SPATIALMIND: TEST-TIME SPATIAL POINTING VIA OBSERVATION DEPENDENCY GRAPH

Anonymous authors
Paper under double-blind review

ABSTRACT

This paper presents a test-time framework for spatial pointing with frozen vision-language models. The presented method features a set of zoom procedures, which we call the registry, and treats their outputs as dependent observations. This casts test-time pointing as selection and acquisition. For selection, we propose SpatialMind (SM), a relational answer model that estimates the posterior of correctness of each observation. SM draws on the frozen model's prefill states and messages over dependency relations, starting from a boosted-tree prior. For acquisition, we propose Adaptive SpatialMind (ASM), a practical extension of SM for reduced call budgets. ASM keeps SM unchanged and adds a cost-conditioned policy that decides what to acquire next and when to stop. Our analysis shows that SM learns to update its prior by Bayes' rule in log-odds over dependent observations, and that ASM includes full acquisition and every fixed schedule as special cases. The proposed framework is applied to three frozen backbones on five public pointing benchmarks. The experiments demonstrate the accuracy of SM and the cost efficiency of ASM against both training-free and learned test-time methods.

1 INTRODUCTION

Predicting an image point that satisfies a spatial instruction is known as spatial pointing (Yuan et al., 2025; Zhou et al., 2025; Deitke et al., 2025). Spatial pointing plays an important role in robot manipulation, since a downstream controller takes the predicted point as its target (Nasiriany et al., 2024; Song et al., 2025). A point outside the admissible region therefore makes the action fail. Benchmarks cover referring expressions, spatial relations and free-space placement (Lu et al., 2023; Zhou et al., 2025; Song et al., 2025; Yuan et al., 2025; Cheng et al., 2025).

Since retraining a pointing model is expensive, more computation is often spent at test time to raise accuracy. Self-consistency takes a majority vote over samples (Wang et al., 2023), an adaptive variant stops once the vote is settled (Aggarwal et al., 2023), and GUI-RC votes by region overlap (Du et al., 2026). Learned verifiers replace the vote (Cobbe et al., 2021) and allocate the budget by difficulty (Snell et al., 2025). Coverage grows with the draws, while voting and verifiers saturate (Brown et al., 2024). These approaches, however, treat the draws as exchangeable samples of one call, although they share one image and have correlated errors.

A second family re-examines the image. ZoomEye searches a tree of crops (Shen et al., 2025), V* and DyFo guide it with an auxiliary model (Wu & Xie, 2024; Li et al., 2025), ViCrop crops where the model attends (Zhang et al., 2025a), and Chain-of-Focus trains the backbone to zoom (Zhang et al., 2025b). Each crop thus depends on an earlier prediction. Visual prompting marks candidates for the frozen model to choose from (Nasiriany et al., 2024; Yang et al., 2023), as RegionFocus does for the points of its zoomed regions (Luo et al., 2025). Yet these methods answer from the end of the path or leave the choice to the model, and do not learn a selection rule over the observations.

Combining the two families, we run the frozen model on the image and on crops around earlier points with a procedure set, which we call the registry (Figure 1(a)). At similar cost, the registry contains a correct point more often than repeated sampling (Section 3). Its observations are dependent, as each crop is cut around an earlier point, so the crops that follow a wrong point can repeat its error (Figure 2). Accuracy is the product of coverage (some observation is correct) and selection (the rule returns a correct one). The best selection ranks by the posterior of correctness (Section 2). What is missing is an estimate of this posterior for dependent observations and a stopping rule.

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.

Hence, we treat spatial pointing as selection and acquisition over an observation graph whose edges link each call to the crop around its point. Both problems share one answer model. Its log-odds is a boosted-tree prior (Friedman, 2001) plus a correction computed by message passing (Gilmer et al., 2017; Schlichtkrull et al., 2018). Unlike the trees, the correction uses the typed relations between calls and each call's prefill state, the hidden state that the call already computes before decoding. The acquisition policy follows cost-sensitive feature acquisition (Dulac-Arnold et al., 2011; Shim et al., 2018; Janisch et al., 2019), reuses the answer model on every partial history and is trained by approximate policy iteration (Bertsekas & Tsitsiklis, 1996).

In this paper, SpatialMind (SM) and Adaptive SpatialMind (ASM) are proposed for selection and acquisition, respectively. Our contributions are as follows. 1) Firstly, we formulate spatial pointing as selection and acquisition over dependent observations, with a lemma factoring accuracy into coverage and selection. 2) Secondly, we propose SM, which corrects the log-odds of a boosted-tree prior with messages over typed relations. We show that, when errors are inherited down a chain, typed messages can recover the posterior of correctness where relation-agnostic tree features cannot. 3) Thirdly, we propose ASM, which reuses SM's answer model and learns what to acquire and when to stop. It includes SM, the plain call and every fixed schedule as special cases. As the cost weight grows, its best policy trades accuracy for fewer calls along one Pareto front. 4) Lastly, on three frozen backbones, SM is more accurate on average than every baseline, including the registry vote and boosted trees. At the main cost weight, ASM has higher cost-adjusted utility than every baseline with fewer than half of SM's calls. On held-out benchmarks, both still outperform every external baseline. Sections 2 to 5 develop the formulation, SM and ASM, and Section 6 evaluates them.

2 Problem Statement

We consider a frozen pointing vision-language model, called the backbone. A query x pairs an image with a spatial instruction, and an answer is a point in the normalised frame $[0,1]^2$. The instruction defines an admissible region $M_x \subset [0,1]^2$, and an answer inside M_x is correct. The backbone is queried through a procedure set $\mathcal{V} = \{v_0, \dots, v_{n-1}\}$ designed for the application, which we call the registry. Each procedure runs the backbone on one view of the image and returns a point z_i with $o_i = 1$, or no point with $o_i = 0$. The procedure v_0 is the plain call on the whole image and is always available. A procedure is a root, like v_0, whose view depends on no earlier output, or a child, whose view is a crop around its parent's point. An edge $(v_j, v_i) \in \mathcal{E}$ links the parent v_j to its child v_i, so v_i is run only if v_j has returned a point. The edges $\mathcal{E}$ form a forest. Procedure v_i costs $c_i \geq 1$ backbone calls, which may include auxiliary calls that prepare its view. The available procedures $\mathcal{F} \subseteq \mathcal{V}$ may depend on the query. The correctness of v_i is $y_i = \mathbf{1}[z_i \in M_x]$. The observed set $\mathcal{O}(H)$ consists of the procedures of a history H that returned a point.

The regions M_x are known only in training. At test time a rule uses only the returned points, the prefill states, the available set $\mathcal{F}$ with the cost of each procedure, and a query embedding from frozen encoders. The first problem is selection. Every available procedure is run unless its parent returned no point, giving the full history H_{full}. A rule returns one observed point as follows:

$$\hat{z} = z_{i^*(H)}, \qquad i^*(H) = \operatorname*{arg\,max}_{i \in \mathcal{O}(H)} s_i(H), \tag{1}$$

where $s_i(H)$ is the score of observation i given H. The objective is to maximise $\mathbb{E}[y_{i^*(H_{\text{full}})}]$.

The second problem is acquisition. Observations are acquired one at a time, and the history H_t contains the acquisitions before step t and their outputs. A feasible action in $\mathcal{A}(H_t)$ acquires a new procedure of $\mathcal{F}$ whose parent, if any, has returned a point, or stops after the first acquisition. Stopping at step τ returns $z_{i^*(H_\tau)}$ by Eq. 1. For a cost weight $\lambda \geq 0$, the objective is given as:

$$J_\lambda(\pi) = \mathbb{E}\Big[\, y_{i^*(H_\tau)} - \lambda \sum_{t<\tau} c(\mathrm{act}_t) \Big], \tag{2}$$

where π is the policy that chooses $\mathrm{act}_t \in \mathcal{A}(H_t)$, $c(\mathrm{act}_t)$ is the cost of the procedure acquired at step t, and $y_{i^*(H_\tau)} = 0$ when no point was observed. The weight λ is set by the user and trades accuracy against calls. The mean number of calls is thus a result rather than a budget.

Lemma 1 (Coverage and selection): Let H be a history, that is, the information available to a rule at test time. Let $p_i(H) = \mathbb{P}(y_i = 1 \mid H)$, and let the coverage $\bar{y} = \max_{i \in \mathcal{O}(H)} y_i$ indicate whether

some observation is correct. Every rule that returns $\hat{\imath}(H) \in \mathcal{O}(H)$ satisfies:

$$\mathbb{P}(y_{\hat{\imath}} = 1) = \mathbb{P}(\bar{y} = 1)\,\mathbb{P}(y_{\hat{\imath}} = 1 \mid \bar{y} = 1) \le \mathbb{E}\big[\max_{i \in \mathcal{O}(H)} p_i(H)\big] \le \mathbb{P}(\bar{y} = 1), \tag{3}$$

where the Bayes selector $i^{\mathrm{B}}(H) = \arg\max_{i \in \mathcal{O}(H)} p_i(H)$ attains the first bound. Neither bound falls when more procedures are acquired (Appendix B). Acquisition thus controls coverage, and the answer model controls selection.

To apply rules learned on training queries to new ones, the following assumptions are made:

Assumption 1: The backbone is frozen, every registry call is decoded greedily, and the prefill state of each call is accessible, as with an open-weight model.

Assumption 2: The registry $\mathcal{V}$, its edges $\mathcal{E}$ and the cost rule are the same for training and test queries.

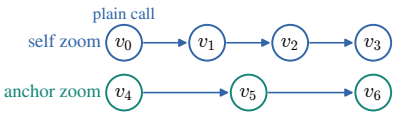
Assumption 3: The output of a procedure depends only on its view, not on the order of acquisition, so an observation does not depend on when or by which policy it is acquired.

3 Framework Overview

3.1 Registry and Diagnosis

The registry used here has $n = 7$ procedures (Table 1). The self-zoom chain $v_0 \to \cdots \to v_3$ starts at the plain call v_0, and each step scales the sides of the preceding window by 0.7 around the parent's point. The anchor-zoom chain $v_4 \to v_5 \to v_6$ starts from a crop around the landmarks that a small text-only language model extracts from the instruction, and then zooms in the same way (Appendix E). Each of the $n_{\mathrm{loc}} \le 3$ landmarks is located by one backbone call, so $c_4 = 1 + n_{\mathrm{loc}}$ and $c_i = 1$ otherwise. The anchor chain is available if and only if $n_{\mathrm{loc}} > 0$. To make the two chains fail differently, self zoom refines coarse to fine around the backbone's own point, and anchor zoom keeps the instruction's reference objects in view.

Table 1: Procedure registry (left, arrows mark crops around the parent's point) and accuracy (%) of each child given a correct or wrong parent (right), on queries where both returned a point.



		RR-8B		RR-2B		Molmo-7B-D	
child	parent	correct	wrong	correct	wrong	correct	wrong
v_1	v_0	93.32	21.23	91.12	24.46	90.84	20.17
v_2	v_1	94.24	18.98	88.07	20.00	90.03	15.53
v_3	v_2	93.85	15.29	35.53	9.22	90.84	13.92
v_5	v_4	93.31	19.07	40.55	8.60	90.35	18.10
v_6	v_5	94.48	11.71	53.50	3.42	91.27	12.37

Table 2 compares the seven registry observations with seven plain-call samples at temperature T on three backbones, RoboRefer-8B-SFT and RoboRefer-2B-SFT (RR-8B, RR-2B) (Zhou et al., 2025) and Molmo-7B-D (Deitke et al., 2025). With slightly more calls, the registry has higher coverage on every backbone, by 7.95 points pooled at $T = 0.3$ and 4.16 at $T = 0.6$. Ten samples still have lower coverage (Appendix C). Voting over the registry improves on the plain call by 3.49 points, but voting over samples does not. We rely on coverage rather than on the correlation ρ in Table 2, since a low ρ can also result from crops that fail under a correct parent (Appendix C).

Table 1 shows that errors are inherited down each chain of this registry. A child is 3.7 to 15.6 times more often correct under a correct parent than under a wrong one. Every step also corrects some errors of its parent and makes new ones, and pooled over backbones only the first zoom step has a net gain (Appendix C). Since a child is cropped around its parent's point, agreement with a child can be weaker evidence than agreement across chains (Proposition 2 in Section 4.5).

The diagnosis thus leads to two requirements. Selection must estimate the posterior of each observation and weigh agreement by its relation, since the registry often contains a correct point that a single call misses. Acquisition must decide per query how far to extend each chain, since the value of a deeper step varies. Section 3.2 addresses both with a shared answer model.

Table 2: Registry against repeated sampling: correctness correlation (ρ), queries with a correct observation (any, %) and accuracy (%) of the plain call or first sample and of the radius-0.05 vote.

	7 registry observations					7 samples, T=0.3				7 samples, T=0.6			
backbone	calls	ρ	any	plain	vote	ρ	any	first	vote	ρ	any	first	vote
RR-8B	7.83	0.645	84.19	71.09	74.72	0.869	76.94	70.79	71.16	0.773	80.07	69.56	71.34
RR-2B	7.95	0.360	80.32	63.04	65.56	0.826	71.77	61.99	62.05	0.740	74.97	59.53	61.56
Molmo-7B-D	7.96	0.658	79.64	63.10	67.40	0.821	71.59	62.48	63.22	0.726	76.63	60.95	63.65

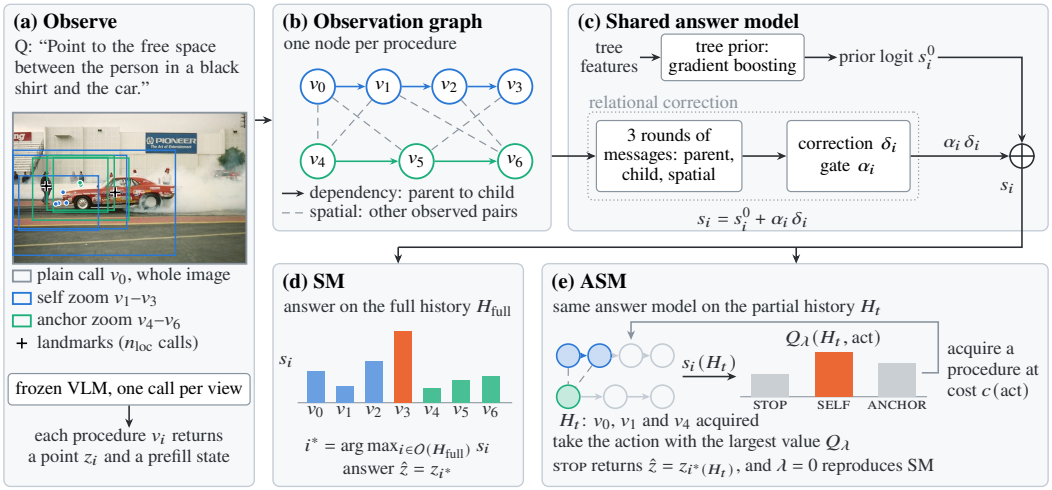


Figure 1: Overview. (a) Each procedure runs the frozen model on one view and returns a point and a prefill state. (b) Solid edges link a parent to the crop around its point, and dashed edges join other observed points. (c) The answer model adds a relational correction to a boosted-tree prior logit. (d) SM returns the top-scored point. (e) ASM uses the same model to stop or extend a chain.

3.2 Overall Architecture

Figure 1 shows the overall framework. (a) Each procedure returns a point and a prefill state. (b) The observations form a graph with dependency and spatial edges. (c) The answer model assigns each observation i a score s_i, its estimated log-odds of being correct. The score is the logit s_i^0 of a boosted-tree prior plus a relational correction δ_i, scaled by a gate $\alpha_i \in (0, 1)$, so $s_i = s_i^0 + \alpha_i \delta_i$. The most accurate selection ranks the observations by their posterior of correctness (Lemma 1 in Section 2). The correction captures evidence that the tree features miss, such as whether an agreeing point was cropped around the point it agrees with. Adding it to the prior logit is Bayes' rule in log-odds (Proposition 1 in Section 4.5). Figure 2 shows one query on which the correction changes the returned point. (d) SM scores the full history. (e) ASM scores the partial history with the same model and stops or extends a chain. In the training phase, the answer model is fitted to the labels from M_x, and the policy to its out-of-fold scores. At test time, the top-scored point is returned.

4 SpatialMind

In this section, following hybrids of boosting and neural networks (Ivanov & Prokhorenkova, 2021), we define the score s_i of Eq. 1, an estimate of the posterior $p_i(H)$ of Lemma 1 (Section 2). Section 4.1 scores each observation with a boosted-tree prior. Section 4.2 builds the observation graph from prefill states and point geometry. Section 4.3 corrects the prior logit with messages over typed relations. The selection rule and the loss follow in Section 4.4. Section 4.5 shows why the correction targets the posterior and why three rounds of typed messages suffice.

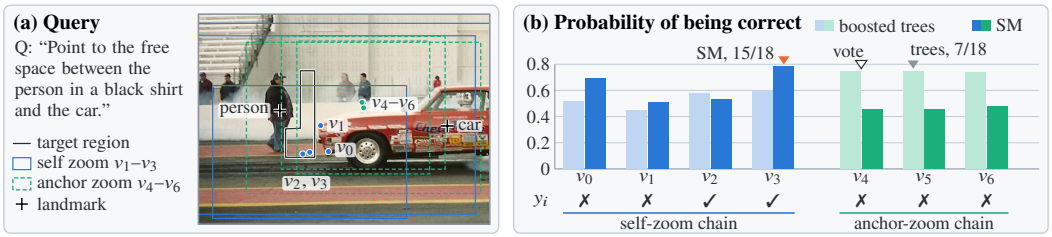


Figure 2: One query on which SM avoids an error inherited down a chain (RR-8B, PointArena, as in Figure 1(a)). (a) The anchor chain starts from a window around the two located landmarks. v_4 falls on the people behind the car, and v_5 and v_6, each cropped around the previous point, fall next to it. v_2 and v_3 reach the free space, the target region. (b) Mean probability of being correct over the 18 folds that test this query (Section 6.1). SM lowers the agreeing points v_4 to v_6, which boosted trees rate highest. Triangles mark the vote's pick and the most frequent pick of boosted trees and of SM, with its count. SM returns a correct point in 15 of the 18 folds, boosted trees in none.

4.1 Tree Prior

The tree prior gives each observation i the prior logit $s_i^0(H) = \text{logit}\, \mathcal{G}(\mathbf{t}_i(H))$, where $\mathcal{G}$ is a boosted-tree classifier (Pedregosa et al., 2011) and $\mathbf{t}_i(H)$ denotes hand-crafted features. They encode the procedure, which procedures returned a point, and each procedure's success rate on the training split and on the training queries nearest in image and instruction embedding. They also include the point z_i and six counts and distances of the other observed points around it (Appendix D.1). These agreement features ignore whether an agreeing point comes from the parent, a child or another chain. Recomputed from the current history, the features define the prior on every partial history.

4.2 Observation Graph

Each procedure v_i is one node of the graph. The initial state of a node is computed from its prefill state, its tree features and its prior logit s_i^0. The prefill state comes from the call's own prefill pass before any answer token, so it requires no additional call. Each ordered pair of nodes is also described by its geometry, the displacement $z_j - z_i$ between their points and their distance $d_{ij} = \|z_j - z_i\|$ at five scales (Eqs. 7 and 8 in Appendix D.1).

4.3 Relational Correction

We distinguish three relations among the observed nodes $\mathcal{O} = \mathcal{O}(H)$: the parent of a node, its children and, as the spatial relation, every other observed node. The parent and child relations follow the edges $\mathcal{E}$, so they model the dependence of a crop on its parent's point. Each relation has its own weights, so agreement with a child can count for less than agreement with another chain (Proposition 2 in Section 4.5). In each of three rounds, which share weights, a node averages the messages it receives over each relation and updates its state. Each message is scaled by a learned weight, so a node can down-weight an uninformative neighbour (Eqs. 9 and 10 in Appendix D.1). After the last round, a small network maps the node's states from all rounds and their mean over $\mathcal{O}$ to a correction δ_i and a gate $\alpha_i \in (0, 1)$ that controls how much of the correction is applied:

$$s_i = s_i^0 + \alpha_i\, \delta_i. \tag{4}$$

The layer that outputs δ_i starts at zero, so training starts at the prior (Eq. 11 in Appendix D.1).

4.4 Selection and Training

SM returns $\hat{z} = z_{i^*(H_{\text{full}})}$, and we train its answer model with two terms (Eq. 12 in Appendix D.1). A pairwise logistic loss (Burges et al., 2005) ranks each correct observed point above each wrong one. A cross-entropy term with weight $1/4$ fits $\sigma(s_i)$ to the label y_i, where σ is the logistic function. The loss is taken on the full history with weight 0.9 and on one partial history per query with weight 0.1, drawn uniformly among those the query can reach. The same model is thus trained on the partial

histories ASM visits. Each training query's prior features and logits come from trees fitted without its image, that is, out of fold. Training uses AdamW for 10 epochs.

4.5 Analysis

In short, the correction adds what the trees miss, and the relations weigh agreement by its source. Proposition 1 shows that the optimal correction is the log-likelihood-ratio term of Bayes' rule in log-odds. SM approaches the Bayes selector of Lemma 1 (Section 2) as its scores approach the posterior $p_i(H)$, which the cross-entropy term of the loss targets. By Proposition 2, a wrong child tends to agree with a wrong parent, so the posterior can depend on a relation that tree features ignore. Three rounds of messages cover the longest chain. Proofs are in Appendix B.

Proposition 1 (Correction as Bayes' rule in log-odds): For an observed i, let E_i be the correction's inputs beyond $\mathbf{t}_i$ (prefill states, other observations), and let $q_i = \mathbb{P}(y_i = 1 \mid \mathbf{t}_i, E_i) \in (0, 1)$. (a) Among corrections $\Delta(\mathbf{t}_i, E_i)$, the expected cross-entropy of $\sigma(s_i^0 + \Delta)$ has the unique minimiser $\operatorname{logit} q_i - s_i^0$ (Gneiting & Raftery, 2007). If the prior is calibrated, $\sigma(s_i^0) = \mathbb{P}(y_i = 1 \mid \mathbf{t}_i)$, this minimiser is the log-likelihood ratio of E_i between $y_i = 1$ and $y_i = 0$ given $\mathbf{t}_i$. (b) With a zero correction, as at initialisation, $s_i = s_i^0$, so SM's class contains its own tree prior used as a selector. (c) The accuracy of Eq. 1 is below that of the Bayes selector by at most the square root of twice the excess cross-entropy of the scores, summed over $\mathcal{O}(H)$, which is 0 if $\sigma(s_i) = p_i(H)$ on $\mathcal{O}(H)$.

Proposition 2 (Inherited agreement and chain depth): Let $\Omega_j \subseteq [0, 1]^2$, of area $|\Omega_j|$, be the view of observation j: the image for v_0, for another root a region that depends on no output, and for a child j of i a crop around z_i. Assume Model M: given the query, its view, its label and the outputs of procedures that do not descend from it, a wrong point is uniform over its view outside M_x. (a) Let j be a child of i, and let the disc of radius $\bar{d}$ around z_i lie in $\Omega_j \setminus M_x$. Then $\mathbb{P}(d_{ij} \le \bar{d} \mid x, y_j = 0, z_i) \ge \pi\bar{d}^2/|\Omega_j|$. A wrong point uniform over the image outside M_x falls in this disc with probability at most $\pi\bar{d}^2/(1 - |M_x|)$. (b) Let j be a child of a root $v_i \ne v_0$, with $|\Omega_j| < 1$. There are histories H, H' with $\mathbf{t}_i(H) = \mathbf{t}_i(H')$, in which the point that agrees with z_i is its child's point z_j in H and z_0 in H', and laws satisfying M under which $\mathbb{P}(y_i = 1 \mid H) < \mathbb{P}(y_i = 1 \mid H')$. (c) Let the pairs of label and evidence along a chain $u_1 \to \cdots \to u_L$ form a Markov chain. Then each posterior given the chain's evidence is the normalised product of a forward message from the parent and a backward one from the child (Pearl, 1988; Kschischang et al., 2001). Synchronous rounds of these messages make all posteriors exact after $L-1$ rounds, and end nodes need all $L-1$ in general.

From (a), a wrong child falls within $\bar{d}$ of a wrong parent at least $(1 - |M_x|)/|\Omega_j|$ times as often as a wrong whole-image point (Figure 2). For v_1 to v_3, the crop of v_j keeps 0.7^j of each image side, so $1/|\Omega_j| \ge 2.0$, 4.2 and 8.5 unless the 224-pixel minimum crop size applies (Appendix B). By (b), the tree features encode that a point agrees with z_i but not whether it is the child's point. The child relation takes $z_j - z_i$ as input (Section 4.2), so it detects whether the agreeing point is the child's and separates H and H'. Finally, by (c), the longest chain $v_0 \to \cdots \to v_3$ has $L = 4$, so three rounds of parent and child messages suffice, and the spatial relation joins all pairs in one round.

5 Adaptive SpatialMind

In this section, we describe Adaptive SpatialMind (ASM), which keeps the answer model of SM and its weights unchanged and adds a policy that decides only what to acquire and when to stop, to maximise the utility of Eq. 2. Section 5.1 defines partial histories and the feasible actions. Section 5.2 values each action by the best fixed schedule, adapted to the history, and fits the policy by policy iteration with exact rollouts. The test-time rule of Section 5.3 equals SM at $\lambda = 0$. Section 5.4 analyses the policy class of ASM and the trade-off traced by λ.

5.1 Partial Histories and Feasible Actions

A history H_t is the set of procedures acquired before step t with their outputs. After each acquisition, the answer model of Section 4 is applied again to $\mathcal{O}(H_t)$, giving $s_i(H_t)$ and $i^*(H_t)$. Only acquired procedures enter the features and messages, so no decision depends on a later output.

An action acquires a feasible procedure (Section 2) or answers with STOP after the first acquisition. Acquisition may start at any root, and procedure v_i costs c_i. Since parents come first, each chain is acquired from its root down, and an acquisition extends one chain. In our registry, an acquisition state counts the acquisitions per chain, up to four on self zoom and three on anchor zoom, which gives twenty states. The actions SELF and ANCHOR extend the two chains (Appendix D.2).

5.2 Action Value and Policy Iteration

The value $Q_\lambda(H_t, \mathrm{act})$ estimates the utility of Eq. 2 from H_t on when act is taken first. It has three parts (Eq. 15 in Appendix D.2). The first is the mean utility of the best fixed schedule that starts with the action, over training queries with the same state and feasible actions. A fixed schedule runs each chain to a preset depth or until a call returns no point. The second is an offset per action, fitted on the same queries. The third is a small network whose inputs are a summary of the current history, the action and λ. It starts at zero, is centred over the feasible actions and is shared across all λ.

We fit the policy by approximate policy iteration (Bertsekas & Tsitsiklis, 1996; Lagoudakis & Parr, 2003) on out-of-fold answer scores at every partial history of each training query. The first policy π_0 takes the action whose best fixed schedule has the largest value. In round $k \in \{1, 2\}$, every feasible action is taken from every state, and π_{k-1} is simulated on the query's precomputed observations (Assumption 3 in Section 2). This gives an exact rollout return without bootstrapping:

$$G_\lambda(H_t, \mathrm{act}) = y_{i^*(H_\tau)} - \lambda \textstyle\sum_{t'=t}^{\tau-1} c(\mathrm{act}_{t'}), \tag{5}$$

where $\mathrm{act}_t = \mathrm{act}$, π_{k-1} stops at H_τ, and i^* uses the out-of-fold scores. We refit the network and then the offsets to these returns, and π_k is greedy in the new value (Appendix D.2).

5.3 Test-Time Rule

At test time the history starts empty, and each action is chosen as follows:

$$\mathrm{act}_t = \begin{cases} \mathrm{argmax}_{\mathrm{act} \in \mathcal{A}(H_t)} \, Q_\lambda(H_t, \mathrm{act}), & \lambda > 0, \\ \text{the first feasible acquisition in a fixed order, else STOP}, & \lambda = 0, \end{cases} \tag{6}$$

where the order is SELF, then ANCHOR, for the registry of Section 3. At $\lambda = 0$, ASM thus acquires every available procedure and returns the same answer as SM (Proposition 3 in Section 5.4).

5.4 Analysis

The best policy of ASM's class has at least the utility of SM, the plain call and every fixed schedule, and λ selects a point on a single trade-off (Proposition 3, proved in Appendix B).

Proposition 3 (Containment and monotone trade-off): Let Π be the policies that choose $\mathrm{act}_t \in \mathcal{A}(H_t)$ from H_t and answer on STOP by Eq. 1 with the shared answer model. Write Eq. 2 as $J_\lambda(\pi) = \mathrm{acc}(\pi) - \lambda \, \mathrm{calls}(\pi)$, and let $\pi^\star_\lambda$ attain its maximum $J^\star_\lambda$ over Π. (a) $s_i(H_t)$ and $Q_\lambda(H_t, \cdot)$ depend only on H_t, so ASM lies in Π. (b) Π contains SM (full acquisition), the plain call (acquire v_0, then STOP) and every fixed schedule, so $J^\star_\lambda$ is at least their utility. (c) For $0 \le \lambda < \lambda'$, $\mathrm{calls}(\pi^\star_\lambda) \ge \mathrm{calls}(\pi^\star_{\lambda'})$ and $\mathrm{acc}(\pi^\star_\lambda) \ge \mathrm{acc}(\pi^\star_{\lambda'})$. For $\lambda > 0$, $\pi^\star_\lambda$ is Pareto-optimal in accuracy and calls over Π. The optimum $J^\star_\lambda$ is convex and strictly decreasing in λ, with subgradient $-\mathrm{calls}(\pi^\star_\lambda)$.

By Proposition 4 (Appendix B), a policy greedy in values within ε of the best fixed schedule's is at most 2ε per remaining decision below it. We thus fit the values to exact rollout returns (Eq. 5).

6 Experiment

6.1 Setup

The evaluation set combines five public benchmarks: RoboRefIt (Lu et al., 2023), RefSpatial (Zhou et al., 2025), PointArena (Cheng et al., 2025), RoboSpatial (Song et al., 2025) and Where2Place (Yuan et al., 2025). It has 1,626 queries on 1,139 images (Appendix F). On each backbone, the output of each procedure is computed once per query. By Assumption 3 (Section 2), it equals that of

a new call, so ASM is evaluated as if acquiring online, and only the calls it makes are counted. On RefSpatial, the anchor chain is used only for instructions that the benchmark annotates as multi-step, so it is disabled on 57 queries. Every margin in Table 17 remains positive without them.

No image appears on both sides of a split. For each of three random image-grouped splits, the images are divided into five, two and five folds for training fractions of 20, 50 and 80%. This gives 108 folds over the three backbones. Every component, including the prior and the policy, uses only the training split. All folds share the hyperparameters (Appendix D). Each benchmark is also held out in turn, with SM and ASM fitted on the other four (15 folds on 926 to 1,504 queries, Table 15).

6.2 Analysis Method

To evaluate our framework, we use the plain call v_0 as the baseline. We also compare against external training-free methods in their published form. RegionFocus (Luo et al., 2025) and ZoomEye (Shen et al., 2025) search over image crops as in their public code, adapted to pointing. SC-10 (Wang et al., 2023), AC-10 (Aggarwal et al., 2023) and GUI-RC-10 (Du et al., 2026) aggregate ten samples of the plain call at $T = 0.3$ and 0.6. Over our registry, we use the registry vote of Table 2 as a training-free rule. Boosted trees, a gradient-boosted classifier with default settings on the observations' own features, are used as a basic learned selector on the same observations. SM, ASM and boosted trees are fitted on each fold's labelled training split, 325 to 1,301 queries, while the other methods use no labels. Related work is reviewed in Appendix A, and baseline details are in Appendix E.

The evaluation metrics are as follows: 1) accuracy, the share of returned points inside M_x, 2) the mean backbone calls per query, and 3) the utility J_λ of Eq. 2 at the main cost weight $\lambda = 0.01$, where one call costs one accuracy point. Each metric is averaged over nine conditions, three backbones at three training fractions. Margins are per-query differences with 95% paired image-bootstrap intervals and sign-flip p-values adjusted for multiplicity (Appendix H).

6.3 Result

Table 3 shows that SM is the most accurate method, at 70.71%, 4.97 points above the plain call, and that every margin is significant after adjustment. RegionFocus, the strongest external baseline, is

Table 3: Accuracy, mean calls and pooled margins (nine-condition average): accuracy points for SM, utility margins $100\,\Delta J_{0.01}$ for ASM (λ=0.01). Every margin has adjusted $p = 0.002$, the smallest possible over 20 tests (Holm, or Bonferroni where marked † and ‡, Appendix H).

method	calls	accuracy (%)	SM − method accuracy points [95% CI]	$\text{ASM}_{\lambda=0.01}$ − method utility margin [95% CI]
external training-free baselines				
plain call	1.00	65.74	+4.97 [4.00, 5.93]	+1.37 [0.67, 2.09]
RegionFocus	4.47	67.16	+3.55 [2.61, 4.49]	+3.42 [2.65, 4.23]
ZoomEye	19.23	63.53	+7.18 [6.11, 8.26]	+21.82 [19.65, 24.47]
SC-10 (T=0.3)	10.00	65.42	+5.29 [4.28, 6.32]	+10.70 [9.89, 11.52]
SC-10 (T=0.6)	10.00	65.01	+5.70 [4.66, 6.78]	+11.11 [10.22, 12.02]
AC-10 (T=0.3)	6.60	65.44	+5.27 [4.24, 6.32]	+7.28 [6.43, 8.13]†
AC-10 (T=0.6)	7.63	64.99	+5.72 [4.66, 6.82]	+8.76 [7.85, 9.69]†
GUI-RC-10 (T=0.3)	10.00	65.81	+4.90 [3.92, 5.91]	+10.31 [9.53, 11.10]
GUI-RC-10 (T=0.6)	10.00	65.56	+5.15 [4.09, 6.23]	+10.56 [9.67, 11.46]
training-free rule over the registry				
registry vote (radius 0.05)	7.91	69.23	+1.48 [0.88, 2.08]‡	+4.80 [4.05, 5.57]‡
fitted on the registry				
boosted trees	7.91	69.08	+1.63 [1.25, 2.02]	+4.95 [4.44, 5.47]
ours, fitted on the registry				
SM	7.91	70.71	–	–
ASM, λ=0.003	4.99	70.20	–	–
ASM, λ=0.01	2.93	69.05	–	–

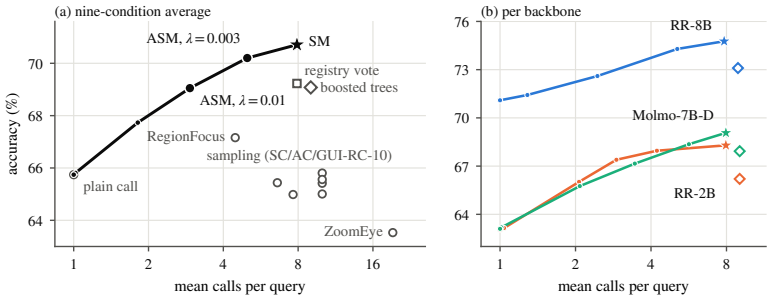


Figure 3: Accuracy against mean calls (log axis), where up and to the left is better. (a) Nine-condition average of every method, with the ASM curve over λ from SM ($\lambda = 0$) to the plain call. (b) Per backbone: ASM, SM (star) and boosted trees (diamond). Boosted trees make the same calls as SM, so their diamonds are shifted right in both panels.

3.55 points lower. In terms of Lemma 1 (Section 2), the registry raises coverage to 81.39%, and the rules that use all its observations differ only in selection. With the same observations and calls, SM is 1.48 points $[0.88, 2.08]$ above the registry vote and 1.63 points $[1.25, 2.02]$ above boosted trees. SM closes 31.74% of the gap between the plain call and the coverage, the largest share on every backbone, against 22.28% for the vote and 21.31% for boosted trees (Table 16).

Figure 3 shows that ASM traces the accuracy–call trade-off with SM's answer model at every λ. On average, each training-free baseline but the plain call has an ASM point with fewer calls and higher accuracy. At $\lambda = 0.003$, ASM reaches 70.20% at 4.99 calls, above the registry vote and 1.12 points $[0.74, 1.52]$ above boosted trees with 37% fewer calls. At $\lambda = 0.01$, it reaches 69.05% at 2.93 calls, 62.9% fewer than SM. It has significantly higher utility $J_{0.01}$ than every baseline, by 1.37 points over the plain call and 3.42 over RegionFocus. As Proposition 3 (Section 5.4) proves for exact maximisers, calls and accuracy do not increase with λ (Appendix G). ASM's policy uses the answer model's scores to spend calls where the plain call is unreliable. At $\lambda = 0.01$, the plain call is correct on 71.0% of queries where ASM stops after it and on 49.3% where it spends five or more.

SM exceeds every external baseline and boosted trees in each of the nine conditions, and ASM at $\lambda = 0.01$ is more accurate than the plain call in each with fewer than half of SM's calls (Appendix G). With 325 or 326 labelled queries (20%), SM still outperforms RegionFocus by 2.08 to 4.59 points. Over the registry vote, SM gains 2.73 and 1.66 points on RR-2B and Molmo-7B-D and is on par on RR-8B ($+0.04$). SM is above every training-free baseline and boosted trees on each benchmark (Table 14). On held-out benchmarks (Table 15), SM (70.60%) remains significantly above every external baseline in accuracy, and so does ASM at $\lambda = 0.01$ (69.29% at 3.49 calls) in utility.

7 Conclusion and Future Work

In this paper, test-time spatial pointing with a frozen model was investigated as selection and acquisition over dependent observations. Accuracy was factored into coverage and selection, SM was derived as a log-odds correction of a boosted-tree prior over typed relations, and ASM reused its answer model. The superior results on three backbones show that SM is more accurate than every baseline on average, and that ASM reaches a higher cost-adjusted utility than every baseline at the main cost weight with fewer than half of SM's calls.

Two directions remain for future work. First, SM selects among a registry's points, whose coverage is the first factor of Lemma 1 (Section 2). The relational states could also propose the next procedure, such as a new crop or prompt, so that ASM raises coverage. Second, SM recovers accuracy that the plain call misses (Section 6.3). This gain can be transferred to the backbone without new annotated data. The points SM selects on unlabelled queries, weighted by their posterior, can serve as fine-tuning targets, and the improved backbone yields a new registry.

AI USE STATEMENT

The authors developed the core research ideas, the problem formulation and the design of SM and ASM. Generative AI tools assisted with implementing and analysing the experiments, drafting the proofs, and drafting and editing the manuscript. The authors reviewed all AI-assisted work and take full responsibility for the final code, experiments, analyses and manuscript.

ETHICS STATEMENT

This work uses only public datasets and publicly released models under their licences, and it involves no human subjects. The method outputs an image point and leaves any physical action to a downstream controller.

REPRODUCIBILITY STATEMENT

The details needed to reproduce the results are given in the appendix. The procedure registry, the prompts and decoding settings, the extraction of the landmark phrases, the counting of calls and the implementations of the baselines are in Appendix E. The evaluation set, including the selection of the PointArena items, is in Appendix F. The answer model and the acquisition policy of SM and ASM, their hyperparameters and the compute used are in Appendix D. The proofs of Lemma 1 and the propositions are in Appendix B, and the statistical protocol is in Appendix H. All datasets and backbones are public. We will release, as supplementary material, the anonymised code, the query identifiers (including the 264 PointArena items), the extracted landmark phrases, the folds, the precomputed procedure outputs from which every table is computed and the 108 per-fold and 15 held-out-benchmark checkpoints.

REFERENCES

Pranjal Aggarwal, Aman Madaan, Yiming Yang, and Mausam. Let's sample step by step: Adaptive-consistency for efficient reasoning and coding with LLMs. In *Conference on Empirical Methods in Natural Language Processing*, pp. 12375–12396, 2023.

Dimitri P. Bertsekas and John N. Tsitsiklis. *Neuro-Dynamic Programming*. Athena Scientific, 1996.

Bradley Brown, Jordan Juravsky, Ryan Ehrlich, Ronald Clark, Quoc V. Le, Christopher Ré, and Azalia Mirhoseini. Large language monkeys: Scaling inference compute with repeated sampling. *arXiv preprint arXiv:2407.21787*, 2024.

Chris Burges, Tal Shaked, Erin Renshaw, Ari Lazier, Matt Deeds, Nicole Hamilton, and Greg Hullender. Learning to rank using gradient descent. In *International Conference on Machine Learning*, pp. 89–96, 2005.

Hyeong Soo Chang, Robert Givan, and Edwin K. P. Chong. Parallel rollout for online solution of partially observable Markov decision processes. *Discrete Event Dynamic Systems*, 14(3):309–341, 2004.

Lingjiao Chen, Matei Zaharia, and James Zou. FrugalGPT: How to use large language models while reducing cost and improving performance. *Transactions on Machine Learning Research*, 2024.

Long Cheng, Jiafei Duan, Yi Ru Wang, Haoquan Fang, Boyang Li, Yushan Huang, Elvis Wang, Ainaz Eftekhar, Jason Lee, Wentao Yuan, Rose Hendrix, Noah A. Smith, Fei Xia, Dieter Fox, and Ranjay Krishna. PointArena: Probing multimodal grounding through language-guided pointing. *arXiv preprint arXiv:2505.09990*, 2025.

Robert T. Clemen and Robert L. Winkler. Limits for the precision and value of information from dependent sources. *Operations Research*, 33(2):427–442, 1985.

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

A. P. Dawid and A. M. Skene. Maximum likelihood estimation of observer error-rates using the EM algorithm. *Journal of the Royal Statistical Society: Series C (Applied Statistics)*, 28(1):20–28, 1979.

Matt Deitke, Christopher Clark, Sangho Lee, Rohun Tripathi, Yue Yang, Jae Sung Park, Mohammadreza Salehi, Niklas Muennighoff, Kyle Lo, Luca Soldaini, Jiasen Lu, Taira Anderson, Erin Bransom, Kiana Ehsani, Huong Ngo, YenSung Chen, Ajay Patel, Mark Yatskar, Chris Callison-Burch, Andrew Head, Rose Hendrix, Favyen Bastani, Eli VanderBilt, Nathan Lambert, Yvonne Chou, Arnavi Chheda, Jenna Sparks, Sam Skjonsberg, Michael Schmitz, Aaron Sarnat, Byron Bischoff, Pete Walsh, Chris Newell, Piper Wolters, Tanmay Gupta, Kuo-Hao Zeng, Jon Borchardt, Dirk Groeneveld, Crystal Nam, Sophie Lebrecht, Caitlin Wittlif, Carissa Schoenick, Oscar Michel, Ranjay Krishna, Luca Weihs, Noah A. Smith, Hannaneh Hajishirzi, Ross Girshick, Ali Farhadi, and Aniruddha Kembhavi. Molmo and PixMo: Open weights and open data for state-of-the-art vision-language models. In *IEEE/CVF Conference on Computer Vision and Pattern Recognition*, pp. 91–104, 2025.

Yong Du, Yuchen Yan, Fei Tang, Zhengxi Lu, Chang Zong, Weiming Lu, Shengpei Jiang, and Yongliang Shen. Test-time reinforcement learning for GUI grounding via region consistency. *Proceedings of the AAAI Conference on Artificial Intelligence*, 40(36):30593–30601, 2026.

Gabriel Dulac-Arnold, Ludovic Denoyer, Philippe Preux, and Patrick Gallinari. Datum-wise classification: A sequential approach to sparsity. In *European Conference on Machine Learning and Knowledge Discovery in Databases*, pp. 375–390, 2011.

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

Jerome H. Friedman. Greedy function approximation: A gradient boosting machine. *The Annals of Statistics*, 29(5):1189–1232, 2001.

Giacomo Frisoni, Lorenzo Molfetta, Mattia Buzzoni, and Gianluca Moro. Graph-of-Mark: Promote spatial reasoning in multimodal language models with graph-based visual prompting. *Proceedings of the AAAI Conference on Artificial Intelligence*, 40(36):30726–30734, 2026.

Justin Gilmer, Samuel S. Schoenholz, Patrick F. Riley, Oriol Vinyals, and George E. Dahl. Neural message passing for quantum chemistry. In *International Conference on Machine Learning*, volume 70 of *Proceedings of Machine Learning Research*, pp. 1263–1272, 2017.

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

Tanmay Gupta and Aniruddha Kembhavi. Visual programming: Compositional visual reasoning without training. In *IEEE/CVF Conference on Computer Vision and Pattern Recognition*, pp. 14953–14962, 2023.

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

Sergei Ivanov and Liudmila Prokhorenkova. Boost then convolve: Gradient boosting meets graph neural networks. In *International Conference on Learning Representations*, 2021.

Jaromír Janisch, Tomáš Pevný, and Viliam Lisý. Classification with costly features using deep reinforcement learning. *Proceedings of the AAAI Conference on Artificial Intelligence*, 33(01): 3959–3966, 2019.

Yusuf Kalayci, Vinod Raman, and Shaddin Dughmi. Optimal stopping vs Best-of-N for inference time optimization. *arXiv preprint arXiv:2510.01394*, 2025.

Frank R. Kschischang, Brendan J. Frey, and Hans-Andrea Loeliger. Factor graphs and the sum-product algorithm. *IEEE Transactions on Information Theory*, 47(2):498–519, 2001.

Ludmila I. Kuncheva and Christopher J. Whitaker. Measures of diversity in classifier ensembles and their relationship with the ensemble accuracy. *Machine Learning*, 51(2):181–207, 2003.

Michail G. Lagoudakis and Ronald Parr. Reinforcement learning as classification: Leveraging modern classifiers. In *International Conference on Machine Learning*, pp. 424–431, 2003.

Geng Li, Jinglin Xu, Yunzhen Zhao, and Yuxin Peng. DyFo: A training-free dynamic focus visual search for enhancing LMMs in fine-grained visual understanding. In *IEEE/CVF Conference on Computer Vision and Pattern Recognition*, pp. 9098–9108, 2025.

Yuhao Lu, Yixuan Fan, Beixing Deng, Fangfu Liu, Yali Li, and Shengjin Wang. VL-Grasp: a 6-Dof interactive grasp policy for language-oriented objects in cluttered indoor scenes. In *IEEE/RSJ International Conference on Intelligent Robots and Systems*, pp. 976–983, 2023.

Tiange Luo, Lajanugen Logeswaran, Justin Johnson, and Honglak Lee. Visual test-time scaling for GUI agent grounding. In *IEEE/CVF International Conference on Computer Vision*, pp. 19989–19998, 2025.

Soroush Nasiriany, Fei Xia, Wenhao Yu, Ted Xiao, Jacky Liang, Ishita Dasgupta, Annie Xie, Danny Driess, Ayzaan Wahid, Zhuo Xu, Quan Vuong, Tingnan Zhang, Tsang-Wei Edward Lee, Kuang-Huei Lee, Peng Xu, Sean Kirmani, Yuke Zhu, Andy Zeng, Karol Hausman, Nicolas Heess, Chelsea Finn, Sergey Levine, and Brian Ichter. PIVOT: Iterative visual prompting elicits actionable knowledge for VLMs. In *International Conference on Machine Learning*, volume 235 of *Proceedings of Machine Learning Research*, pp. 37321–37341, 2024.

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 *International Conference on Learning Representations*, 2025.

Judea Pearl. *Probabilistic Reasoning in Intelligent Systems: Networks of Plausible Inference*. Morgan Kaufmann, 1988.

Fabian Pedregosa, Gaël Varoquaux, Alexandre Gramfort, Vincent Michel, Bertrand Thirion, Olivier Grisel, Mathieu Blondel, Peter Prettenhofer, Ron Weiss, Vincent Dubourg, Jake Vanderplas, Alexandre Passos, David Cournapeau, Matthieu Brucher, Matthieu Perrot, and Édouard Duchesnay. Scikit-learn: Machine learning in Python. *Journal of Machine Learning Research*, 12: 2825–2830, 2011.

Nils Reimers and Iryna Gurevych. Sentence-BERT: Sentence embeddings using Siamese BERT-networks. In *Conference on Empirical Methods in Natural Language Processing and the International Joint Conference on Natural Language Processing*, pp. 3982–3992, 2019.

Stuart Russell and Eric Wefald. Principles of metareasoning. *Artificial Intelligence*, 49(1–3):361–395, 1991.

Michael Schlichtkrull, Thomas N. Kipf, Peter Bloem, Rianne van den Berg, Ivan Titov, and Max Welling. Modeling relational data with graph convolutional networks. In *European Semantic Web Conference*, pp. 593–607, 2018.

Haozhan Shen, Kangjia Zhao, Tiancheng Zhao, Ruochen Xu, Zilun Zhang, Mingwei Zhu, and Jianwei Yin. ZoomEye: Enhancing multimodal LLMs with human-like zooming capabilities through tree-based image exploration. In *Conference on Empirical Methods in Natural Language Processing*, pp. 6602–6618, 2025.

Hajin Shim, Sung Ju Hwang, and Eunho Yang. Joint active feature acquisition and classification with variable-size set encoding. In *Advances in Neural Information Processing Systems*, volume 31, 2018.

Satinder P. Singh and Richard C. Yee. An upper bound on the loss from approximate optimal-value functions. *Machine Learning*, 16(3):227–233, 1994.

Charlie Snell, Jaehoon Lee, Kelvin Xu, and Aviral Kumar. Scaling LLM test-time compute optimally can be more effective than scaling parameters for reasoning. In *International Conference on Learning Representations*, 2025.

Chan Hee Song, Valts Blukis, Jonathan Tremblay, Stephen Tyree, Yu Su, and Stan Birchfield. RoboSpatial: Teaching spatial understanding to 2D and 3D vision-language models for robotics. In *IEEE/CVF Conference on Computer Vision and Pattern Recognition*, pp. 15768–15780, 2025.

Dídac Surís, Sachit Menon, and Carl Vondrick. ViperGPT: Visual inference via Python execution for reasoning. In *IEEE/CVF International Conference on Computer Vision*, pp. 11888–11898, 2023.

Fei Tang, Bofan Chen, Zhengxi Lu, Tongbo Chen, Songqin Nong, Tao Jiang, Wenhao Xu, Weiming Lu, Jun Xiao, Yueting Zhuang, and Yongliang Shen. UI-Zoomer: Uncertainty-driven adaptive zoom-in for GUI grounding. *arXiv preprint arXiv:2604.14113*, 2026.

Petar Veličković, Guillem Cucurull, Arantxa Casanova, Adriana Romero, Pietro Liò, and Yoshua Bengio. Graph attention networks. In *International Conference on Learning Representations*, 2018.

Qingni Wang, Yue Fan, and Xin Eric Wang. SafeGround: Know when to trust GUI grounding models via uncertainty calibration. *arXiv preprint arXiv:2602.02419*, 2026.

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 *International Conference on Learning Representations*, 2023.

Hang Wu, Hongkai Chen, Yujun Cai, Chang Liu, Qingwen Ye, Ming-Hsuan Yang, and Yiwei Wang. DiMo-GUI: Advancing test-time scaling in GUI grounding via modality-aware visual reasoning. In *Conference on Empirical Methods in Natural Language Processing*, pp. 26246–26256, 2025.

Penghao Wu and Saining Xie. V*: Guided visual search as a core mechanism in multimodal LLMs. In *IEEE/CVF Conference on Computer Vision and Pattern Recognition*, pp. 13084–13094, 2024.

Keyulu Xu, Chengtao Li, Yonglong Tian, Tomohiro Sonobe, Ken-ichi Kawarabayashi, and Stefanie Jegelka. Representation learning on graphs with jumping knowledge networks. In *International Conference on Machine Learning*, volume 80 of *Proceedings of Machine Learning Research*, pp. 5453–5462, 2018.

An Yang, Baosong Yang, Beichen Zhang, Binyuan Hui, Bo Zheng, Bowen Yu, Chengyuan Li, Dayiheng Liu, Fei Huang, Haoran Wei, Huan Lin, Jian Yang, Jianhong Tu, Jianwei Zhang, Jianxin Yang, Jiaxi Yang, Jingren Zhou, Junyang Lin, Kai Dang, Keming Lu, Keqin Bao, Kexin Yang, Le Yu, Mei Li, Mingfeng Xue, Pei Zhang, Qin Zhu, Rui Men, Runji Lin, Tianhao Li, Tianyi Tang, Tingyu Xia, Xingzhang Ren, Xuancheng Ren, Yang Fan, Yang Su, Yichang Zhang, Yu Wan, Yuqiong Liu, Zeyu Cui, Zhenru Zhang, Zihan Qiu, et al. Qwen2.5 technical report. *arXiv preprint arXiv:2412.15115*, 2024a.

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

Lihe Yang, Bingyi Kang, Zilong Huang, Zhen Zhao, Xiaogang Xu, Jiashi Feng, and Hengshuang Zhao. Depth anything V2. In *Advances in Neural Information Processing Systems*, volume 37, pp. 21875–21911, 2024b.

Wentao Yuan, Jiafei Duan, Valts Blukis, Wilbert Pumacay, Ranjay Krishna, Adithyavairavan Murali, Arsalan Mousavian, and Dieter Fox. RoboPoint: A vision-language model for spatial affordance prediction in robotics. In *Conference on Robot Learning*, volume 270 of *Proceedings of Machine Learning Research*, pp. 4005–4020, 2025.

Manzil Zaheer, Satwik Kottur, Siamak Ravanbakhsh, Barnabás Póczos, Ruslan Salakhutdinov, and Alexander J. Smola. Deep Sets. In *Advances in Neural Information Processing Systems*, volume 30, 2017.

Xiaohua Zhai, Basil Mustafa, Alexander Kolesnikov, and Lucas Beyer. Sigmoid loss for language image pre-training. In *IEEE/CVF International Conference on Computer Vision*, pp. 11975–11986, 2023.

Jiarui Zhang, Mahyar Khayatkhoei, Prateek Chhikara, and Filip Ilievski. MLLMs know where to look: Training-free perception of small visual details with multimodal LLMs. In *International Conference on Learning Representations*, 2025a.

Xintong Zhang, Zhi Gao, Bofei Zhang, Pengxiang Li, Xiaowen Zhang, Yang Liu, Tao Yuan, Yuwei Wu, Yunde Jia, Song-Chun Zhu, and Qing Li. Adaptive Chain-of-Focus reasoning via dynamic visual search and zooming for efficient VLMs. *arXiv preprint arXiv:2505.15436*, 2025b.

Puzhuo Zheng and Hasan Kurban. It's the decoding format, not the perturbation: Auditing consistency-based selection for vision-language test-time scaling. *arXiv preprint arXiv:2608.01207*, 2026.

Enshen Zhou, Jingkun An, Cheng Chi, Yi Han, Shanyu Rong, Chi Zhang, Pengwei Wang, Zhongyuan Wang, Tiejun Huang, Lu Sheng, and Shanghang Zhang. RoboRefer: Towards spatial referring with reasoning in vision-language models for robotics. In *Advances in Neural Information Processing Systems*, volume 38, pp. 28404–28481, 2025.

A Relation to Prior Work

This appendix positions SpatialMind within four lines of prior work. Each paragraph closes with how this paper differs.

Test-time scaling, verifiers and agreement. Test-time computation is usually spent on several samples of one call, and self-consistency returns their majority answer (Wang et al., 2023). An adaptive variant sets the number of samples from the agreement so far (Aggarwal et al., 2023), and the best use of a budget depends on the query (Snell et al., 2025). For coordinates, agreement is measured as spatial overlap among repeated predictions (Du et al., 2026), and the dispersion of sampled points serves as an uncertainty signal (Wang et al., 2026). Learned verifiers score each candidate and return the top one (Cobbe et al., 2021). The share of queries with a correct sample keeps growing with their number, while selection by vote or verifier saturates (Brown et al., 2024). Lemma 1 makes this split between coverage and selection exact for any rule that returns an observed point. Classical aggregation weights fallible sources by their error rates under conditional independence (Dawid & Skene, 1979). A further source is worth less the more it depends on those already consulted (Clemen & Winkler, 1985). For a vote, this dependence is diagnosed by the correlation between the correctness of its members (Kuncheva & Whitaker, 2003). Our observations, by comparison, come from different procedures of a registry, some cropped around the outputs of others. Their dependence follows the chains of the registry (Table 1), as Proposition 2 models. Following the call for matched controls in consistency-based selection (Zheng & Kurban, 2026), we apply the same vote rule to the registry and to repeated samples (Table 2). We also include the registry vote as a training-free baseline and, as a basic learned baseline, a gradient-boosted selector with default settings over the observations' own features.

Visual search, zoom and visual prompting. Looking again at sub-regions helps a vision-language model on small targets (Wu & Xie, 2024). A frozen model often attends to the right region even when its answer is wrong (Zhang et al., 2025a). Training-free methods search a tree of zoomed regions (Shen et al., 2025) or zoom into candidate regions and aggregate the resulting predictions (Luo et al., 2025). In GUI grounding, zooming follows a fixed controller or is triggered only when sampled boxes disagree (Wu et al., 2025; Tang et al., 2026). Other work fine-tunes the model to zoom (Zhang et al., 2025b), whereas generated programs compose procedures over frozen models without training (Surís et al., 2023; Gupta & Kembhavi, 2023). Visual prompting draws marks onto the image for the model to refer to, as numbered regions in Set-of-Mark (Yang et al., 2023) or as a graph of objects (Frisoni et al., 2026). PIVOT iteratively samples candidate actions, marks them and lets the model choose (Nasiriany et al., 2024). RegionFocus, one of our baselines, aggregates its own crops by such a marked choice. V^* and DyFo guide the search with a trained search model and a visual expert, respectively (Wu & Xie, 2024; Li et al., 2025). These methods leave the final choice to the frozen model, with further calls or auxiliary models. SpatialMind instead keeps every observation it acquires from its registry. It selects among them with a learned answer model that also uses the prefill states already produced by these calls.

Acquisition under a cost and adaptive computation. Which feature to acquire next and when to stop is learned per input as a sequential decision process (Dulac-Arnold et al., 2011; Shim et al., 2018), with a reward that weighs the gain against the cost of each feature (Janisch et al., 2019). Metareasoning chooses the next computation by its expected value net of cost (Russell & Wefald, 1991). Cascades spend computation only where cheap stages are inconclusive (Chen et al., 2024). Routers commit to a model before any output is seen (Ong et al., 2025). Acquiring further generations at a cost per sample is an optimal stopping problem under independent draws (Kalayci et al., 2025). Approximate policy iteration alternates policy evaluation and greedy improvement (Bertsekas & Tsitsiklis, 1996), and rollouts from sampled states can train a classifier as the improved policy (Lagoudakis & Parr, 2003). We fit the policy of ASM by approximate policy iteration on exact rollouts over the observations of the training queries, instead of by Bellman regression (Section 5.2). Proposition 4 (Appendix B) bounds the loss of a round greedy in fitted action values against every fixed schedule, or the previous round, by twice the largest error of the fitted values per remaining decision. Unlike this line of work, our features are the outputs of a frozen model, and a parent's output determines the view of its child. The answer space is the set of acquired observations. The answer model of SM scores every partial history, and a single cost-conditioned policy covers every cost weight.

Set and graph operators over candidates. Permutation-invariant functions on a set are sums of per-element encodings (Zaheer et al., 2017). Message passing networks add edge features and separate message, update and readout phases (Gilmer et al., 2017). Variants weight the aggregation by attention (Veličković et al., 2018) and use relation-specific message functions (Schlichtkrull et al., 2018). Jumping knowledge networks read out the node states of all rounds (Xu et al., 2018). Graphs in vision-language work usually connect the objects of a scene (Frisoni et al., 2026). Gradient boosting (Friedman, 2001) has also been combined with neural networks. BGNN refines the predictions of boosted trees on tabular node features with a graph network and trains both jointly (Ivanov & Prokhorenkova, 2021). Our approach differs in that the nodes are the dependent observations of a frozen model and include its prefill states, and in that the correction is an additive log-odds term that starts at the prior (Proposition 1). In addition, the typed relations of SM can separate histories whose tree features coincide (Proposition 2), and a single answer model is used for both selection and acquisition.

B Proofs of the Propositions

This appendix proves Lemma 1 and Propositions 1 to 3, and states and proves Proposition 4, for any registry as defined in Section 2, unless a remark names that of Section 3. Probabilities are over queries drawn as in the evaluation set, with fitted components fixed and outputs determined by the query (Assumptions 1 and 3). A history H contains the available set $\mathcal{F}$ with its costs, the acquired procedures with their return indicators, points and prefill states, and the query embedding of the tree prior. It omits M_x, so $p_i(H)$ need not be 0 or 1. An empty maximum is 0, and a missing answer counts as wrong.

Proof of Lemma 1. The idea is that a correct answer needs a correct observation, and that, given H, a pick is correct with probability $p_i(H)$. The first fact gives the product in Eq. 3, with both sides 0 if $\mathbb{P}(\bar{y} = 1) = 0$. The second and the tower rule give $\mathbb{P}(y_{\hat{\imath}} = 1) = \mathbb{E}[p_{\hat{\imath}}(H)]$, as $\hat{\imath}$ is a function of H. As the pick is observed, this is at most the first bound, with equality if and only if the rule almost surely picks a point with the largest $p_i(H)$, as i^{B} does. As $y_i \le \bar{y}$ on $\mathcal{O}(H)$, the largest such $p_i(H)$ is at most $\mathbb{E}[\bar{y} \mid H]$, whose mean is the second bound. No answer model can exceed coverage. If H' adds acquisitions to H, then H is a function of H' and $\mathcal{O}(H) \subseteq \mathcal{O}(H')$, so coverage cannot fall. For $i \in \mathcal{O}(H)$, $p_i(H) = \mathbb{E}[p_i(H') \mid H]$, so Jensen's inequality for the maximum gives $\max_{i \in \mathcal{O}(H)} p_i(H) \le \mathbb{E}[\max_{i \in \mathcal{O}(H')} p_i(H') \mid H]$. Taking expectations, the Bayes selector cannot lose accuracy either. A policy that depends only on its history can be simulated on H_{full} (Assumption 3), so its stopped history is a function of H_{full}. Both bounds are thus largest under full acquisition. □

Proof of Proposition 1. The cross-entropy is minimised by the true log-odds, which Bayes' rule splits into the prior and a likelihood ratio. Fix an observed i and drop the index. Let $q = \mathbb{P}(y = 1 \mid \mathbf{t}, E)$ and $q_0 = \mathbb{P}(y = 1 \mid \mathbf{t}) = \mathbb{E}[q \mid \mathbf{t}]$. (a) Given $(\mathbf{t}, E)$, a score u has the expected cross-entropy $\Phi_q(u) = \log(1 + e^u) - qu$. Its derivative $\sigma(u) - q$ increases strictly, so Φ_q is strictly convex with the unique minimiser $\operatorname{logit} q$. By the tower rule, $\sigma(s^0 + \Delta)$ has the expected cross-entropy $\mathbb{E}[\Phi_q(s^0 + \Delta)] \ge \mathbb{E}[h(q)]$ for the binary entropy h, with equality if and only if $\Delta = \operatorname{logit} q - s^0$ almost surely. The gate does not restrict Δ, as $\alpha\delta$ ranges over $\mathbb{R}$. Bayes' rule gives $\operatorname{logit} q = \operatorname{logit} q_0 + \log \Lambda$, with Λ the ratio of the densities of E given $(y = 1, \mathbf{t})$ and $(y = 0, \mathbf{t})$. The minimiser is thus $\log \Lambda$ if $\sigma(s^0) = q_0$. The expected cross-entropy then falls from $\mathbb{E}[h(q_0)] = \mathrm{H}(y \mid \mathbf{t})$ at $\Delta = 0$ to $\mathrm{H}(y \mid \mathbf{t}, E)$, that is, by $I(y; E \mid \mathbf{t})$, what E adds to the tree features. In Eq. 12 (Appendix D.1), the weight $1/(4|\mathcal{O}(H)|)$ is a function of $\mathbf{t}$ and does not move the minimiser. (b) A zero last layer of ψ_δ (Eq. 11 in Appendix D.1) gives $\delta_i = 0$ and $s_i = s_i^0$ for every input, so Eq. 1 returns the argmax of $\mathcal{G}(\mathbf{t}_i)$, that is, SM's own tree prior used as a selector. These parameters lie in the class of SM, whose lowest loss is thus at most that of the prior.

(c) As i^* and i^{B} are functions of H, the accuracy gap is $\mathbb{E}[p_{i^{\mathrm{B}}}(H) - p_{i^*}(H)]$, as in Lemma 1. Given H, the expected cross-entropy of $\sigma(s_i)$ exceeds that of $p_i(H)$ by the Bernoulli divergence $\mathrm{KL}_i = \mathrm{KL}(p_i(H) \,\|\, \sigma(s_i))$, and Pinsker's inequality gives $|\sigma(s_i) - p_i(H)| \le (\mathrm{KL}_i/2)^{1/2}$. Since

$s_{i^*} \geq s_{i^{\mathrm{B}}}$, adding $\sigma(s_{i^*}) - \sigma(s_{i^{\mathrm{B}}}) \geq 0$ to the gap given H gives:

$$\begin{aligned} p_{i^{\mathrm{B}}}(H) - p_{i^*}(H) &\leq \{p_{i^{\mathrm{B}}}(H) - \sigma(s_{i^{\mathrm{B}}})\} + \{\sigma(s_{i^*}) - p_{i^*}(H)\} \\ &\leq 2 \max_{i \in \mathcal{O}(H)} |\sigma(s_i) - p_i(H)| \\ &\leq \{2 \sum\nolimits_{i \in \mathcal{O}(H)} \mathrm{KL}_i\}^{1/2}. \end{aligned}$$

Hence the gap given H is at most twice the largest calibration error, and Pinsker's inequality with $\mathrm{KL}_i \geq 0$ gives the last line. By Jensen's inequality, the gap is at most $\{2\,\mathbb{E} \sum_{i \in \mathcal{O}(H)} \mathrm{KL}_i\}^{1/2}$. The sum is the excess cross-entropy of the scores, and both vanish if $\sigma(s_i) = p_i(H)$ on $\mathcal{O}(H)$. □

Two-node ranking. Swapping misordered scores lowers the expected loss $\ell(H)$ of Eq. 12, so its minimisers order two observations as their posteriors do. Let H have two observed nodes, with $p_1(H) > p_2(H)$, and let P_{10} and P_{01} be the probabilities given H that only node 1 or only node 2 is correct, so $P_{10} - P_{01} = p_1 - p_2$. Given H, the two terms of $\ell(H)$ have the expectations $P_{10} \log(1 + e^{s_2 - s_1}) + P_{01} \log(1 + e^{s_1 - s_2})$ and $\{\Phi_{p_1}(s_1) + \Phi_{p_2}(s_2)\}/8$. As $\log(1 + e^u) - \log(1 + e^{-u}) = u$, exchanging s_1 and s_2 changes them by $(p_1 - p_2)(s_1 - s_2)$ and $(p_1 - p_2)(s_1 - s_2)/8$, which are negative if $s_1 < s_2$. At $s_1 = s_2$, raising s_1 and lowering s_2 lowers them at the rates $p_1 - p_2$ and $(p_1 - p_2)/8$. Every minimiser of either term or of $\ell(H)$ thus has $s_1 > s_2$. For $P_{10}, P_{01} > 0$, the pairwise term alone is minimised at $s_1 - s_2 = \log(P_{10}/P_{01})$, which depends on the joint law of the labels. □

Proof of Proposition 2. (a) A wrong child is drawn from a small crop around its parent, so it falls near the parent more often than a wrong whole-image point. The view Ω_j of a child contains z_i, as the rectangle of Eq. 17 does. Let $\mathcal{D}$ be the disc of radius $\bar{d}$ around z_i. Under M, given x, z_i and $y_j = 0$, z_j is uniform on $\Omega_j \setminus M_x$. If $\mathcal{D} \subseteq \Omega_j \setminus M_x$, z_j thus falls in $\mathcal{D}$ with probability $\pi\bar{d}^2/|\Omega_j \setminus M_x| \geq \pi\bar{d}^2/|\Omega_j|$. If $\mathcal{D}$ misses M_x but leaves Ω_j, and both sides of Ω_j are at least $2\bar{d}$, a quarter of $\mathcal{D}$ still lies in Ω_j, so the probability is at least $\pi\bar{d}^2/(4|\Omega_j|)$. A wrong point uniform over the image outside M_x falls in $\mathcal{D}$ with probability at most $\pi\bar{d}^2/(1 - |M_x|)$. The ratio of the two probabilities is thus at least $(1 - |M_x|)/|\Omega_j|$. In the registry of Section 3, the sides of the crop of v_j, $1 \leq j \leq 3$, are at most 0.7^j unless the 224-pixel minimum applies (Table 7), so $1/|\Omega_j| \geq 0.7^{-2j}$, which is 2.04, 4.16 and 8.50. On the 1,626 evaluation queries, the minimum raises a side of these crops on 3, 158 and 1,311 queries, and the median of $1/|\Omega_j|$ is 2.04, 4.18 and 6.12.

(b) We swap the point that agrees with z_i between the child and v_0 ($i = 4$ and $j = 5$ in the registry of Section 3). This gives two histories that the tree features cannot distinguish, one where the agreeing point is the child's and one where it is the plain call's. The child's view is smaller than the image, so a wrong child agrees with a wrong parent more easily than a wrong plain call does. The child's agreement is thus weaker evidence that z_i is correct.

Step 1 (equal features). Take $\nu > 0$ and points $\mathbf{p}, \mathbf{n}, \mathbf{f}$ with $\|\mathbf{n} - \mathbf{p}\| < 2\nu < \|\mathbf{f} - \mathbf{p}\|$, so $\mathbf{n}$ is near $\mathbf{p}$ and $\mathbf{f}$ is far from it. Let H and H' observe only v_0, v_i and v_j, with $z_i = \mathbf{p}$, and differ only in (z_j, z_0), which is $(\mathbf{n}, \mathbf{f})$ in H and $(\mathbf{f}, \mathbf{n})$ in H'. The features $\mathbf{t}_i$ depend on the other points only through the mask and their distances to $\mathbf{p}$ (Appendix D), so $\mathbf{t}_i(H) = \mathbf{t}_i(H')$. Both histories have the same Ω_i, as v_i is a root, and the same crop Ω_j around $\mathbf{p}$. We let the discs of radius 2ν around $\mathbf{p}$, $\mathbf{n}$ and $\mathbf{f}$ lie in $\Omega_i \cap \Omega_j$, as they do for small ν when $\mathbf{p}$ is interior to both views.

Step 2 (a law under M). Let M_x be the disc of radius ν around a centre $\mathbf{c}$, uniform on the centres whose disc lies in the image, and let $\mathbf{1}_{\mathbf{c}}(z) = \mathbf{1}[z \in M_x]$. Given $\mathbf{c}$, each label depends on the other labels and points only through its view, and is 1 with probability $\theta \in (0, 1)$ if M_x lies in the view and 0 otherwise. A correct point is uniform on M_x, a wrong one on its view outside M_x, and the other components of H are independent of these. This law satisfies M and $y_k = \mathbf{1}[z_k \in M_x]$. Given $\mathbf{c}$, if M_x lies in the view Ω_k, z_k thus has the density $\theta/(\pi\nu^2)$ on M_x and $(1 - \theta)/(|\Omega_k| - \pi\nu^2)$ on the rest of Ω_k, with $|\Omega_0| = 1$. A correct point has the same density in every view, and a wrong point is denser in a smaller view.

Step 3 (the posteriors). For $b \in \{0, 1\}$, let $w_b(\mathbf{c})$ be the density of $z_i = \mathbf{p}$ jointly with $y_i = b$ given $\mathbf{c}$, that is, how well a region with centre $\mathbf{c}$ explains $\mathbf{p}$ with the label b. Let $\Sigma_b(H)$ be the integral over $\mathbf{c}$ of $w_b(\mathbf{c})$ times the densities of z_j and z_0, that is, how well all regions together explain the three points with $y_i = b$. As $\mathbf{c}$ is uniform, the density of the points of H jointly with $y_i = b$ is proportional to $\Sigma_b(H)$, so the posterior odds of y_i are $\Sigma_1(H)/\Sigma_0(H)$. The swap changes the

product of the densities of z_j and z_0 only for centres whose disc M_x contains exactly one of $\mathbf{n}$ and $\mathbf{f}$. If it contains both or neither, z_j and z_0 each have the same density at $\mathbf{n}$ as at $\mathbf{f}$. If it contains one, M_x lies within 2ν of $\mathbf{n}$ or $\mathbf{f}$, hence in every view, and the densities of Step 2 apply. If M_x contains $\mathbf{n}$ but not $\mathbf{f}$, H has a correct child and a wrong plain call, H' has the reverse, and the product is as follows:

$$\text{in } H: \ \frac{\theta}{\pi\nu^2}\cdot\frac{1-\theta}{1-\pi\nu^2}, \qquad \text{in } H': \ \frac{1-\theta}{|\Omega_j|-\pi\nu^2}\cdot\frac{\theta}{\pi\nu^2}.$$

Only the wrong point's view differs, the image in H and Ω_j in H'. Let Γ be the second product minus the first, the gain of the swap at such a centre. If M_x contains $\mathbf{f}$ but not $\mathbf{n}$, the roles of H and H' swap, and the product falls by Γ. Integrating over $\mathbf{c}$ against w_b gives:

$$\Sigma_b(H')-\Sigma_b(H)=\Gamma\int w_b(\mathbf{c})\big\{\mathbf{1}_{\mathbf{c}}(\mathbf{n})-\mathbf{1}_{\mathbf{c}}(\mathbf{f})\big\}\,d\mathbf{c}, \qquad \Gamma=\frac{\theta(1-\theta)}{\pi\nu^2}\Big(\frac{1}{|\Omega_j|-\pi\nu^2}-\frac{1}{1-\pi\nu^2}\Big).$$

The swap therefore moves Σ_b by Γ times the w_b-weight of the centres whose disc contains $\mathbf{n}$, less that of the centres whose disc contains $\mathbf{f}$. Here $\Gamma>0$ as $|\Omega_j|<1$. A wrong point is denser in the child's smaller view, so the child's agreement is weaker evidence.

It remains to evaluate the integral for each b. Where the swap matters, M_x also lies in Ω_i, so w_1 is $\theta/(\pi\nu^2)$ for centres within ν of $\mathbf{p}$, w_0 is $(1-\theta)/(|\Omega_i|-\pi\nu^2)$ for the others, and each is 0 where the other is not. By Step 1, the centres within ν of $\mathbf{p}$ are farther than ν from $\mathbf{f}$. Let A be the area of the centres within ν of both $\mathbf{p}$ and $\mathbf{n}$, which is positive as $\|\mathbf{n}-\mathbf{p}\|<2\nu$. For $b=1$, the integrand is thus $\theta/(\pi\nu^2)$ on these centres and 0 elsewhere. For $b=0$, w_0 is 0 on these centres and $(1-\theta)/(|\Omega_i|-\pi\nu^2)$ on the other centres where the swap matters. The centres within ν of $\mathbf{n}$ and of $\mathbf{f}$ have equal areas, so only the first loses A. The case split on b gives:

$$\begin{aligned} b=1: &\quad \Sigma_1(H')-\Sigma_1(H)=\theta\Gamma A/(\pi\nu^2)>0,\\ b=0: &\quad \Sigma_0(H')-\Sigma_0(H)=-(1-\theta)\Gamma A/(|\Omega_i|-\pi\nu^2)<0. \end{aligned}$$

The swap thus raises Σ_1 and lowers Σ_0. As $\Sigma_0>0$ in both histories, the posterior odds Σ_1/Σ_0 of y_i are larger under H', and $\mathbb{P}(y_i=1\mid H)<\mathbb{P}(y_i=1\mid H')$. This holds on an open set of $(\mathbf{p},\mathbf{n},\mathbf{f})$, where the points have a positive density, so no version of the posterior is a function of $\mathbf{t}_i$ alone. The child relation $\mathcal{N}_2$ separates H and H' (Eqs. 8 and 9 in Appendix D.1), as e_{ij} begins with $\mathbf{n}-\mathbf{p}$ in H and $\mathbf{f}-\mathbf{p}$ in H'.

(c) Along a chain, the evidence that a node needs splits into what lies up to the node and what lies after it. The forward message conveys the first part from the parent, and the backward message conveys the second from the child. Each round moves each message one node further. After $L-1$ rounds every node thus has both parts, and an end node cannot have the other end's evidence sooner.

Step 1 (factorisation). Let u_k have the label y_k and the evidence ξ_k (its point, prefill state and offset from its parent), which form its pair (y_k,ξ_k), and write p for densities. For a label value y, the forward message $F_k(y)=p(y_k=y,\xi_{1:k})$ is the joint density of the label y at u_k and the evidence up to u_k. The backward message $B_k(y)=p(\xi_{k+1:L}\mid y_k=y,\xi_k)$ is the density of the evidence after u_k given the pair of u_k, with $B_L\equiv 1$ as nothing follows u_L. As the pairs form a Markov chain, the pairs after k are independent of those before given (y_k,ξ_k). The chain rule, with this independence in the last step, gives:

$$\mathbb{P}(y_k=y\mid\xi_{1:L})\propto p(y_k=y,\xi_{1:L})=F_k(y)\,p(\xi_{k+1:L}\mid y_k=y,\xi_{1:k})=F_k(y)B_k(y).$$

The posterior of u_k is thus the normalised product of its two messages, one for the evidence up to u_k and one for the evidence after it.

Step 2 (recursions). Let $\Psi_k(y',y)=p(y_k=y,\xi_k\mid y_{k-1}=y',\xi_{k-1})$ be the density of the pair of u_k given the pair of its parent, for a parent label y'. Summing over the parent's label in F_k and over the child's label in B_k, and using the Markov chain again, gives:

$$\begin{aligned} F_k(y)&=\textstyle\sum_{y'}F_{k-1}(y')\,\Psi_k(y',y), && 2\le k\le L,\\ B_k(y)&=\textstyle\sum_{y'}\Psi_{k+1}(y,y')\,B_{k+1}(y'), && 1\le k\le L-1. \end{aligned}$$

As Ψ_k depends on the evidence of u_k and of its parent, F_k needs the parent's message and B_k the child's, as in the sum–product algorithm on a chain (Pearl, 1988; Kschischang et al., 2001).

Step 3 ($L-1$ rounds suffice). In a synchronous round, every node applies the recursions to its parent's and child's messages of the previous round. The messages $F_1 = p(y_1, \xi_1)$ and $B_L \equiv 1$ are fixed, and all others start from arbitrary values. By induction on ℓ, after ℓ rounds the forward messages are exact up to node $\ell + 1$ and the backward ones from node $L - \ell$. The fixed messages give the case $\ell = 0$. For the step, a round makes F_k exact if F_{k-1} is exact before it, and B_k if B_{k+1} is. Node k thus has both exact messages, and by Step 1 its exact posterior, from round $\max(k-1, L-k)$. This round is at most $L - 1$, so every node is exact from round $L - 1$.

Step 4 ($L - 1$ rounds are needed). Let each node start from its own evidence, and let each round use only the node itself, its parent and its child. By induction, the state of u_k after ℓ rounds then depends on ξ_j only for $|j - k| \le \ell$. Let $y_1 \sim \mathrm{Bernoulli}(1/2)$, $y_k = y_{k-1}$ and $\xi_k \sim \mathcal{N}(y_k, 1)$ given y_k. These pairs form a Markov chain, and every label equals y_1. Each ξ_k adds its log-likelihood ratio $\xi_k - 1/2$ to the prior logit 0, so $\operatorname{logit} \mathbb{P}(y_1 = 1 \mid \xi_{1:L}) = \sum_{k=1}^{L} (\xi_k - 1/2)$. This depends on ξ_L, which reaches u_1 only after $L - 1$ rounds. Hence u_1, and likewise u_L, needs $L - 1$ rounds in general.

As a child is acquired only if its parent returned a point, the observed nodes of a chain form an initial segment of it. In the registry of Section 3, the longest chain, $v_0 \to \cdots \to v_3$, has $L = 4$, so three rounds suffice for every chain. The relation $\mathcal{N}_3$ reaches every observed node in one round. □

Proof of Proposition 3. Parts (a) and (b) verify which information each rule uses, and (c) compares the optima at two weights. (a) The scores depend only on the acquired procedures (Section 5.1), and $\chi(H_t)$ of Eq. 15 (Appendix D.2) summarises these scores, the points and the acquired procedures. The cell of V^{st}_λ and β_λ is determined by H_t, which contains $\mathcal{F}$, and r_ϑ takes $\chi(H_t)$, $\mathbf{k}_{\mathrm{act}}$ and λ as input. With a fixed tie-break, Eq. 6 thus depends only on H_t, which contains no output of a procedure not yet acquired. (b) At $\lambda = 0$, Eq. 6 acquires feasible procedures in a fixed order until none is feasible. This depends only on $\mathcal{F}$ and on which procedures returned a point, and ends at SM's full history, answer and calls (Assumption 3). The plain call acquires v_0 and stops. A fixed schedule acquires its preset procedures in order, parents first, skipping infeasible ones, and stops, or acquires v_0 if it would stop on the empty history. Both depend only on what was acquired and what is feasible. Each acquisition runs a new procedure, so an episode ends within $n + 1$ decisions, and backward induction over histories with a fixed tie-break gives a maximiser $\pi^\star_\lambda$ (Bertsekas & Tsitsiklis, 1996).

(c) Let $\pi = \pi^\star_\lambda$ and $\pi' = \pi^\star_{\lambda'}$ for $0 \le \lambda < \lambda'$. *Step 1 (calls do not increase).* As each policy is optimal at its own weight, $J_\lambda(\pi) \ge J_\lambda(\pi')$ and $J_{\lambda'}(\pi') \ge J_{\lambda'}(\pi)$. Writing these out and adding them gives:

$$\begin{aligned} \mathrm{acc}(\pi) - \lambda\,\mathrm{calls}(\pi) &\ge \mathrm{acc}(\pi') - \lambda\,\mathrm{calls}(\pi'), \\ \mathrm{acc}(\pi') - \lambda'\,\mathrm{calls}(\pi') &\ge \mathrm{acc}(\pi) - \lambda'\,\mathrm{calls}(\pi), \\ (\lambda' - \lambda)\{\mathrm{calls}(\pi) - \mathrm{calls}(\pi')\} &\ge 0. \end{aligned}$$

The third line is their sum, in which the accuracies cancel. As $\lambda' > \lambda$, it gives $\mathrm{calls}(\pi) \ge \mathrm{calls}(\pi')$. Hence a larger cost weight never leads to more calls. *Step 2 (accuracy does not increase).* Rearranging the first line of Step 1 gives:

$$\begin{aligned} \mathrm{acc}(\pi) - \mathrm{acc}(\pi') &\ge \lambda\{\mathrm{calls}(\pi) - \mathrm{calls}(\pi')\} \\ &\ge 0, \end{aligned}$$

where the second line uses $\lambda \ge 0$ and Step 1. A larger weight thus never raises the accuracy either. *Step 3 (Pareto optimality).* Let $\lambda > 0$, and let $\pi'' \in \Pi$ have $\mathrm{acc}(\pi'') \ge \mathrm{acc}(\pi)$ and $\mathrm{calls}(\pi'') \le \mathrm{calls}(\pi)$, with one of the two strict. As $\lambda > 0$, this gives:

$$J_\lambda(\pi'') = \mathrm{acc}(\pi'') - \lambda\,\mathrm{calls}(\pi'') > \mathrm{acc}(\pi) - \lambda\,\mathrm{calls}(\pi) = J_\lambda(\pi),$$

which contradicts the optimality of π. Hence no policy of Π improves on π in accuracy or calls without being worse in the other. *Step 4 (convexity).* For a fixed $\pi'' \in \Pi$, $J_\lambda(\pi'')$ is a line in λ with slope $-\mathrm{calls}(\pi'') \le -1$, as every policy acquires at least once, at a cost of at least one call. The optimum $J^\star_\lambda = \max_{\pi'' \in \Pi} J_\lambda(\pi'')$ is the upper envelope of these lines, so it is convex. *Step 5 (strict decrease).* As π' attains $J^\star_{\lambda'}$, following its line from λ to λ' gives:

$$\begin{aligned} J^\star_{\lambda'} &= J_\lambda(\pi') - (\lambda' - \lambda)\,\mathrm{calls}(\pi') \\ &\le J^\star_\lambda - (\lambda' - \lambda), \end{aligned}$$

where the second line uses $J_\lambda(\pi') \le J^\star_\lambda$ and $\mathrm{calls}(\pi') \ge 1$. The optimum thus falls by at least $\lambda' - \lambda$ and is strictly decreasing. *Step 6 (subgradient).* The line of π touches the envelope at λ, as $J_\lambda(\pi) = J^\star_\lambda$. Comparing the envelope with this line at any $\mu \ge 0$ gives:

$$\begin{aligned} J^\star_\mu &\ge J_\mu(\pi) \\ &= J^\star_\lambda - (\mu - \lambda)\,\mathrm{calls}(\pi). \end{aligned}$$

Hence the line of π never lies above the optimum, and its slope $-\mathrm{calls}(\pi)$ is a subgradient at λ. □

Values on histories. Fix $\lambda \ge 0$. An acquisition reveals the outputs of its procedure at the cost $c(\mathrm{act})$, which the history fixes, and STOP ends with the reward $y_{i^*(H)}$. For $\pi \in \Pi$, $V^\pi(H)$ is the expected return of following π from H, and $Q^\pi(H, \mathrm{act})$ that of taking act first. Their maxima over the finite set Π^{fix} of fixed schedules are $\bar V$ and $\bar Q$, the values of the best fixed schedule from H. If H contains k acquired procedures, at most $N(H) = n + 1 - k$ decisions remain.

Proposition 4 (Approximate policy improvement): By Assumptions 1 and 3, acquisition is a Markov decision process on histories. (a) Let $\pi \in \Pi$ be greedy in some $\tilde Q$ with $|\tilde Q - \bar Q| \le \varepsilon$ on the feasible actions of every history that π reaches. Then $V^\pi(H) \ge \bar V(H) - 2\varepsilon N(H)$ at each such H, and $J_\lambda(\pi) \ge \max_{\pi' \in \Pi^{\mathrm{fix}}} J_\lambda(\pi') - 2(n+1)\varepsilon$, including the plain call and SM. At $\varepsilon = 0$, $V^\pi \ge \bar V$, the guarantee that Chang et al. (2004) prove for parallel rollout. (b) If $\pi_k \in \Pi$ is greedy in some $\tilde Q$ with $|\tilde Q - Q^{\pi_{k-1}}| \le \varepsilon$ in the same sense, then $V^{\pi_k}(H) \ge V^{\pi_{k-1}}(H) - 2\varepsilon N(H)$ at each history it reaches, and $J_\lambda(\pi_k) \ge J_\lambda(\pi_{k-1}) - 2(n+1)\varepsilon$ (Singh & Yee, 1994; Bertsekas & Tsitsiklis, 1996). (c) Given H_t, the mean of $G_\lambda(H_t, \mathrm{act})$ of Eq. 5 over training queries is the value of act followed by π_{k-1}, with out-of-fold scores for policy and answer. The error ε of a value fitted to them thus includes regression and sampling errors and the change to the answer model used at test time, but no bootstrap bias.

Proof of Proposition 4. The idea is that a greedy step on values accurate to ε loses at most 2ε, and at most $N(H)$ decisions remain. *Step 1 (Markov property).* A query fixes its outputs, whatever the order of acquisition (Assumptions 1 and 3). A policy in Π decides from what it has revealed, so whether it reaches $H_t = h$ depends only on the outputs in h. The next output thus has the same law given h under every policy, so the process on histories is Markov, with transitions free of the policy. Backward induction on N gives $Q^\pi(H, \mathrm{STOP}) = p_{i^*(H)}(H)$ and $Q^\pi(H, \mathrm{act}) = -\lambda c(\mathrm{act}) + \mathbb{E}[V^\pi(H') \mid H, \mathrm{act}]$ for an acquisition with next history H'. Stopping thus earns the same under every policy. *Step 2 (a greedy step loses at most 2ε).* For a finite $\Pi' \subseteq \Pi$, let $\bar V$ and $\bar Q$ here be the maxima over Π', and let π be greedy in $\tilde Q$ with $|\tilde Q - \bar Q| \le \varepsilon$ at the feasible pairs of histories it reaches. At such an H, $\tilde Q$ ranks $\pi(H)$ no lower than the best action of $\bar Q(H, \cdot)$ and errs by at most ε at each, so $\bar Q(H, \pi(H)) \ge \max \bar Q(H, \cdot) - 2\varepsilon \ge \bar V(H) - 2\varepsilon$, as each $\pi' \in \Pi'$ takes a feasible action at H. *Step 3 (the losses add up).* We show $V^\pi(H) \ge \bar V(H) - 2\varepsilon N(H)$ by induction on $N(H)$. If π stops at H, as it must when $N(H) = 1$, then $V^\pi(H) = \bar Q(H, \mathrm{STOP})$, and Step 2 gives the bound. Otherwise π acquires $\mathrm{act} = \pi(H)$, $N(H') = N(H) - 1$, and $\bar Q(H, \mathrm{act}) \le -\lambda c(\mathrm{act}) + \mathbb{E}[\bar V(H') \mid H, \mathrm{act}]$, as each $Q^{\pi'}$ satisfies it. Applying the recursion of Step 1 for π with the induction hypothesis at H', and then this bound, gives:

$$\begin{aligned} V^\pi(H) &\ge -\lambda c(\mathrm{act}) + \mathbb{E}[\bar V(H') \mid H, \mathrm{act}] - 2\varepsilon\{N(H) - 1\} \\ &\ge \bar Q(H, \mathrm{act}) - 2\varepsilon\{N(H) - 1\}. \end{aligned}$$

Step 2 then gives the bound, so each remaining decision adds at most 2ε to the loss. Every policy starts at the empty history H_0, with $N(H_0) = n + 1$ (8 for the registry of Section 3), so $J_\lambda(\pi) = \mathbb{E}[V^\pi(H_0)] \ge \mathbb{E}[\bar V(H_0)] - 2(n+1)\varepsilon \ge \max_{\pi' \in \Pi'} J_\lambda(\pi') - 2(n+1)\varepsilon$.

(a) Take $\Pi' = \Pi^{\mathrm{fix}}$, which contains the plain call and SM as the preset sets $\{v_0\}$ and $\mathcal{V}$, the latter in the order of Eq. 6 at $\lambda = 0$. Parallel rollout is greedy in the one-step lookahead on $\bar V$, an upper bound on $\bar Q$ (Step 3), and at $\varepsilon = 0$ Step 3 gives its guarantee to a policy greedy in $\bar Q$. (b) Take $\Pi' = \{\pi_{k-1}\}$. At $\varepsilon = 0$ this is policy improvement (Singh & Yee, 1994; Bertsekas & Tsitsiklis, 1996). (c) By Assumption 3, the rollout of act, then π_{k-1}, simulated on a training query's precomputed observations, visits the same histories and incurs the same costs as online acquisition, and ends with the label of the out-of-fold answer. Its return is thus observed rather than estimated, with mean $Q^{\pi_{k-1}}(H_t, \mathrm{act})$ up to the change to the test-time answer model and to new queries. □

The bound for ASM. In ASM, π_0 is greedy in V^{st}_λ, the best cell mean of $Q^{\pi'}$ over Π^{fix}. Each π_k, $k = 1, 2$, is greedy in the value fitted to the returns of π_{k-1}, and π_2 is the policy used at

test time. With errors ε_0, ε_1 and ε_2, part (a) for π_0 and part (b) for each round give $J_\lambda(\pi_2) \geq \max_{\pi' \in \Pi^{\text{fix}}} J_\lambda(\pi') - 2(n+1)(\varepsilon_0 + \varepsilon_1 + \varepsilon_2)$. Here ε_0 includes the variation of $Q^{\pi'}(H_t, \text{act})$ within a cell and the sampling error of the cell means, and ε_1 and ε_2 include what H_t adds to $\chi(H_t)$, λ and the cell, and the regression error. All three include the change to the answer model used at test time. As V_λ^{st} is constant on a cell while $p_{i^*(H_t)}(H_t)$ varies within it, the bound motivates exact rollout targets.

C Dependence Diagnostics

This appendix presents the dependence measurements behind Section 3. They are computed, without any fitted model, from the registry observations and from ten samples of the plain call per query at $T = 0.3$ and $T = 0.6$. Table 4 lists, per backbone, the mean pairwise correlation of correctness ρ (Kuncheva & Whitaker, 2003) and the coverage, the share of queries with a correct observation. For the registry, the correlation is also split into pairs inside a chain and across the two chains. Each pair is restricted to the queries on which both procedures returned a point. Restricting every pair to the queries on which all procedures returned a point changes the correlation of the registry by at most 0.003. Samples are shown for seven and for ten draws, and ten draws cost more than the 7.91 calls of the registry. The vote of Table 2, the registry vote of Table 3, returns the observed point with the most observed points within a radius of 0.05. A tie goes to the point with the smallest summed distance to them. The same rule applies to the registry and to the samples. A training-free vote is sensitive to its radius. At a radius of 0.1, the registry vote falls below the plain call on RR-2B, 61.87 against 63.04%.

Within the registry, correctness is more correlated inside a chain than across the two chains on every backbone. On RR-8B and Molmo-7B-D the correlation inside a chain, 0.71 to 0.77, is comparable to that of samples at $T = 0.6$, 0.73 to 0.77, and below that at $T = 0.3$, 0.82 to 0.87. The lower overall correlation of the registry comes from the pairs across the two chains and from the deep crops of RR-2B, which fail even under a correct parent (Table 1). A low correlation is thus not by itself a sign of useful diversity. We therefore use coverage, the first factor of Lemma 1, as the evidence in Section 3. With ten samples, and hence more calls, repeated sampling still has lower coverage than the registry on every backbone.

Table 5 separates what each edge corrects from what it inherits. The first self-zoom step corrects 19.15 to 24.46% of its parent's errors and loses 7.01 to 9.16% of its parent's correct answers. Its estimated net gain is positive on every backbone and $+1.89$ $[0.85, 2.91]$ pooled. On RR-2B the edges $v_2 \to v_3$, $v_4 \to v_5$ and $v_5 \to v_6$ lose 46.50 to 64.47% of their parent's correct answers, against at most 11.93% on every other edge of the three backbones. Where v_1 or v_2 is wrong, the plain call is still correct on 16.40 to 24.07% of those queries, so the points before a failed step remain candidates for the answer. This dependence of a child on its parent is consistent with the chain structure of Proposition 2.

Table 4: Dependence of the registry observations and of repeated samples of the plain call. Mean pairwise correlation of correctness (ρ) with its 95% interval, each pair taken over the queries on which both procedures returned a point, and share of queries with a correct observation (any, %). For the registry, ρ is also given within the self-zoom chain v_0–v_3, within the anchor-zoom chain v_4–v_6 and across the two chains.

	RR-8B		RR-2B		Molmo-7B-D	
set	ρ [95% CI]	any	ρ [95% CI]	any	ρ [95% CI]	any
registry (n=7 observations)	0.645 [0.617, 0.673]	84.19	0.360 [0.336, 0.385]	80.32	0.658 [0.634, 0.683]	79.64
within the self-zoom chain v_0–v_3	0.706	–	0.428	–	0.705	–
within the anchor-zoom chain v_4–v_6	0.771	–	0.411	–	0.741	–
across the two chains	0.583	–	0.314	–	0.614	–
samples, T=0.3 (n=7)	0.869 [0.850, 0.888]	76.94	0.826 [0.807, 0.844]	71.77	0.821 [0.801, 0.840]	71.59
samples, T=0.3 (n=10)	0.871 [0.852, 0.888]	77.49	0.826 [0.808, 0.843]	72.57	0.827 [0.808, 0.844]	72.82
samples, T=0.6 (n=7)	0.773 [0.750, 0.796]	80.07	0.740 [0.718, 0.761]	74.97	0.726 [0.704, 0.747]	76.63
samples, T=0.6 (n=10)	0.774 [0.752, 0.796]	80.87	0.733 [0.713, 0.753]	77.00	0.733 [0.713, 0.753]	77.92

Table 5: Recovery, loss and net gain of every edge of the registry. For the queries on which the parent is wrong, the table gives their number, the share on which the child is correct (recovered, %) and the accuracy of the plain call on the same queries (plain, %). For the queries on which the parent is correct, it gives their number and the share on which the child is wrong (lost, %). The last column is the net gain of returning the child instead of the parent, in points with its 95% interval. All queries are counted, and a procedure that produced no observation on a query is wrong there, so a query for which no landmark is provided falls under parent wrong on the anchor-zoom edges. On the first self-zoom edge the plain call is the parent, so that column is left empty. The pooled rows average the per-query difference over the three backbones.

		parent wrong			parent correct		
backbone	edge	queries	recovered	plain	queries	lost	net gain
RR-8B	$v_0 \to v_1$	470	19.15	–	1156	7.01	+0.55 [−0.96, 2.07]
	$v_1 \to v_2$	461	16.92	17.57	1165	5.84	+0.62 [−0.81, 2.07]
	$v_2 \to v_3$	451	13.53	19.96	1175	6.47	−0.92 [−2.40, 0.49]
	$v_4 \to v_5$	666	12.91	46.10	960	6.98	+1.17 [−0.36, 2.68]
	$v_5 \to v_6$	647	7.73	47.60	979	5.62	−0.31 [−1.49, 0.92]
RR-2B	$v_0 \to v_1$	601	24.46	–	1025	8.88	+3.44 [1.66, 5.34]
	$v_1 \to v_2$	545	20.00	16.70	1081	11.93	−1.23 [−3.17, 0.74]
	$v_2 \to v_3$	565	9.20	24.07	1061	64.47	−38.87 [−41.94, −35.81]
	$v_4 \to v_5$	895	6.82	46.15	731	59.51	−23.00 [−25.62, −20.43]
	$v_5 \to v_6$	1269	2.92	58.47	357	46.50	−7.93 [−9.66, −6.21]
Molmo-7B-D	$v_0 \to v_1$	600	20.17	–	1026	9.16	+1.66 [−0.12, 3.42]
	$v_1 \to v_2$	573	15.53	16.40	1053	9.97	−0.98 [−2.65, 0.69]
	$v_2 \to v_3$	589	13.92	19.02	1037	9.16	−0.80 [−2.38, 0.79]
	$v_4 \to v_5$	766	13.71	37.86	860	9.65	+1.35 [−0.31, 3.04]
	$v_5 \to v_6$	744	9.27	37.63	882	8.73	−0.49 [−1.87, 0.94]
pooled	$v_0 \to v_1$	–	–	–	–	–	+1.89 [0.85, 2.91]
	$v_1 \to v_2$	–	–	–	–	–	−0.53 [−1.55, 0.46]
	$v_2 \to v_3$	–	–	–	–	–	−13.53 [−14.80, −12.29]
	$v_4 \to v_5$	–	–	–	–	–	−6.83 [−8.03, −5.65]
	$v_5 \to v_6$	–	–	–	–	–	−2.91 [−3.76, −2.03]

D Implementation Details

D.1 Answer Model

This appendix presents the equations that Sections 4 and 5 state in words, and what is needed to reproduce SM and ASM on the registry of Section 3, with the constants in Table 6.

Node and edge attributes. Each procedure $v_i \in \mathcal{V}$ is one node, whose initial state is given as:

$$h_i^{(0)} = o_i \,\mathrm{GELU}\big(W_{\mathrm{in}}\,[\,P\hat{\eta}_i\,;\,\mathbf{t}_i\,;\,s_i^0\,]\big) \in \mathbb{R}^{32}, \qquad \hat{\eta}_i = \sqrt{D}\,\eta_i/\|\eta_i\|, \tag{7}$$

where $\eta_i \in \mathbb{R}^D$, the prefill state, is the backbone's hidden state at the last input position of the call returning z_i, after the final normalisation and before the language-model head, and $P \in \mathbb{R}^{64\times D}$ and W_{in} are learned. A procedure that returned no point has $o_i = 0$ and starts from a zero state. The geometry of each ordered pair (i, j) of nodes is encoded in the edge attribute, as follows:

$$e_{ij} = \big[\,z_j - z_i\,;\; d_{ij}^2\,;\; b_{ij}\,;\; \phi_1(d_{ij}), \ldots, \phi_5(d_{ij})\,\big] \in \mathbb{R}^9, \tag{8}$$

where $d_{ij} = \|z_j - z_i\|$ in the normalised frame, $b_{ij} = o_i o_j$ indicates a pair whose ends both returned a point, and $\phi_k(d) = \exp(-d^2/2\kappa_k^2)$ with $(\kappa_1, \ldots, \kappa_5) = (0.02, 0.05, 0.1, 0.2, 0.4)$.

Messages. The three relations are restricted to the observed set $\mathcal{O} = \mathcal{O}(H)$. $\mathcal{N}_1(i)$ contains the parent of i and $\mathcal{N}_2(i)$ its children, both given by $\mathcal{E}$, and $\mathcal{N}_3(i) = \mathcal{O} \setminus \{i\}$ contains every other observed node. At round ℓ, the message from j to i over relation r and the update of the state are

Table 6: Hyperparameters of the answer model and the acquisition policy, fixed for all folds.

answer model		*acquisition policy*	
boosting iterations / leaves	80 / ≤ 7, no early stopping	inputs χ / $\mathbf{k}_{\text{act}}$ / cost	17 / 14 / $\lambda/0.05$
min. leaf / ℓ_2 / learning rate	40 / 10 / 0.08	r_ϑ (GELU) / parameters	32→32→1 / 1,089
neighbours $\lvert\mathcal{K}(x)\rvert$	25, cosine	V^{st} tables	$20 \times 8 \times 3 \times 20$
P / W_{in}	D→64 / 101→32	inner folds	3, image-grouped
W_r / g_r, per relation	41→32 / 73→16→1	policy-iteration rounds	2
U / dropout / rounds	32→32 / 0.2 / 3	AdamW steps per round	1,000
W_J / ψ_δ, ψ_α	128→32 / 64→16→1	AdamW lr / weight decay	0.01 / 10
parameters, D=3,584 / 1,536	0.25M / 0.12M	groups per step	1,365
AdamW lr / weight decay	10^{-3} / 0.01	gradient clip	5
batch / epochs / gradient clip	64 / 10 / 5	cost weights	7, trained jointly
prefill dropout / γ	0.5 / 0.1	precision	float64, CPU

given as:

$$\mu_{ij}^r = \sigma\big(g_r([\,h_i^{(\ell)}; h_j^{(\ell)}; e_{ij}\,])\big)\, W_r\,[\,h_j^{(\ell)}; e_{ij}\,], \qquad m_i^r = \frac{1}{\max(1, |\mathcal{N}_r(i)|)} \sum_{j \in \mathcal{N}_r(i)} \mu_{ij}^r, \tag{9}$$

$$h_i^{(\ell+1)} = \mathrm{LN}\Big(h_i^{(\ell)} + \mathrm{Dropout}\big(\mathrm{GELU}\big(U h_i^{(\ell)} + \textstyle\sum_{r=1}^{3} m_i^r\big)\big)\Big), \tag{10}$$

where W_r and g_r are the message map and gate of relation r, U is a self transform, LN is layer norm, and Dropout has rate 0.2. Rounds $\ell = 0, 1, 2$ share weights.

Readout. After the last round, the correction and the gate of Eq. 4 are read out as follows:

$$\tilde{h}_i = \mathrm{GELU}\big(W_J[\,h_i^{(0)}; \ldots; h_i^{(3)}\,]\big), \qquad \bar{u} = \frac{1}{|\mathcal{O}|} \sum_{j \in \mathcal{O}} \tilde{h}_j,$$

$$\delta_i = \psi_\delta\big([\,\tilde{h}_i; \bar{u}\,]\big), \qquad \alpha_i = \sigma\big(\psi_\alpha([\,\tilde{h}_i; \bar{u}\,])\big), \tag{11}$$

where $\tilde{h}_i$ is a jumping-knowledge readout (Xu et al., 2018) and ψ_δ, ψ_α are two-layer networks. The last layer of ψ_δ is initialised at zero, so $\delta_i = 0$ and $s_i = s_i^0$ at the start of training.

Loss. The answer model minimises the following loss:

$$\ell(H) = \frac{1}{|\mathcal{P}(H)|} \sum_{(i,j) \in \mathcal{P}(H)} \log\big(1 + e^{\,s_j(H) - s_i(H)}\big) + \frac{1}{4\,|\mathcal{O}(H)|} \sum_{i \in \mathcal{O}(H)} \mathrm{BCE}\big(\sigma(s_i(H)), y_i\big),$$

$$\mathcal{L} = (1 - \gamma)\,\ell(H_{\text{full}}) + \gamma\,\ell(\tilde{H}), \tag{12}$$

where $\mathcal{P}(H) = \{(i,j) \in \mathcal{O}(H)^2 : y_i = 1,\, y_j = 0\}$ and the first term is the pairwise logistic loss (Burges et al., 2005), omitted when $\mathcal{P}(H)$ is empty. The partial history $\tilde{H}$ is drawn uniformly among those the query can reach, and $\gamma = 0.1$. We chose the weight γ and the number of epochs as described in Appendix D.3.

Prefill state. The state η_i of Eq. 7 is the input of the language-model head in the prefill pass of the call that returns z_i. Its width D is 3,584 for RR-8B and Molmo-7B-D and 1,536 for RR-2B. Every linear map of Eqs. 7 to 11 has a bias term, which the equations omit.

Tree features. The $4n + 8$ features $\mathbf{t}_i$ of Section 4.1, 36 for $n = 7$, form six groups: procedure identity (n), coordinate z_i (2), observation mask (n), success rate ϱ_k of each procedure (n), retrieval prior $\hat{\varrho}_k(x)$ (n) and agreement (6). The success rate ϱ_k is the fraction of correct points among those v_k returned on the training split, and 0.5 if it returned none. The retrieval prior of procedure v_k is given as:

$$\hat{\varrho}_k(x) = \frac{\varrho_k + \sum_{q \in \mathcal{K}(x)} o_{qk}\, y_{qk}}{1 + \sum_{q \in \mathcal{K}(x)} o_{qk}}, \tag{13}$$

where $\mathcal{K}(x)$ is the set of the 25 training queries nearest to x in cosine distance of a frozen 1,920-dimensional embedding, and o_{qk} and y_{qk} are the return indicator and label of v_k on training query q. The embedding concatenates the SigLIP image embedding (1,152) (Zhai et al., 2023) and the

MPNet instruction embedding (768) (Reimers & Gurevych, 2019). The retrieval prior smooths the neighbours' success towards ϱ_k with one pseudo-count. The agreement group consists of the number of other observed points, the nearest and mean distances to them, and their counts within radii 0.025, 0.05 and 0.1. The nearest distance is 2 when no other point is observed.

Tree prior. $\mathcal{G}$ is the HistGradientBoosting classifier of scikit-learn (Pedregosa et al., 2011). Its training examples are the observations at every reachable state of every training query, weighted so that each query counts once. We choose a pointwise or a class-balanced objective by three-fold image-grouped cross-validation on the training split. Within one training split, the success rates and retrieval priors of the training examples are cross-fitted over the same folds, so no example uses statistics computed on its own image. The correction is trained with nested cross-fitting, in which the features and prior logits of each of its training examples come from trees fitted on the other two inner folds. The correction is thus trained on prior log-odds for images not used to fit the trees, as at test time, which is the setting of Proposition 1. The prior used at test time is fitted on the whole training split. Probabilities are clipped to $[10^{-6}, 1 - 10^{-6}]$ before the logit.

Correction training. The partial term of Eq. 12 draws one reachable state per query in each batch and uses the cross-fitted features and logits of that state. Prefill dropout sets the projected states of all n nodes of a query to zero together, without rescaling, and is disabled at test time. Training and inference run in double precision on one CPU thread with a fixed seed. Each query is scored on its own, so the scores depend neither on batching nor on the order of acquisition.

D.2 Acquisition Policy

Feasible actions. For a general registry, the feasible set of Section 2 is given as:

$$\textsc{stop} \in \mathcal{A}(H_t) \iff H_t \neq \emptyset, \qquad v_i \in \mathcal{A}(H_t) \iff v_i \in \mathcal{F} \setminus H_t \ \wedge\ \mathrm{pa}(v_i) \subseteq \mathcal{O}(H_t), \tag{14}$$

where $\mathrm{pa}(v_i)$ is the parent of v_i, empty for a root, so acquisition may start at any root.

Action value. The value of Section 5.2 is given as:

$$Q_\lambda(H_t, \mathrm{act}) = V^{\mathrm{st}}_\lambda(\zeta_t, \mathrm{act}) + \beta_\lambda(\zeta_t, \mathrm{act}) + \bar{r}_\vartheta\big(\chi(H_t), \mathbf{k}_{\mathrm{act}}, \lambda\big), \tag{15}$$

where $\zeta_t = (a, b, \mathcal{A}(H_t))$, the cell of H_t, consists of its state, given by the numbers a and b of procedures acquired on the self-zoom and anchor chains, and its feasible set. A fixed schedule $\pi \in \Pi^{\mathrm{fix}}$ runs each chain to a preset depth or its first failure, and $V^{\mathrm{st}}_\lambda(\zeta_t, \mathrm{act}) = \max_{\pi \in \Pi^{\mathrm{fix}}} \hat{Q}^{\pi}_\lambda(\zeta_t, \mathrm{act})$ is the largest mean return of act followed by π over training queries in ζ_t, answered out of fold. The offset β_λ is fitted per cell, and $\bar{r}_\vartheta$ centres a zero-initialised head r_ϑ over $\mathcal{A}(H_t)$. The head takes as input a summary $\chi(H_t)$ of the history, the action $\mathbf{k}_{\mathrm{act}}$ and λ, which are listed below.

Policy iteration. The first policy π_0 is greedy in V^{st}_λ. In round $k \in \{1, 2\}$, the returns G_λ of Eq. 5 follow π_{k-1}. We fit r_ϑ by squared error to these returns, with their out-of-fold cell mean removed and centred over the feasible actions. We then re-estimate β_λ out of fold as the cell mean of $G_\lambda - \bar{r}_\vartheta - V^{\mathrm{st}}_\lambda$, and π_k is greedy in the new value.

States and feasible actions. In the registry of Section 3, the acquired procedures form an initial segment of each chain, $C(a, b) = \{v_0, \ldots, v_{a-1}\} \cup \{v_4, \ldots, v_{3+b}\}$, so H_t is summarised by one of twenty states (a, b) with $a \leq 4$ and $b \leq 3$. SELF acquires v_a and ANCHOR acquires v_{4+b}. With o_{a-1} and o_{3+b} the return indicators of the last procedure acquired on each chain, the feasible set of Eq. 14 becomes:

$$\begin{aligned} \textsc{stop} \in \mathcal{A}(H_t) &\iff a + b > 0, \qquad \textsc{self} \in \mathcal{A}(H_t) \iff a < 4 \ \wedge\ \big(a = 0 \ \vee\ o_{a-1} = 1\big), \\ \textsc{anchor} \in \mathcal{A}(H_t) &\iff b < 3 \ \wedge\ \big[(b = 0 \ \wedge\ n_{\mathrm{loc}} > 0) \ \vee\ (b > 0 \ \wedge\ o_{3+b} = 1)\big]. \end{aligned} \tag{16}$$

Fixed-schedule value. The tables behind V^{st}_λ in Eq. 15 are indexed by state, feasible-action mask, action and target state. Each cell contains the mean out-of-fold accuracy at the stopping state and the mean calls of the plan "take the action, run both chains up to the target, truncated where a chain fails, and stop". STOP has the current state as its only target. For a feasible-action mask that never occurs in training, the value is pooled over every training query that reaches the state. A cell that no training query reaches takes the value -1. Each plan is linear in λ, so a single pair of tables covers every cost weight. These plans are the fixed schedules of Proposition 4.

Offset and features. The out-of-fold scores come from three inner answer models, each with its own nested cross-fitting. The offset β_λ is cross-fitted over the same three folds, used at test time as the mean of the three per-fold tables and interpolated linearly between the seven cost weights. The input $\chi(H_t)$ of Eq. 15 consists of the top score, the top-two margin, the entropy, the sorted scores, the observed share, the mean pairwise distance, the largest agreement cluster, whether the top point is at a chain frontier, $a/4$, $b/3$ and $n_{\mathrm{loc}}/3$. Its scores are the probabilities $\sigma(s_i)$, the entropy is taken after normalising them to sum to one, and the sorted scores are zero-padded to $n = 7$. The cost weight enters as $\lambda/0.05$, which maps the grid to $[0, 4]$. $\mathbf{k}_{\mathrm{act}}$ consists of the one-hot action (3) and the attributes of the procedure it acquires (11): its one-hot identity, an anchor-chain indicator, its depth divided by 3, 0.7^{depth} and a chain-root indicator.

Training and consistency checks. Each round fits r_ϑ on the decision groups, one per query, state and cost weight, that have at least two feasible actions. Each step samples 1,365 whole groups, with about 3,500 state-action pairs and at most 4,095. On every test query of every fold, forced full acquisition in either chain order matched SM in scores, choice, stopping state and calls. The rule of Eq. 6 at $\lambda = 0$ also matched SM in choice, state and calls. For the first two test queries of every fold, each decision at every reachable state and cost weight was recomputed with every not-yet-acquired output masked as missing, and the scores and action values were unchanged.

D.3 Model Selection and Compute

Hyperparameters. The weight $\gamma \in \{0, 0.1\}$ of the partial-history term and the number of epochs, from $\{10, 20, 40, 80\}$, were chosen by three-fold image-grouped cross-validation on one training split at the 50% fraction, by out-of-fold accuracy averaged over the three backbones. They are shared by all folds.

Compute. Training the answer models, tree priors and policies of the 108 folds and of the 15 held-out-benchmark folds took at most 1,706 s of wall-clock time on 32 single-threaded CPU processes, that is, at most 15.2 CPU-hours. Training boosted trees on the 108 folds took at most 5.8 s of wall-clock time on 16 such processes. Backbone outputs, prefill states and embeddings were computed beforehand. Costs in this paper are backbone calls. The image-text encoders of the tree prior, the landmark extractor and RoboRefer's per-view depth estimator are not backbone calls.

E Procedure Registry and Baselines

This appendix describes what is needed to reproduce the registry of Section 3 (Table 7). A view of an image I of $W_I \times H_I$ pixels is fixed by a centre $q \in [0, 1]^2$ and a window $\omega \in (0, 1]$. The crop size and the mapping of a returned point back to the frame of I are given as:

$$s_x = \max\big(\lfloor \omega W_I \rfloor, 224\big), \quad s_y = \max\big(\lfloor \omega H_I \rfloor, 224\big), \qquad z = \Big(\frac{x_1 + p_x s_x}{W_I}, \frac{y_1 + p_y s_y}{H_I}\Big), \tag{17}$$

where $(p_x, p_y) \in [0, 1]^2$ is the point returned on the crop and (x_1, y_1) is the top-left pixel of the crop. The crop is centred on q, shifted inside the image at the border, and passed to the backbone without resizing. The shortest image side in the set is 300 pixels, so every crop fits inside its image. The anchor-zoom chain starts from the located landmarks $\mathbf{m}_1, \ldots, \mathbf{m}_{n_{\mathrm{loc}}} \in [0, 1]^2$. For $n_{\mathrm{loc}} \geq 2$ its first view is given as:

$$q = \tfrac{1}{2}\big(\min_k \mathbf{m}_k + \max_k \mathbf{m}_k\big), \qquad \omega_0 = \mathrm{clip}\big(1.4\,\Delta,\, 0.3,\, 1\big), \tag{18}$$

where the minimum and the maximum are taken per coordinate and Δ is the longer side of the bounding box of the landmarks. For $n_{\mathrm{loc}} = 1$ the view is centred on the landmark with $\omega_0 = 0.5$.

Prompts, decoding and depth. Every call is decoded greedily. The RoboRefer backbones receive the instruction followed by a fixed sentence asking for normalised coordinates as `[(x1, y1)]`, and the first pair inside $[0, 1]^2$ is taken. Molmo-7B-D receives `Point to: {instruction}`, unless the instruction already begins with `point`, `please point` or `where`. Its first point is taken and divided by 100. A landmark is located by one call with `Please point out the {landmark}.` on the whole image. An answer without a valid coordinate gives no observation ($o_i = 0$), and its children are not acquired and incur no cost. RoboRefer also receives a depth map estimated on each view by Depth Anything V2 (Yang et al., 2024b), identically for every method.

Molmo-7B-D receives no depth map. Depth estimation is not counted as a call. On $4{,}236$ views of RR-2B it took 180 s, against $1{,}901$ s for the corresponding backbone calls. Cropping one full-image depth map instead of estimating depth per view does not restore the accuracy of the deepest self-zoom crop v_3 of RR-2B. The accuracy is 25.09% per view against 23.12% and 21.77% with the cropped map, re-normalised per crop or not, on 813 training queries of one fold.

Landmarks. The landmark phrases are extracted from the instruction text alone by Qwen2.5-1.5B-Instruct (Yang et al., 2024a), a text-only model that receives neither the image nor any annotation. In the evaluation they are extracted once per query and shared unchanged by every method that uses the registry. On 76 of the $1{,}440$ queries with an available anchor chain, a phrase does not occur verbatim in the instruction. Such phrases are inferred by the extractor, and no annotation is used. On RefSpatial the anchor chain is used only for the instructions that the benchmark annotates as multi-step. This disables the chain on 57 queries with an extracted phrase. On 53 of them, the phrases name the target or the viewer or are ill-formed. In all, n_{loc} is 0, 1, 2 and 3 on 186, 839, 531 and 70 queries.

Table 7: Procedure registry: views and costs. Each procedure pairs the unchanged instruction with one view of the image. The window is the fraction of each side of the original image that the view keeps, and the cost is counted in backbone calls. Here z_j is the point returned by procedure v_j, ω_0 the first window of the anchor-zoom chain, and n_{loc} the number of located landmarks of the query, each found by one call of its own.

	procedure	parent	centre of the view	window	cost
v_0	plain	–	whole image	1	1
v_1	self-zoom $\times 1$	v_0	z_0	$r = 0.7$	1
v_2	self-zoom $\times 2$	v_1	z_1	$r^2 = 0.49$	1
v_3	self-zoom $\times 3$	v_2	z_2	$r^3 = 0.343$	1
v_4	anchor-zoom $\times 1$	–	landmarks (Eq. 18)	ω_0	$1 + n_{\mathrm{loc}}$
v_5	anchor-zoom $\times 2$	v_4	z_4	$r\,\omega_0$	1
v_6	anchor-zoom $\times 3$	v_5	z_5	$r^2\omega_0$	1

Every baseline was run on the same checkpoints, queries and correctness rule as SM. We keep each external baseline at its basic published form and adapt only its input and output to pointing. Table 8 lists the calls counted for each. Every prompt sent to the backbone is one call, whether or not a coordinate can be read from its answer.

RegionFocus (Luo et al., 2025) keeps the official judge prompt, the marked point and the multi-crop stage of its public implementation. The judge, the same backbone, accepts or rejects the first point, and an accepted point is returned after two calls. Otherwise, the candidates from four crops around a focus point are numbered on the image and one is chosen by a last call. The judge accepts the first point on 88.3, 86.2 and 0.6% of the queries with RR-8B, RR-2B and Molmo-7B-D, so on Molmo-7B-D it almost always proceeds to the multi-crop stage. ZoomEye (Shen et al., 2025) follows its public tree search over image patches, guided by confidence queries for each visual cue. Its calls have a long tail from repeated cues, with medians of 13, 15 and 2 calls per query. As in its original design, the number of calls is not limited. The final answer is obtained by one pointing call on a composite image that shows the whole image and the selected patch side by side. In a synthetic test, the original two-image answering step was correct on 0 to 12 of 60 items on these backbones, against 57 to 59 with the composite image.

SC-10 (Wang et al., 2023), AC-10 (Aggarwal et al., 2023) and GUI-RC-10 (Du et al., 2026) aggregate ten samples of the plain call at $T = 0.3$ or $T = 0.6$. SC-10 returns the medoid of the most populated cell of a grid of width 0.05. AC-10 processes the samples in order and stops once a Beta posterior over the two leading cells puts at least 0.95 of its mass above one half, at 6.60 and 7.63 mean calls for the two temperatures. GUI-RC-10 votes on the overlap of 50-pixel squares around the points and returns the centre of the largest region of maximal vote. Over the registry, the registry vote applies the radius-0.05 vote of Appendix C to the observations without training. Boosted trees are a gradient-boosted selector over the observations' own features, a basic learned rule on the same observations. The HistGradientBoosting classifier of scikit-learn (Pedregosa et al., 2011) is fitted with its default settings and a fixed seed. Each observation in the full history of a training

query gives one training example, labelled by y_i. The $2n + 8$ features of an example, 22 for $n = 7$, describe its observation in the current query, namely the procedure (n), the observation mask (n), the point z_i (2) and the six agreement features of Appendix D.1. At test time, boosted trees return the observed point with the highest predicted probability. Both rules use the same observations and calls as SM. The accuracy margins of Table 3 do not depend on the call counts of any baseline.

Table 8: Calls counted per baseline. Every prompt sent to the backbone is one call, whatever its view and whether or not a coordinate can be read from its answer. Values are the mean number of calls per query over the 1,626 queries. Where a median and a maximum are given in parentheses, they are taken over the same queries. The last column is the mean over the three backbones.

method	calls counted	RR-8B	RR-2B	Molmo-7B-D	mean
plain call	one pointing call	1.00	1.00	1.00	1.00
RegionFocus	initial point, judgement, focus retries, four crop calls, aggregation	2.66 (2; 8)	2.83 (2; 8)	7.91 (8; 10)	4.47
ZoomEye	cue generation, every confidence query of the search, final pointing call	17.98 (13; 192)	33.71 (15; 4,148)	6.00 (2; 134)	19.23
SC-10 (T=0.3)	ten sampled pointing calls	10.00	10.00	10.00	10.00
SC-10 (T=0.6)	ten sampled pointing calls	10.00	10.00	10.00	10.00
AC-10 (T=0.3)	the same ten samples in order, until the stopping rule is met	5.77 (4; 10)	6.60 (4; 10)	7.42 (10; 10)	6.60
AC-10 (T=0.6)	the same ten samples in order, until the stopping rule is met	6.80 (7; 10)	7.65 (10; 10)	8.45 (10; 10)	7.63
GUI-RC-10 (T=0.3)	the same ten samples	10.00	10.00	10.00	10.00
GUI-RC-10 (T=0.6)	the same ten samples	10.00	10.00	10.00	10.00
SM, full registry	every feasible procedure, and one call per landmark phrase	7.83	7.95	7.96	7.91

F Evaluation Set

This appendix describes the evaluation set of Table 9. RoboRefIt refers to objects in clutter, and RefSpatial to objects and placements given by spatial relations. PointArena adds spatial, affordance and reasoning prompts, and RoboSpatial and Where2Place ask for free space related to an object. Two queries share an image when their decoded pixels are identical, which gives 1,139 distinct images for the 1,626 queries. These images are the groups of the splits and of the bootstrap (Appendix H). RefSpatial and Where2Place share 13 images, and no other pair of benchmarks shares

Table 9: Composition of the evaluation set and accuracy (%) of the plain call on each benchmark. Images with identical decoded pixels are counted once, so a count below the number of queries means that one image is paired with several instructions. RefSpatial and Where2Place share 13 images, so the image column sums to 1,152, while the set contains 1,139 distinct images. The column n_{loc}=0 gives the share of queries (%) on which the anchor chain is not available.

					plain call (%)		
benchmark	reference	queries	images	n_{loc}=0	RR-8B	RR-2B	Molmo-7B-D
RoboRefIt	Lu et al., 2023	700	518	14.3	80.00	72.14	81.57
RefSpatial	Zhou et al., 2025	440	228	13.0	60.68	49.09	46.82
PointArena	Cheng et al., 2025	264	264	8.0	71.59	65.91	76.14
RoboSpatial	Song et al., 2025	122	42	6.6	60.66	52.46	13.93
Where2Place	Yuan et al., 2025	100	100	0.0	66.00	66.00	31.00
all		1,626	1,139	11.4	71.09	63.04	63.10

one. RefSpatial combines RefSpatial-Bench and RefSpatial-Expand, and items with the same image and instruction are merged, which leaves 199 and 241 items. Where2Place is used in full. RoboSpatial contributes the 122 items of its context split, the only split with masks. From the 2,000 items of the RoboRefIt test split, 200 are taken at evenly spaced positions and 500 by selecting every ninth item of the remaining 1,800 in repeated passes, each starting one position after the previous one. RoboRefIt annotates boxes, so its admissible region is the annotated box filled in. PointArena contributes its three point-in-mask categories, spatial, affordance and reasoning, whose items are taken from each category in turn, in an order fixed before collection. We use the first 264 items of that order, 88 per category. The items are identical for all backbones, no item was excluded on the basis of its outcome, and their identifiers will be released with the code. The anchor chain is not available ($n_{\mathrm{loc}} = 0$) on 186 of the 1,626 queries. The last three columns of Table 9 give the accuracy of the plain call on each benchmark, and Table 14 gives that of every method (Appendix G).

G Additional Results

This appendix presents the full results behind Section 6.3, namely the cost-weight sweep, the budget of ASM, the results per backbone, per benchmark and on held-out benchmarks, and the coverage and selection of the rules that answer from the full registry.

Cost-weight sweep. Table 10 lists the operating point of ASM at each λ of the grid, with the utility J_λ of ASM, the plain call and SM. At $\lambda = 0$ ASM acquires every procedure that can be run and equals SM by construction (Section 5.3). As λ grows, ASM spends fewer calls, and from $\lambda = 0.1$ on it uses the plain call only. Its calls and accuracy never increase with λ, from 7.91 to 1.00 calls and from 70.71 to 65.74%, the order that Proposition 3 proves for exact maximisers of J_λ. Since SM and the plain call belong to its policy class, an exact maximiser is never below either in utility. ASM is above SM at every $\lambda > 0$, and at or above the plain call at every λ except $\lambda = 0.05$, where the difference is $100\Delta J_{0.05} = -0.02\ [-0.07, 0.04]$. SM always acquires the whole registry, and its utility falls below that of the plain call from $\lambda = 0.01$ on.

Table 10: Operating points of ASM over the cost weight λ and utility $100\,J_\lambda$ (accuracy in % minus 100λ times the mean calls, Eq. 2) of ASM, the plain call and SM on the nine-condition average. At λ=0 ASM acquires every observation and equals SM. J_λ is not bounded below, so the value of SM turns negative at large λ.

	ASM		$100\,J_\lambda$		
λ	calls	accuracy (%)	ASM	plain call	SM
0	7.91	70.71	70.71	65.74	70.71
0.003	4.99	70.20	68.70	65.44	68.34
0.01	2.93	69.05	66.12	64.74	62.80
0.02	1.81	67.73	64.11	63.74	54.88
0.05	1.02	65.81	60.73	60.74	31.14
0.1	1.00	65.74	55.74	55.74	−8.42
0.2	1.00	65.74	45.74	45.74	−87.56

Budget allocation. Table 11 presents descriptive statistics of the acquisitions in the 108 folds, aggregated like the accuracy (Appendix H). The count of acquired procedures on a query includes a last one that returned no point. The calls exceed this count because the first anchor-zoom step costs $c_4 = 1 + n_{\mathrm{loc}}$ calls. ASM stops early on queries where the plain call is reliable and acquires more where it is unreliable. At $\lambda = 0.01$ it stops after the plain call on 14.2% of the queries, where the plain call is correct 71.0% of the time. It spends five or more calls on 8.2%, where the plain call is correct 49.3% of the time. Its gain over the plain call is largest there. It starts the anchor chain on 11.8% of the queries, against 88.6% for SM, which starts it whenever it is available.

Per backbone and condition. Table 12 lists the accuracy and mean calls of each method on each backbone, averaged over the three training fractions. Table 13 gives the accuracy in each of the nine conditions. The training-free baselines use no training data, so their results do not depend on the fraction. SM is above every external baseline and boosted trees on each backbone and in each condition, including the 20% fraction, whose training splits have 325 or 326 queries. It is above the

Table 11: Budget allocation of SM and ASM, derived from the per-query acquisitions of the 108 image-grouped folds with the aggregation of the main results. Upper block: calls, acquired procedures and shares of queries (%). Lower block: queries grouped by the calls ASM spends at λ=0.01, with their share (%) and the accuracy (%) of the plain call, ASM at λ=0.01 and SM on them.

		ASM		
quantity	SM	λ=0.003	λ=0.01	λ=0.02
mean calls	7.91	4.99	2.93	1.81
procedures acquired	6.62	4.27	2.77	1.78
stops after the plain call (%)	0.06	2.16	14.20	46.80
started the self-zoom chain (%)	99.12	93.04	81.53	51.04
started the anchor chain (%)	88.56	50.29	11.81	2.26
five or more calls (%)	88.15	45.93	8.23	0.45
ASM at λ=0.01 spends	share	plain call	ASM	SM
stops after the plain call	14.20	70.96	70.96	75.02
2–4 calls	77.57	66.53	70.12	71.43
5 or more calls	8.23	49.34	55.65	56.53

registry vote on each backbone, by 0.04 points on RR-8B. In the RR-8B condition at 20%, it is 0.08 points below. ASM at $\lambda = 0.003$ is above boosted trees on each backbone with fewer calls. Pooled, it is 1.12 points $[0.74, 1.52]$ above them in accuracy and 2.00 $[1.62, 2.39]$ in $100\,\Delta J_{0.003}$, both with adjusted $p = 0.002$ (Appendix H). ASM at $\lambda = 0.01$ trades accuracy for a further reduction of calls. It is more accurate than the plain call in every condition with less than half the calls of SM. The sampling baselines are shown at $T = 0.3$.

Table 12: Mean calls per query and accuracy (%) on each backbone. Each value is the equal-weight mean over the three training fractions (20, 50 and 80%), so the mean of the three backbones is the nine-condition average of Table 3. Sampling baselines are shown at T=0.3.

	RR-8B		RR-2B		Molmo-7B-D	
method	calls	accuracy	calls	accuracy	calls	accuracy
plain call	1.00	71.09	1.00	63.04	1.00	63.10
RegionFocus	2.66	72.57	2.83	63.16	7.91	65.74
ZoomEye	17.98	69.86	33.71	59.29	6.00	61.44
SC-10 (T=0.3)	10.00	70.97	10.00	62.30	10.00	62.98
AC-10 (T=0.3)	5.77	71.09	6.60	62.42	7.42	62.79
GUI-RC-10 (T=0.3)	10.00	71.03	10.00	62.98	10.00	63.41
boosted trees	7.83	73.10	7.95	66.20	7.96	67.93
SM	7.83	74.77	7.95	68.29	7.96	69.07
ASM, λ=0.003	5.09	74.29	4.22	67.96	5.66	68.36
ASM, λ=0.01	2.45	72.60	2.91	67.39	3.45	67.16

Per benchmark. Table 14 lists the accuracy of each method on each of the five benchmarks, with every query aggregated over the nine conditions as in Table 3. On each benchmark, SM is above every training-free baseline, including the registry vote, and above boosted trees. ASM at $\lambda = 0.01$ is above every external baseline on RoboRefIt, RefSpatial, RoboSpatial and Where2Place. On PointArena, it equals RegionFocus and is 0.13 points below GUI-RC-10 at $T = 0.3$.

Held-out benchmarks. This protocol tests whether the margins of SM and ASM over the external baselines hold on a benchmark that is absent from their training data. For each backbone and each of the five benchmarks, SM and ASM are fitted on the queries of the other four benchmarks and tested on the held-out one, which gives 15 folds. Training queries that share an image with a held-out query are removed, 13 when RefSpatial is held out and 30 when Where2Place is held out. Each model is trained on 926 to $1{,}504$ queries. It uses the policy training of the main protocol and the hyperparameters of the answer model, including γ and the number of epochs chosen on a training

Table 13: Accuracy (%) in each of the nine conditions, with the mean calls of ASM. The plain call, RegionFocus (the strongest external baseline in every condition) and the registry vote use no training data, so they do not depend on the training fraction. SM and boosted trees make 7.83, 7.95 and 7.96 calls on the three backbones.

				registry	boosted		ASM, λ=0.003		ASM, λ=0.01	
backbone	train	plain call	RegionFocus	vote	trees	**SM**	accuracy	calls	accuracy	calls
RR-8B	20%	71.09	72.57	74.72	72.68	74.65	73.91	5.41	72.73	2.70
	50%	71.09	72.57	74.72	73.04	74.74	74.29	4.77	72.28	2.18
	80%	71.09	72.57	74.72	73.58	74.91	74.66	5.09	72.80	2.45
RR-2B	20%	63.04	63.16	65.56	65.36	67.75	67.50	4.62	66.57	3.22
	50%	63.04	63.16	65.56	66.32	68.63	67.94	4.31	67.12	2.46
	80%	63.04	63.16	65.56	66.93	68.49	68.43	3.74	68.49	3.05
Molmo-7B-D	20%	63.10	65.74	67.40	67.38	68.48	67.30	5.27	65.63	3.15
	50%	63.10	65.74	67.40	67.94	69.39	68.84	6.12	68.16	3.75
	80%	63.10	65.74	67.40	68.47	69.33	68.94	5.61	67.67	3.45

Table 14: Accuracy (%) on each benchmark of the evaluation set. Each value is the mean, over the queries of one benchmark, of the per-query outcomes averaged over the nine conditions as in Table 3. The last column is the nine-condition average over all queries. Sampling baselines are shown at T=0.3.

method	RoboRefIt (700)	RefSpatial (440)	PointArena (264)	RoboSpatial (122)	Where2Place (100)	all (1,626)
external training-free baselines						
plain call	77.90	52.20	71.21	42.35	54.33	65.74
RegionFocus	79.57	53.18	71.84	44.54	57.00	67.16
ZoomEye	76.00	50.45	66.79	41.26	52.33	63.53
SC-10 (T=0.3)	77.24	51.74	70.58	44.81	54.33	65.42
AC-10 (T=0.3)	77.24	51.52	70.33	45.90	55.00	65.44
GUI-RC-10 (T=0.3)	77.90	51.06	71.97	43.99	56.33	65.81
training-free rule over the registry						
registry vote (radius 0.05)	80.86	57.80	69.95	50.27	59.33	69.23
fitted on the registry						
boosted trees	81.12	56.25	71.30	50.48	58.11	69.08
ours, fitted on the registry						
SM	81.83	59.12	73.44	50.87	60.84	70.71
ASM, λ=0.003	81.60	58.33	72.74	50.41	60.04	70.20
ASM, λ=0.01	80.83	56.54	71.84	48.93	58.83	69.05

split of the pooled set (Appendix D). Each query is tested once per backbone. Its outcomes are averaged over the three backbones before the margins are computed as in Appendix H.

Table 15 shows that SM reaches 70.60% on held-out benchmarks, 0.11 points below its accuracy in Table 3, and is above every external baseline with every margin significant. ASM at $\lambda = 0.01$ reaches 69.29% at 3.49 calls. Its utility $J_{0.01}$ is significantly above that of every external baseline. As in Table 3, its smallest utility margin is over the plain call, $+1.06$ $[0.17, 1.96]$ with adjusted $p = 0.021$.

Coverage and selection. Lemma 1 splits the accuracy of any rule that returns an observed point into two factors, the coverage (the share of queries with a correct observation) and the selection rate (the accuracy on those covered queries). Table 16 lists both factors for the four rules that answer from the full registry. Their coverage is the same, 81.39% pooled. On the remaining 16 to 20% of the queries, depending on the backbone, no observation is correct. Every rule is scored on every query, so these queries count as errors for all four. The four rules differ only in selection. SM has the highest selection rate of the four on every backbone, 86.88% pooled, against 85.06% for the registry vote, 84.88% for boosted trees and 80.78% for the plain call.

Table 15: Accuracy, mean calls and pooled margins with each benchmark held out from training. SM and ASM are fitted on the other four benchmarks (15 folds), and each query's outcome is averaged over the three backbones. The external baselines use no training data, so their accuracy and calls equal those of Table 3. Adjusted p is 0.021 for ASM over the plain call and 0.002 for every other margin (Holm, or Bonferroni for the †AC-10 utility, which uses the per-query calls of AC-10, Appendix H).

method	calls	accuracy (%)	SM − method accuracy points [95% CI]	$\text{ASM}_{\lambda=0.01}$ − method utility margin [95% CI]
external training-free baselines				
plain call	1.00	65.74	+4.86 [3.84, 5.88]	+1.06 [0.17, 1.96]
RegionFocus	4.47	67.16	+3.44 [2.43, 4.46]	+3.11 [2.19, 4.09]
ZoomEye	19.23	63.53	+7.07 [5.96, 8.19]	+21.50 [19.25, 24.20]
SC-10 (T=0.3)	10.00	65.42	+5.19 [4.13, 6.25]	+10.39 [9.41, 11.38]
SC-10 (T=0.6)	10.00	65.01	+5.60 [4.52, 6.72]	+10.80 [9.77, 11.84]
AC-10 (T=0.3)	6.60	65.44	+5.17 [4.10, 6.26]	+6.96 [5.96, 7.98]†
AC-10 (T=0.6)	7.63	64.99	+5.62 [4.52, 6.76]	+8.45 [7.40, 9.52]†
GUI-RC-10 (T=0.3)	10.00	65.81	+4.80 [3.79, 5.84]	+10.00 [9.06, 10.98]
GUI-RC-10 (T=0.6)	10.00	65.56	+5.04 [3.96, 6.16]	+10.24 [9.22, 11.29]
ours, fitted on the other four benchmarks				
SM	7.91	70.60	–	–
ASM, λ=0.003	5.45	69.97	–	–
ASM, λ=0.01	3.49	69.29	–	–

Table 16: Coverage and selection (Lemma 1) for the rules that answer from the full registry. Coverage is the share of queries with a correct observation, and the selection rate is accuracy divided by coverage. The gap closed is the share of the gap between the plain call and the coverage. All values in %. Per-backbone accuracies are averaged over the three training fractions, and the coverage does not depend on the fraction.

	all	RR-8B	RR-2B	Molmo-7B-D
coverage	81.39	84.19	80.32	79.64
selection rate				
plain call	80.78	84.44	78.48	79.23
registry vote	85.06	88.75	81.62	84.63
boosted trees	84.88	86.82	82.43	85.29
SM	86.88	88.80	85.03	86.72
gap closed from the plain call to the coverage				
registry vote	22.28	27.70	14.59	26.02
boosted trees	21.31	15.31	18.32	29.19
SM	31.74	28.03	30.41	36.08

Remaining gap. Since the coverage is common to the four rules, the share of the gap between the plain call and the coverage that a rule closes equals the relative reduction of its selection error on covered queries. Without training, the registry vote closes 22.3% of this gap. Boosted trees close 21.3% and SM 31.7%, the largest share on every backbone. By Lemma 1, the remaining gap of SM splits into its distance to the Bayes selector over the same histories and the distance of that selector to the coverage. The second part is positive if and only if, with positive probability, a history is more likely to contain a correct observation than any one of its observations is to be correct, $\max_{i \in \mathcal{O}(H)} p_i(H) < \mathbb{P}(\bar{y} = 1 \mid H)$. The coverage thus bounds the attainable accuracy but is not attainable in general. Neither bound of Lemma 1 falls when a procedure is added to the registry, since every full history then contains the current one (Assumption 3 and Appendix B).

H STATISTICAL PROTOCOL

This appendix states how the margins, intervals and adjusted p-values are computed, and how the margins change without the RefSpatial restriction.

Splits. Queries are grouped by their exact decoded image, which gives $1{,}139$ groups over the $1{,}626$ queries. Each of the three random image-grouped splits partitions the groups into five size-balanced parts at training fractions of 20% and 80%, and into two at 50%. At 20% a fold trains on one part and tests on the other four, so each query is tested four times per split. At 50% the two folds swap roles, and at 80% a fold trains on four parts and tests on one. Every query is thus tested in every condition. Inner folds divide a training split into three image-grouped parts.

Aggregation. A query's outcome under one method and condition is averaged over its test appearances within a split, then over the three splits. These values are averaged with equal weight over the nine backbone $\times$ fraction conditions and then over queries. A margin is the mean of the per-query differences, in accuracy or in J_λ. Folds and splits are never treated as independent samples.

Intervals and tests. The 95% intervals are percentile intervals from $10{,}000$ cluster-bootstrap resamples (Efron, 1979) over the $1{,}139$ images. The two-sided p-values come from $10{,}000$ paired sign flips in which all queries of an image share one sign, so the smallest attainable value is $1/10{,}001$. The main contrasts are SM against the nine external settings and boosted trees in accuracy, and ASM at $\lambda = 0.01$ against the same ten in $J_{0.01}$. Each of the two families contains twenty contrasts, these ten and the corresponding ten on held-out benchmarks. Table 15 lists those against the external baselines. Their p-values are adjusted by Holm's correction (Holm, 1979) within the family. The utility contrasts against AC-10, on both protocols, are computed with the per-query calls of AC-10 and adjusted by the Bonferroni correction at the same family size (†). The two contrasts against the registry vote, SM in accuracy and ASM in $J_{0.01}$ (‡), and the two of ASM at $\lambda = 0.003$ against boosted trees, in accuracy and in $J_{0.003}$, are computed by the same bootstrap and sign flips and adjusted by the Bonferroni correction at the same family size. The vote contrasts use the per-query values of the registry vote, which do not depend on the training fraction and whose calls are those of SM. At a family size of twenty, the smallest attainable adjusted p is $20/10{,}001 \approx 0.002$ under either correction. Every contrast of Table 3 and both contrasts of ASM at $\lambda = 0.003$ attain it, and so does every contrast of Table 15 except ASM over the plain call, whose adjusted p is 0.021. A contrast is called significant when its adjusted p is below 0.05 and its interval lies above zero.

Sensitivity to the RefSpatial restriction. Table 17 repeats the margins of SM and ASM over the plain call, RegionFocus, the registry vote and boosted trees without the 57 RefSpatial queries on which the multi-step annotation disables the anchor chain, and without RefSpatial. On the other $1{,}569$ queries, the anchor chain is available if and only if a landmark phrase is extracted. Every margin remains positive, with an interval above zero.

Table 17: Margins of Table 3 on all queries, without the 57 RefSpatial queries on which the multi-step annotation disables the anchor chain, and without RefSpatial. Computed from the per-query outcomes with the image-cluster bootstrap of Appendix H ($10{,}000$ resamples). The intervals on the subsets are descriptive.

margin [95% CI]	all queries (1,626)	without the 57 restricted (1,569)	without RefSpatial (1,186)
SM − method, accuracy points			
plain call	+4.97 [4.00, 5.93]	+5.00 [4.03, 5.96]	+4.24 [3.19, 5.30]
RegionFocus	+3.55 [2.61, 4.49]	+3.53 [2.59, 4.48]	+2.67 [1.60, 3.74]
registry vote	+1.48 [0.88, 2.08]	+1.62 [1.03, 2.22]	+1.54 [0.91, 2.16]
boosted trees	+1.63 [1.25, 2.02]	+1.66 [1.28, 2.05]	+1.17 [0.78, 1.57]
ASM$_{\lambda=0.01}$ − method, $100\,\Delta J_{0.01}$			
plain call	+1.37 [0.67, 2.09]	+1.36 [0.63, 2.08]	+1.03 [0.27, 1.82]
RegionFocus	+3.42 [2.65, 4.23]	+3.36 [2.55, 4.17]	+2.98 [2.07, 3.90]
registry vote	+4.80 [4.05, 5.57]	+5.03 [4.29, 5.79]	+5.31 [4.53, 6.12]
boosted trees	+4.95 [4.44, 5.47]	+5.08 [4.57, 5.60]	+4.94 [4.40, 5.49]