	0.580	0.570	0.870	0.360	0.590	0.670	0.630	0.520	0.614
Ours (+ s_{view})	**0.640**	**0.484**	0.464	**0.517**	**0.316**	0.550	**0.620**	**0.460**	**0.409**	**0.495**
acc@.3										
Random	0.503	0.546	0.588	0.752	0.756	0.774	0.778	0.807	0.831	0.704
LN-Confidence	0.515	0.581	0.624	0.763	0.764	0.786	0.794	0.819	0.866	0.724
LN-Entropy	0.515	0.590	0.629	0.766	0.783	0.794	0.795	0.829	0.866	0.730
Self-Consist.	0.531	0.599	**0.637**	0.777	0.790	0.797	0.806	0.830	**0.869**	0.737
PSS	0.513	0.580	0.615	0.750	0.783	0.784	0.784	0.824	0.868	0.722
SafeGround	0.539	0.594	0.633	0.779	0.785	**0.800**	0.807	0.830	0.863	0.737
Ours (s_{scale})	0.529	0.593	0.612	0.774	0.806	0.791	0.821	0.841	0.861	0.737
Ours (+ s_{view})	**0.542**	**0.601**	0.627	**0.786**	**0.811**	0.794	**0.835**	**0.856**	0.867	**0.747**

Its native offloading rule selects an uncertainty threshold on the calibration set for a user-specified risk level on the accepted error rate, and a test sample is offloaded when its uncertainty exceeds the threshold. We set the risk level to 0.109, corresponding to the accepted error of SCOPE at $\alpha = 0.1$, so that the two rules are compared at the same risk.

D Additional Results

This section reports detailed results of SCOPE, including the results on each benchmark broken down by category and the full results of uncertainty estimation on each configuration.

D.1 Detailed Results on Each Benchmark

Tables 7, 8, and 9 break down the results by the categories of each benchmark. For each category, we report the accuracy of the cloud agent, the accuracy of each edge agent, the accuracy of the device-cloud system under SCOPE, and its offloading rate at $\alpha = 0.1$.

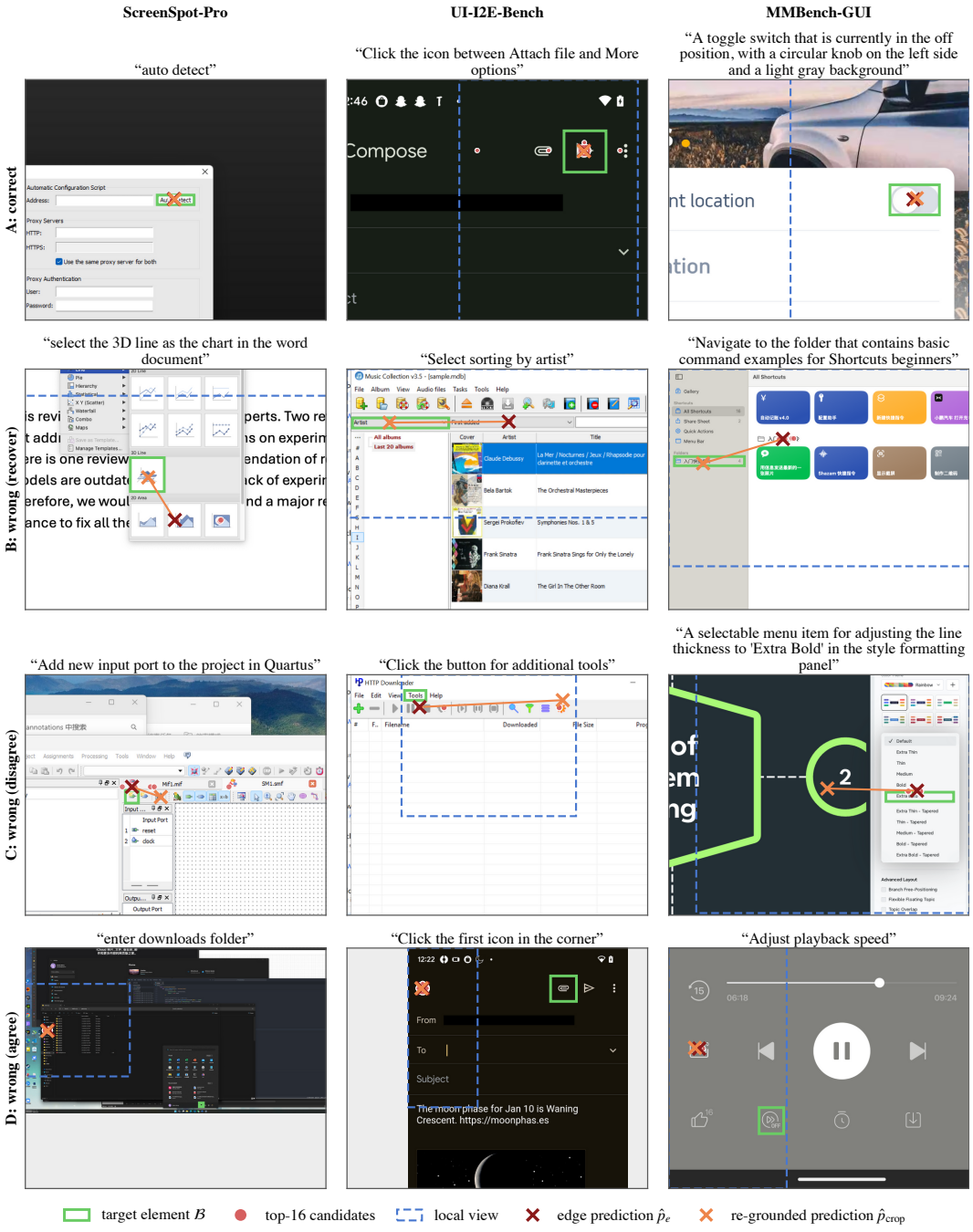


Figure 7: Visualization of twelve cases on the Qwen3.5-4B edge agent.

D.2 Full Results of Uncertainty Estimation

Table 10 details the four metrics of Table 2 by configuration.

E Case Studies

This section shows how SCOPE behaves on different cases of the Qwen3.5-4B edge agent. For each benchmark, we select one case of each of four behaviors: (A) a correct prediction, and three wrong predictions whose second prediction (B) recovers the target, (C) disagrees with the edge prediction, or (D) agrees with it. SCOPE keeps case A and offloads cases B and C, while case D is

Table 11: **Score details of the twelve cases.** g, $\bar{H}$, s_{scale}, and s_{view} are defined in Section 3. Disp. is $\|\hat{p}_e - \hat{p}_{\text{crop}}\|_2/w$, the displacement between the two predictions in units of the target radius. $\Pr(\delta = 0)$ is the fraction of the 100 random splits of Section 4.1 in which the prediction is kept on the device at $\alpha = 0.1$. Correct marks whether the edge prediction $\hat{p}_e$ is correct.

Benchmark	Case	g	$\bar{H}$	s_{scale}	s_{view}	Disp.	$\Pr(\delta = 0)$	Correct
ScreenSpot-Pro	A	4.52	0.12	4.00	−0.001	0.04	0.98	✓
	B	2.27	0.19	1.87	−0.049	2.09	0.00	×
	C	2.06	0.24	1.63	−0.027	3.35	0.00	×
	D	3.76	0.18	3.14	−0.004	0.51	0.07	×
UI-I2E-Bench	A	5.15	0.14	4.50	−0.001	0.02	1.00	✓
	B	2.72	0.24	2.14	−0.157	2.46	0.00	×
	C	1.21	0.26	0.93	−0.257	12.14	0.00	×
	D	4.37	0.11	3.91	−0.003	0.08	1.00	×
MMBench-GUI	A	5.10	0.10	4.61	−0.003	0.12	1.00	✓
	B	1.90	0.20	1.56	−0.080	2.36	0.00	×
	C	3.01	0.17	2.53	−0.090	2.63	0.00	×
	D	4.26	0.17	3.60	−0.005	0.24	1.00	×

its typical failure mode. Figure 7 and Table 11 present the cases and their scores, Figure 8 their digit distributions, and Table 12 the failure mode analysis.

E.1 Examples

Figure 7 presents an overlaid view for each case, including the target element $\mathcal{B}$, the top-16 candidates, the local view, the edge prediction $\hat{p}_e$, and the re-grounded prediction $\hat{p}_{\text{crop}}$. In case A, the candidates cluster within the target, and the two predictions are consistent. In cases B, C, and D, the candidates escape as a whole to a competing element, and the cases differ in the second prediction. In case B, the local view still contains the target, and the second prediction recovers it. In case C, the local view also contains the target, but the second prediction does not ground on it, so the two predictions disagree although neither is correct. In case D, the second prediction confirms the edge prediction on the same wrong element. Table 11 lists the score details of these cases. Both the scale-profile score and the view-consistency score separate cases B and C from case A, but neither separates case D.

E.2 Digit Distributions

Figure 8 shows the token distributions at the coordinate digits of the twelve cases. The edge agent in case A is relatively confident at the high-order digits and hesitates mostly at the units digit, so the hesitation separates by scale and the steepness g is large. The B and C cases hesitate already at the hundreds or tens digits, which flattens the token probabilities and reduces the steepness g. The D cases are similar to the A cases. Their high-order digits are nearly as decisive as those of the A cases, and the steepness g is clearly higher than that of the B and C cases. However, the agent is confidently wrong. The scale-profile score cannot separate case D from case A, and the overconfident second prediction agrees with the edge prediction, which makes case D the main failure mode.

E.3 Failure Mode Analysis

Table 12 analyzes the failure mode. For each wrong edge prediction, we classify the prediction re-grounded on the local view into three outcomes: it *recovers* the target, it *disagrees* with the edge prediction by more than one target radius without recovering, or it *agrees* with the edge prediction within one target radius without recovering. The three outcomes correspond to cases B, C, and D, respectively. The agreeing outcome accounts for 37.9% of the wrong predictions on average, but for 81.0% of the wrong predictions kept on the device. On ScreenSpot-Pro, the benchmark is hard and the calibrated decision is conservative, so almost nothing is kept. On UI-I2E-Bench and MMBench-GUI, neither score separates these predictions, as illustrated in Appendices E.1 and E.2, so the wrong predictions that SCOPE keeps are mostly of this kind. The agent is overconfident, and an overconfident wrong prediction looks like a correct one to both scores. Catching such errors

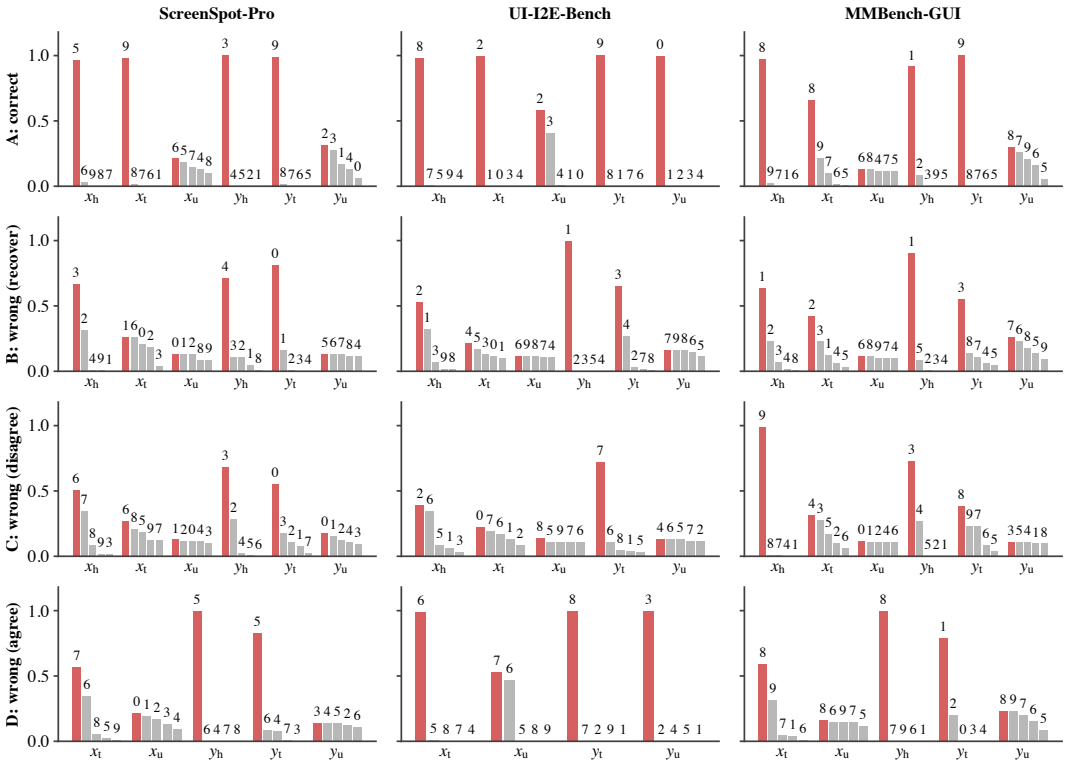


Figure 8: **Digit distributions of the twelve cases.** For every coordinate digit we show the five most probable digit tokens, with the decoded digit in red. Subscripts h, t, and u denote the hundreds, tens, and units digits of the x and y coordinates.

Table 12: **Second predictions of wrong edge predictions.** Wrong is the error rate of the edge agent. Recover, Disagree, and Agree are the fractions of wrong predictions with each outcome of the second prediction. $\Pr(\delta = 0 \mid \text{Agree})$ is the keep probability of the Agree cases over random splits at $\alpha = 0.1$, and Share is their share among all kept wrong predictions.

Agent	Benchmark	n	Wrong	Recover	Disagree	Agree	$\Pr(\delta = 0 \mid \text{Agree})$	Share
InfiGUI	ScreenSpot-Pro	1581	0.555	0.211	0.555	0.234	0.000	–
Holo	ScreenSpot-Pro	1581	0.491	0.248	0.447	0.305	0.000	–
Qwen	ScreenSpot-Pro	1581	0.427	0.233	0.499	0.268	0.006	0.606
InfiGUI	UI-I2E-Bench	1477	0.275	0.259	0.229	0.512	0.579	0.937
Holo	UI-I2E-Bench	1477	0.261	0.236	0.275	0.488	0.680	0.885
Qwen	UI-I2E-Bench	1476	0.254	0.211	0.437	0.352	0.495	0.715
InfiGUI	MMBench-GUI	3594	0.267	0.237	0.324	0.439	0.743	0.984
Holo	MMBench-GUI	3594	0.226	0.258	0.301	0.441	0.812	0.892
Qwen	MMBench-GUI	3594	0.188	0.269	0.364	0.368	0.889	0.648
Mean	–	–	0.327	0.240	0.381	0.379	0.467	0.810

needs an opinion that does not share the agent's bias, such as a lightweight verifier or a second agent. Any such signal can enter the calibration as one more term of the uncertainty scale, so the coverage guarantee carries over.