U-TRACE: UNCERTAINTY-GUIDED ADAPTIVE EVIDENCE ACQUISITION FOR OPEN-DOMAIN VQA

Anonymous authors
Paper under double-blind review

ABSTRACT

Open-domain visual question answering requires systems to answer questions that may involve visual understanding, external knowledge, or multimodal reasoning. Existing approaches typically rely on fixed pipelines and single-pass generation, such as retrieval or visual grounding, without explicitly assessing whether the current answer is reliable or which intervention would improve it. To address these limitations, we introduce U-TRACE, an uncertainty-guided framework for adaptive evidence acquisition in open-domain VQA. U-TRACE probes the vision–language model under complementary visual and textual contexts to construct a model-specific behavioral state that captures semantic dispersion, context sensitivity and prediction stability. PROBEDETECTOR uses this state to estimate the reliability deficit of the initial prediction and determine whether further evidence acquisition is required. For selected instances, a history-conditioned agent chooses the next evidence-seeking action and specifies its request when necessary. The resulting report is added to the evidence history and used to update the uncertainty state, so that each subsequent decision reflects what has already been observed. To learn the adaptive policy, we propose execution-verified learning, which compares alternative next actions from the same evidence history and assigns supervision based on their impact on final-answer quality. Experiments on OK-VQA and A-OKVQA show that U-TRACE consistently improves its base VLMs and outperforms existing open-domain VQA methods.

1 INTRODUCTION

Open-domain visual question answering (VQA) requires a system to connect visual observations with the knowledge and reasoning needed to answer a question. Earlier approaches retrieved external information and combined it with image features and question representations (Marino et al., 2019; Lin & Byrne, 2022). More recent systems use vision language models (VLMs) and large language models (LLMs), which supplemented by external modules or multiple stages of inference (Lu et al., 2023; Hu et al., 2023b). Although existing systems can address a broad range of questions by enhancing a single capability, the information needed for a reliable answer could differ across individual inputs.

Existing methods may augment a model with additional sources of evidence, including retrieved passages, question-guided captions, and visually grounded regions (Lin et al., 2023; Hu et al., 2023a; Liang et al., 2023). Whether an intervention helps depends on the initial prediction and the evidence the model has already considered. For example, a retrieved passage can resolve a question about world knowledge but may also introduce an irrelevant association. Applying a source without considering its likely effect can therefore replace a supported answer with a less reliable one. Figure 1 illustrates two questions in which different sources provide the decisive evidence. The first question benefits from external knowledge, whereas the second one requires grounding the answer in the depicted shape.

Additionally, when the evidence is incomplete or conflicting, existing models can still produce plausible answers, but the reliability of these answers is often unclear (Gurari et al., 2018; Guan et al., 2024; Khan & Fu, 2024). As a solution, uncertainty measurements can help identify unreliable predictions (Kadavath et al., 2022; Kuhn et al., 2023; Farquhar et al., 2024; Hou et al., 2023). In downstream task such as VQA, however, answer reliability and the usefulness of a particular

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

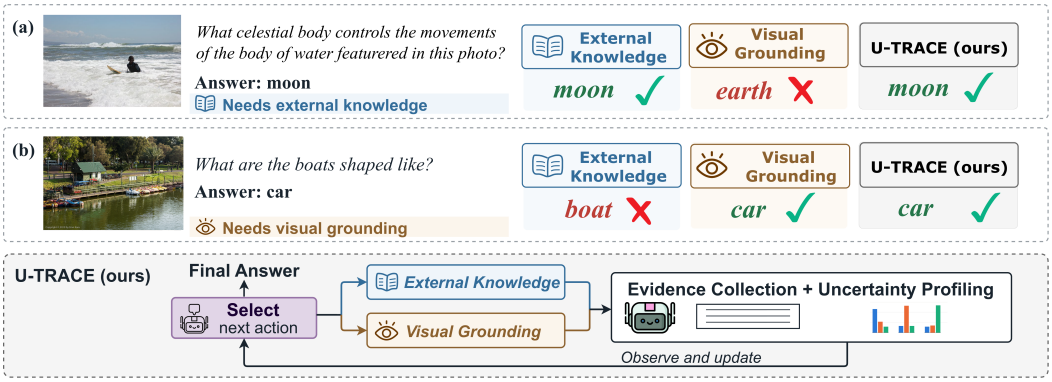


Figure 1: Complementary examples of evidence selection in open-domain VQA. In (a), we augment Qwen2.5-VL with retrieved external knowledge, which yields *moon* for the water question; visual grounding instead yields *earth*. In (b), we augment Qwen2.5-VL with a caption to provide visual grounding, which yields *car*; external knowledge instead yields *boat*. The lower panel shows that U-TRACE selects retrieval to obtain external knowledge in (a) and captioning to provide visual grounding in (b), which produces the correct answer in both cases.

source are distinct quantities. Calibration and abstention methods can assess whether a prediction is trustworthy (Dancette et al., 2023; Khan & Fu, 2024), but an uncertain answer does not indicate whether a skill will be useful.

To address these gaps, we introduce U-TRACE, which first probes a VLM under complementary visual and textual contexts to characterize the stability and grounding of its initial answer. PROBEDETECTOR uses this behavioral information to identify cases that warrant further evidence acquisition. On those cases, ROLEAGENT selects an available source, forms a specific request, and incorporates the returned report into an ordered evidence history. HISTORYUQ updates the uncertainty state after each report so the next decision reflects what was learned. The agent may continue with another request or produce a final answer from the information available. The agent is trained through execution-verified learning. For a collected training history, a policy evaluates stopping and alternative tool actions under their subsequent continuations. The observed effect on final answer quality supervises tool choice, request formation, and answer integration. This procedure learns which information resolves a question without requiring a categorical diagnosis of the initial uncertainty. Our contributions are as follows:

- We study uncertainty in open-domain VQA and show that identifying an unreliable initial answer is insufficient for collecting the evidence that will improve it. Build on this motivation, We build propose U-TRACE, which combines behavioral uncertainty with adaptive evidence acquisition.
- We propose execution-verified learning for training the sequential evidence policy from downstream VQA outcomes. By comparing executed continuations from the same evidence history, it directly learns source selection, request formation, and answer integration for the target task.
- Experiments across multiple VLMs and open-domain VQA benchmarks show consistent improvements over baselines.

2 RELATED WORK

Open-Domain Visual Question Answering. Knowledge based VQA combines images and questions with retrieved or parametric knowledge (Gui et al., 2022; Lin & Byrne, 2022; Lin et al., 2023; 2024). Other approaches supply captions, candidate answers, or visual clues to language models (Yang et al., 2022; Hu et al., 2023a; Shao et al., 2023; Wang & Ge, 2024). TOA directs visual verification from an LLM hypothesis (Liang et al., 2023). These methods emphasize particular evidence sources.

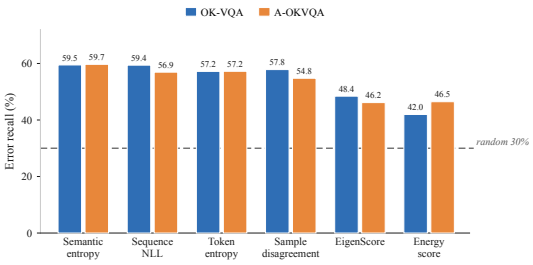


Figure 2: **Behavioral uncertainty signals identify initial errors.** Error recall for six signals on OK-VQA and A-OKVQA. The dashed line marks random selection at 30% coverage. Semantic entropy has the highest recall across the two benchmarks.



Figure 3: **Single interventions at 30% coverage.** Each intervention is applied to the 30% of examples with the highest initial semantic entropy. Δ is the mean VQA score change in points.

Agentic System for Visual Question Answering. Agents can interleave reasoning and tool use, including visual actions (Yao et al., 2022; Schick et al., 2023; Surís et al., 2023; Hu et al., 2023b). Adaptive retrieval uses confidence, reflective tokens, or learned search policies (Jiang et al., 2023; Cocchi et al., 2025; Chen et al., 2026; Wu et al., 2026). Uncertainty estimators use self evaluation, semantic disagreement, or prompt sensitivity to assess prediction reliability (Kadavath et al., 2022; Kuhn et al., 2023; Farquhar et al., 2024; Hou et al., 2023).

3 Preliminaries

3.1 Problem Setting

In open-domain VQA, we denote an input as $x = (I, q)$, which pairs an image with a question that may or may not additional evidence to derive the final answer (Marino et al., 2019; Schwenk et al., 2022). A VLM $\mathcal{M}_\theta$ gives the initial answer $z_0 = \mathcal{M}_\theta(I, q)$.

Answering with evidence. Prior methods use visual clues, candidate answers, retrieved passages, image regions, or tool outputs as additional evidence (Yang et al., 2022; Hu et al., 2023a; Guo et al., 2023; Wang & Ge, 2024; Shao et al., 2023; Lin & Byrne, 2022; Lin et al., 2023; Hao et al., 2024; Lin et al., 2022; Khayatkhoei et al., 2025; Surís et al., 2023; Hu et al., 2023b; Liang et al., 2023). We represent an evidence source $j \in \mathcal{J}$ by a request $a = (j, \omega)$ with source-specific argument $\omega \in \Omega_j$ and response

$$o \sim P_j(\cdot \mid x, E, \omega), \tag{1}$$

where $E = ((a_1, o_1), \ldots, (a_n, o_n))$ records the ordered, attributed evidence already held. The answer conditioned on E is

$$z(E) = \mathcal{M}_\theta(I, q; E), \qquad z_0 = z(\varnothing). \tag{2}$$

Evidence acquisition. Let $R(z; x) \in [0, 1]$ be the VQA score, with reference answers used only for evaluation. Fixed-request methods acquire the same evidence for every input, whereas adaptive methods may use confidence or self assessment to decide when to acquire it (Jiang et al., 2023; Cocchi et al., 2025; Chen et al., 2026). We study when to request evidence, which source to query, and when to stop, using only observations available at decision time.

3.2 Empirical Study

We study Qwen2.5-VL-7B (Bai et al., 2025) on OK-VQA and A-OKVQA. This diagnostic uses a greedy initial answer. Section 5 describes the experimental baseline, which uses random few-shot examples. Let u_0 summarize answer likelihood, sample agreement, and stability under controlled variations (Kadavath et al., 2022; Kuhn et al., 2023; Farquhar et al., 2024; Hou et al., 2023). Because u_0 is computed before acquisition, it cannot contain an unseen response. Initial reliability is

$$\rho(u) = \Pr(R(z_0; x) \geq \gamma \mid u_0 = u)\,, \tag{3}$$

with $\gamma = 0.5$. For each intervention j, let z_j be its resulting answer when applied to the initial input. Protocols and complete results are in Appendix A.

Uncertainty ranks initial reliability. As shown in Figure 2, behavioral signals rank initial errors above chance. The combined estimator reaches 0.82 AUROC on OK-VQA and 0.80 on A-OKVQA, while semantic entropy reaches 0.78 and 0.79. They indicate initial reliability, but do not identify a useful source.

Evidence utility depends on the instance. Visual inspection and retrieval repair different subsets of errors. The repairing set $\mathcal{J}^+(x) = \{j \mid R(z_0; x) < \gamma \leq R(z_j; x)\}$ therefore varies by input. Appendix A.4 shows that uncertainty components only weakly predict this set. Selecting useful evidence matters more than identifying the underlying cause of uncertainty.

Reduced uncertainty need not improve the answer. Across 4,025 report insertions, the rank correlation between changes in uncertainty and VQA score is -0.137. Uncertainty decreases in 42.9% of updates that replace a correct answer with an incorrect one. A lower uncertainty value therefore does not establish that a report helps the current history, as analyzed further in Appendix D.7.

Evidence has a net effect. Evidence can repair an error or overturn a correct answer. With $e_j = \mathbf{1}[R(z_j; x) < \gamma]$ ($j = 0$ for the initial answer), $p_j^+(u) = \Pr(e_j = 0 \mid e_0 = 1, u_0 = u)$, and $p_j^-(u) = \Pr(e_j = 1 \mid e_0 = 0, u_0 = u)$,

$$\mathbb{E}[e_0 - e_j \mid u_0 = u] = [1 - \rho(u)]\, p_j^+(u) - \rho(u)\, p_j^-(u). \tag{4}$$

Reliability alone cannot determine the gain because repair and corruption rates also matter. Figure 3 shows that applying a single tool to the 30% most uncertain examples lowers overall VQA score in eleven of twelve cases. Uniform application lowers it in every case (Appendix Table 6). Thus, an uncertainty gate alone does not establish which intervention will help.

These findings motivate separate reliability and request decisions, subsequent requests conditioned on observed responses, and an explicit option to retain z_0.

4 METHOD

Figure 4 illustrates how U-TRACE uses behavioral uncertainty to route open domain VQA questions. PROBESET and U-STATE characterize the initial answer, and PROBEDETECTOR flags unreliable cases. On flagged inputs, ROLEAGENT selects sources and requests, observes their responses, and may stop at any step. HISTORYUQ updates uncertainty under the acquired evidence. On unflagged inputs, its answer role uses the probe information without requesting a tool. Both branches use the same final rephrasing step. Training compares stopping with executed continuations that may use several tools.

4.1 BEHAVIORAL UNCERTAINTY AND RELIABILITY-BASED ROUTING

The initial prediction alone gives little indication of whether another prompt would produce the same answer or whether the image matters to that answer. PROBESET therefore queries the VLM under controlled contexts $\mathcal{C}_0(x)$ without acquired evidence and records z_0 and a behavioral cache

$$\mathcal{B}(x) = \{(c_m, z_{mk}) : c_m \in \mathcal{C}_0(x),\ k = 1, \ldots, K_m\}. \tag{5}$$

Let p_m group meaning-equivalent answers sampled under sighted context c_m (Kuhn et al., 2023; Farquhar et al., 2024), and let $\bar{p} = M^{-1} \sum_{m=1}^{M} p_m$. U-STATE computes

$$U_{\text{within}} = M^{-1} \sum_{m=1}^{M} H(p_m), \tag{6}$$

$$U_{\text{between}} = H(\bar{p}) - M^{-1} \sum_{m=1}^{M} H(p_m), \tag{7}$$

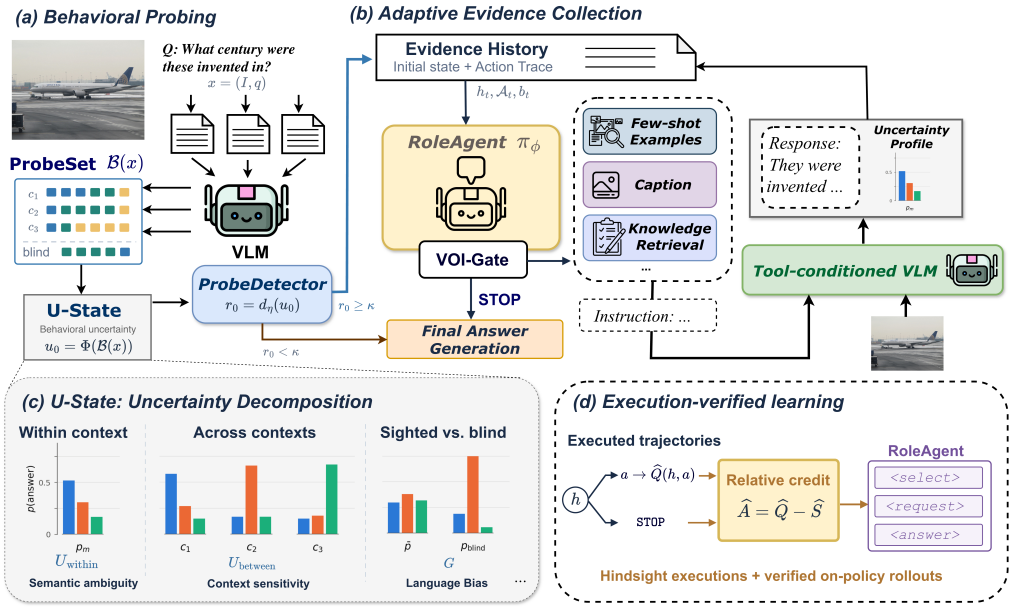


Figure 4: **Overview of U-TRACE.** (a) PROBESET applies complementary prompts to characterize the initial VLM answer, and PROBEDETECTOR uses the resulting behavioral state for reliability routing. (b) For routed inputs, ROLEAGENT conditions on the evolving evidence history to choose a request or stop. Its `answer` role produces the terminal answer on both routing branches, followed by final rephrasing. (c) U-STATE summarizes ambiguity within each probe context, variation across contexts, and dependence on visual input. (d) Execution-verified learning compares alternative executed continuations to supervise source selection, request formation, and answer integration.

for ambiguity within a context and sensitivity to context (Hou et al., 2023). Blind-image predictions remain outside this mixture. Their agreement with sighted answers and $G = D_{\mathrm{KL}}(\bar{p} \| p_{\text{blind}})$ measure grounding. These and stability statistics form $u_0 = \Phi(\mathcal{B}(x))$.

PROBEDETECTOR estimates the reliability deficit $r_0 = d_\eta(u_0) \approx 1 - \rho(u_0)$ from Equation 3. The notation $d_\eta(u_0)$ includes the detector's access to textual responses in the probe cache alongside the numerical state. Inputs with $r_0 \geq \kappa$ enter the agent's evidence acquisition loop. Otherwise, the `answer` role uses the question, initial prediction, and probe outputs to answer without a tool request. Routing identifies risky answers, while source utility requires a separate decision.

4.2 ADAPTIVE EVIDENCE ACQUISITION

History and uncertainty under evidence. At step t, define

$$E_t = ((a_0, o_1), \ldots, (a_{t-1}, o_t)), \qquad h_t = (q, z_0, \mathcal{B}(x), u_0, r_0, E_t), \tag{8}$$

with $E_0 = \varnothing$. A fixed HISTORYUQ assessor samples answers from a rendering $\varrho(h_t)$ and summarizes their semantic distribution $\hat{p}_t$ as

$$u_t^{\text{hist}} = \Psi(\hat{p}_t). \tag{9}$$

This measures uncertainty under E_t, whereas u_0 precedes acquisition. The policy observes

$$s_t = \text{Serialize}(h_t, u_t^{\text{hist}}, \mathcal{A}_t, b_t), \tag{10}$$

where $b_t = H - t$ is the remaining request budget. Ordered responses remain in s_t because uncertainty alone cannot represent their content.

Actions and responses. At each step the agent selects

$$a_t \in \mathcal{A}_t = \{\texttt{STOP}\} \cup \{(j, \omega) : j \in \mathcal{J}_t,\ \omega \in \Omega_j(h_t)\}, \tag{11}$$

where STOP invokes terminal answerer D_ϕ and $\mathcal{J}_t$ contains available sources. A request returns $o_{t+1} \sim P_j(\cdot \mid x, h_t, \omega)$ as defined in Equation 1, and $E_{t+1} = E_t \oplus (a_t, o_{t+1})$. For VLM-mediated sources, TOOLVLM returns

$$o = (\tilde{z},\ \xi,\ \nu), \tag{12}$$

an answer, abstention indicator, and sample dispersion. The terminal answer $D_\phi(h_T)$ integrates attributed responses and may retain z_0. A trajectory contains at most H requests.

Objective and value of a request. For optional request cost $c(h_t, a_t) \geq 0$ and weight $\lambda \geq 0$, the ideal policy maximizes

$$J(\pi; D_\phi) = \mathbb{E}_{x,\tau\sim\pi}\left[R(D_\phi(h_T); x) - \lambda \sum_{t=0}^{T-1} c(h_t, a_t)\right]. \tag{13}$$

Let $S(h) = \mathbb{E}[R(D_\phi(h); x) \mid h]$ be the stopping value and $h' = h \oplus (a, o)$. With one request remaining, the advantage is

$$A_1(h, a) = -\lambda c(h, a) + \mathbb{E}_{o|h,a}[S(h')] - S(h). \tag{14}$$

This value depends on the request and current history, unlike initial reliability. With b requests remaining, the optimal values satisfy

$$V_0^*(h) = S(h), \tag{15}$$

$$Q_b^*(h, a) = -\lambda c(h, a) + \mathbb{E}_{o|h,a}[V_{b-1}^*(h \oplus (a, o))], \tag{16}$$

$$V_b^*(h) = \max\{S(h),\ \sup_{a\neq\texttt{STOP}} Q_b^*(h, a)\}, \tag{17}$$

The corresponding stopping rule is

$$\sup_{a\neq\texttt{STOP}} A_b^*(h, a) > 0, \qquad A_b^*(h, a) = Q_b^*(h, a) - S(h). \tag{18}$$

We call this answer or request comparison the value of information gate (VOI GATE). At inference, ROLEAGENT implements a learned version through its choice between STOPand a tool request. It does not explicitly compute the ideal expectations in Equation 17. The VOI GATE is distinct from the initial reliability routing decision made by PROBEDETECTOR. The difference $A_b^*(h, a) - A_1(h, a)$ is the option value of later requests, as derived in Appendix B.4. These are ideal expectations, whereas our training targets are realized VQA scores from executed continuations. The agent therefore learns from request outcomes rather than treating reduced uncertainty as reward. Appendix B.5 analyzes the value of conditioning later decisions on observed reports.

4.3 ROLE-CONDITIONED DECISION POLICY

The agent must decide what to inspect and how to use what it finds. Giving these decisions a shared model lets the answer role see the same history that motivated the request. One shared ROLEAGENT model parameterizes three roles through role-specific state renderings

$$\pi_\phi^f(y \mid s) = p_\phi(y \mid \mathrm{Render}_f(s)). \tag{19}$$

`select` chooses a source or STOP, `request` forms a request when needed, and `answer` integrates evidence into $D_\phi(h_T)$. HISTORYUQ remains fixed during policy learning. For either routing branch, the terminal answer uses all information available on that branch. A final language model rephrases it as a concise answer in the dataset's expected style. This final rendering does not request a new tool.

4.4 EXECUTION VERIFIED POLICY LEARNING

All supervision comes from training questions, with validation reserved for reporting. We initialize a chooser from probe candidates and successful trajectories, then cross fit it over collected training histories. At each history, a chooser trained on other folds is frozen. Its answer on stopping has VQA score $\widehat{S}(h)$. For each available tool a, we execute one request, append its report, and let the same chooser continue for up to the remaining horizon. Its final answer has score $\widehat{Q}(h, a)$, giving

$$\widehat{A}(h, a) = \widehat{Q}(h, a) - \widehat{S}(h). \tag{20}$$

Table 1: Detector comparison. Every detector receives the same behavioral signals. PROBEDETECTOR also reads textual probe outputs. AUROC measures error ranking independent of threshold, while precision and recall use the reported routing operating point.

	OK-VQA			A-OKVQA		
Detector	AUROC	Precision	Recall	AUROC	Precision	Recall
MLP	0.826	0.589	0.723	0.852	0.719	0.729
FT-Transformer	0.824	0.592	0.727	0.856	0.719	0.729
Set Transformer	0.827	0.587	0.721	0.856	0.714	0.724
PROBEDETECTOR	**0.845**	**0.614**	**0.754**	**0.863**	**0.733**	**0.732**

Only the first tool is forced. The largest positive observed gain selects the tool target, or `STOP`if none helps. Beneficial requests and visited histories supply request and answer targets. These are realized full continuation values, not exact expectations. Gold answers score outcomes offline and never enter inference prompts. At inference, the agent can select several tools, revisiting the decision after each report for at most three steps.

For role f, let $\mathcal{I}_f = \{i \mid f_i = f\}$. Weighted tuples (s_i, f_i, y_i, w_i) train the roles jointly

$$\mathcal{L}(\phi) = -\sum_f \beta_f \frac{\sum_{i\in\mathcal{I}_f} w_i \log \pi_\phi^f(y_i \mid s_i)}{\sum_{i\in\mathcal{I}_f} w_i}, \tag{21}$$

Positive advantages raise request weights, capped to prevent a few large gains from dominating. Verified stopping targets also receive weight. Full data construction and training details are in Appendix B.2.

5 EXPERIMENTS

We evaluate the performance across base VLMs on open-domain VQA benchmarks. The appendix reports complementary analyses of tools, uncertainty states, and policy learning.

5.1 EXPERIMENTAL SETUP

We evaluate on OK-VQA (Marino et al., 2019) and A-OKVQA (Schwenk et al., 2022) using each benchmark's official direct answer VQA score. Appendix E.1 shows the data splits, answer and routing protocols, and implementation details.

5.2 MAIN RESULTS

Table 1 compares detectors under the shared numerical uncertainty signals. PROBEDETECTOR additionally receives sampled answers, a caption, and a rationale. It achieves the strongest error ranking and operating point precision on both benchmarks, with AUROC of 0.845 and 0.863. We therefore use PROBEDETECTOR throughout the remaining experiments. Appendix E.1 defines the input signals.

We show existing open-domain VQA results alongside matched base VLM comparisons in Table 2. The prior methods use different backbones and evaluation settings. In the controlled comparisons, U-TRACE improves every base VLM on both benchmarks, by 8.1 to 19.3 VQA points. With Qwen2.5-VL-7B, it reaches 70.5 on OK-VQA and 73.9 on A-OKVQA.

5.3 RESULTS ACROSS VLMS AND ROUTING BREAKDOWN

Table 3 separates results by the initial routing decision. We observe the largest gains on flagged inputs, where the final VQA score rises by 13.3 to 38.0 points. Unflagged inputs receive no tool calls, yet their scores increase by 0.1 to 5.1 points, as detailed in Appendix Table 11. For these inputs, the ROLEAGENT answer role uses the initial answer and probe outputs before final rephrasing. The gains therefore reflect both answer generation informed by the probe and rephrasing. Appendix D.1 reports detector results across base VLMs.

Table 2: Comparison with prior systems and matched base VLMs.

Method	Visual backbone	LLM backbone	OK-VQA	A-OKVQA
Open-domain VQA methods				
KAT (Gui et al., 2022)	CLIP	T5-large + GPT-3	54.4	-
PromptCap (Hu et al., 2023a)	PromptCap (OFA)	GPT-3	60.4	56.3
TOA (Liang et al., 2023)	TOA vision modules	GPT-3.5-turbo	60.6	61.2
RA-VQA-v2 (Lin et al., 2023)	BLIP-2	-	62.1	-
LION (Chen et al., 2024b)	LION vision modules	Flan-T5-XXL	57.3	60.9
QACap (Yang et al., 2025)	QACap (BLIP)	Claude 3.5	68.2	66.3
ViLA (Kim et al., 2026)	LLaVA-NeXT-13B	GPT-4	68.9	70.0
Base VLMs				
Base VLM	Qwen2.5-VL-7B	-	61.7	62.3
Base VLM	InternVL3.5-8B	-	60.8	62.8
Base VLM	GLM-4.1V-9B	-	49.5	59.0
Base VLM	Llama-3.2-11B-Vision	-	53.6	61.9
Base VLM	Molmo-7B-D	-	54.2	60.2
Ours				
U-TRACE	Qwen2.5-VL-7B	Llama-3.1-8B	**70.5**	**73.9**
U-TRACE	InternVL3.5-8B	Llama-3.1-8B	68.9	72.7
U-TRACE	GLM-4.1V-9B	Llama-3.1-8B	68.8	73.3
U-TRACE	Llama-3.2-11B-Vision	Llama-3.1-8B	69.5	72.8
U-TRACE	Molmo-7B-D	Llama-3.1-8B	66.2	71.6

Table 3: Performance of PROBEDETECTOR under different VLM backbones. Initial denotes the initial VLM answer and final denotes the answer produced by U-TRACE on the same subset, including ROLEAGENT answer generation and final rephrasing. Unflagged inputs receive no tool involved. Calls are tool calls per flagged input.

Data	Base VLM	Flagged initial → final	Unflagged initial → final	Calls
OK-VQA	Qwen2.5-VL-7B	34.5→49.6	81.4→83.9	1.2
	InternVL3.5-8B	29.0→44.6	82.0→83.4	1.2
	GLM-4.1V-9B	12.9→50.9	74.0→79.1	0.5
	Llama-3.2-11B-Vision	19.4→49.4	76.4→81.2	0.5
A-OKVQA	Qwen2.5-VL-7B	43.2→56.5	83.0→84.1	0.7
	InternVL3.5-8B	36.2→55.7	83.9→84.0	0.6
	GLM-4.1V-9B	27.9→59.3	79.7→82.6	0.7
	Llama-3.2-11B-Vision	32.8→56.4	83.0→83.8	0.6

5.4 ABLATION STUDY

Table 4 evaluates the contribution of each component. The complete framework improves on the initial model by 8.8 points on OK-VQA and 11.6 points on A-OKVQA. Without detector guided selection of training examples and inference inputs, the score falls by 4.8 and 5.6 points, respectively. Removing the learned ROLEAGENT functions reduces performance by 2.6 and 4.4 points. The evidence history and TOOLVLM adaptation also contribute on both benchmarks. Appendix D.6 further examines uncertainty states, while Appendix D.12 examines policy learning.

Table 4: Ablations on OK-VQA and A-OKVQA. Δ values give each ablation's change from the complete framework.

	OK-VQA		A-OKVQA	
Configuration	Score	Δ	Score	Δ
Reference configurations				
Initial model only	61.7	-	62.3	-
Complete U-TRACE	**70.5**	-	**73.9**	-
Function removed				
Without detector guided curation	65.7	−4.8	68.3	−5.6
Without evidence history	69.7	−0.8	72.5	−1.4
Without learned ROLEAGENT functions	67.9	−2.6	69.5	−4.4
Without tool-conditioned VLM	69.7	−0.8	70.1	−3.8

Table 5: Execution-verified learning on flagged inputs. The full policy is compared with the raw base prediction, training from successful trajectories without counterfactual verification, and a prompted agent without a trained adapter.

Training variant	OK-VQA	A-OKVQA
Raw base prediction	0.345	0.432
Without verification	0.399	0.489
Prompted agent without training	0.435	0.527
Full execution-verified learning	**0.496**	**0.565**

5.5 OUTCOME GROUNDED POLICY SUPERVISION

In table 5, we compare the learning approaches on flagged inputs. Execution-verified learning reaches 0.496 on OK-VQA and 0.565 on A-OKVQA, exceeding both the prompted agent and training without verification. We attribute the improvement to supervision that compares alternative executed continuations from the same history, since successful trajectories alone do not establish which action was preferable. Further policy learning ablations appear in Appendix D.12.

6 CONCLUSION

In this work, we introduced U-TRACE for adaptive evidence acquisition in open-domain VQA. The framework first estimates the reliability of an initial VLM answer from its behavior under complementary probe contexts, then treats evidence selection as a separate decision conditioned on the question and accumulated evidence. For routed inputs, ROLEAGENT selects a source, specifies a request, and either acquires further evidence or answers using the information already available. Execution-verified learning trains these sequential decisions by comparing the final VQA outcomes of alternative executed continuations from the same history. Across the evaluated VLMs and benchmarks, U-TRACE consistently improves matched base VLMs, with the largest gains on inputs selected for evidence acquisition. Ablations further support the contributions of reliability routing, history-conditioned decisions, and outcome-based supervision.

AI USE STATEMENT

We used generative AI tools to assist with coding and with drafting and editing portions of the manuscript.

ETHICS STATEMENT

We have read the ICLR Code of Ethics and adhere to it. This study does not involve research with human participants or the collection of personally identifiable information. We declare no conflicts of interest or financial relationships that could have influenced this research.

Reproducibility statement

To support reproducibility, we provide the relevant implementation and training details, including the training procedure, model configurations, and evaluation protocols, in the main paper and appendices. The assumptions and proofs underlying our theoretical analysis are also presented in the appendices. All datasets used in our experiments are publicly available and cited in the paper. We additionally include anonymized source code in the supplementary material.

References

Stanislaw Antol, Aishwarya Agrawal, Jiasen Lu, Margaret Mitchell, Dhruv Batra, C Lawrence Zitnick, and Devi Parikh. Vqa: Visual question answering. In *Proceedings of the IEEE international conference on computer vision*, pp. 2425–2433, 2015.

Shuai Bai, Keqin Chen, Xuejing Liu, Jialin Wang, Wenbin Ge, Sibo Song, Kai Dang, Peng Wang, Shijie Wang, Jun Tang, Humen Zhong, Yuanzhi Zhu, Mingkun Yang, Zhaohai Li, Jianqiang Wan, Pengfei Wang, Wei Ding, Zheren Fu, Yiheng Xu, Jiabo Ye, Xi Zhang, Tianbao Xie, Zesen Cheng, Hang Zhang, Zhibo Yang, Haiyang Xu, and Junyang Lin. Qwen2.5-vl technical report, 2025. URL https://arxiv.org/abs/2502.13923.

Chao Chen, Kai Liu, Ze Chen, Yi Gu, Yue Wu, Mingyuan Tao, Zhihang Fu, and Jieping Ye. Inside: Llms' internal states retain the power of hallucination detection. In *International Conference on Learning Representations*, 2024a.

Gongwei Chen, Leyang Shen, Rui Shao, Xiang Deng, and Liqiang Nie. Lion: Empowering multimodal large language model with dual-level visual knowledge. In *2024 IEEE/CVF Conference on Computer Vision and Pattern Recognition (CVPR)*, pp. 26530–26540. IEEE, 2024b.

Zhuohong Chen, Zhenxian Wu, Yunyao Yu, Hangrui Xu, Zirui Liao, Zhifang Liu, Xiangwen Deng, Pen Jiao, and Haoqian Wang. Learning to search: A decision-based agent for knowledge-based visual question answering. *arXiv preprint arXiv:2604.07146*, 2026.

Yung-Sung Chuang, Yujia Xie, Hongyin Luo, Yoon Kim, James R Glass, and Pengcheng He. Dola: Decoding by contrasting layers improves factuality in large language models. In *International Conference on Learning Representations*, volume 2024, pp. 54158–54183, 2024.

Federico Cocchi, Nicholas Moratelli, Marcella Cornia, Lorenzo Baraldi, and Rita Cucchiara. Augmenting multimodal llms with self-reflective tokens for knowledge-based visual question answering. In *2025 IEEE/CVF Conference on Computer Vision and Pattern Recognition (CVPR)*, pp. 9199–9209. IEEE, 2025.

Corentin Dancette, Spencer Whitehead, Rishabh Maheshwary, Ramakrishna Vedantam, Stefan Scherer, Xinlei Chen, Matthieu Cord, and Marcus Rohrbach. Improving selective visual question answering by learning from your peers. In *2023 IEEE/CVF Conference on Computer Vision and Pattern Recognition (CVPR)*, pp. 24049–24059. IEEE, 2023.

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, 2024. URL https://arxiv.org/abs/2409.17146.

Sebastian Farquhar, Jannik Kossen, Lorenz Kuhn, and Yarin Gal. Detecting hallucinations in large language models using semantic entropy. *Nature*, 630(8017):625–630, 2024.

Aaron Grattafiori, Abhimanyu Dubey, Abhinav Jauhri, Abhinav Pandey, Abhishek Kadian, Ahmad Al-Dahle, Aiesha Letman, Akhil Mathur, Alan Schelten, Alex Vaughan, Amy Yang, Angela Fan, Anirudh Goyal, Anthony Hartshorn, Aobo Yang, Archi Mitra, Archie Sravankumar, Artem Korenev, Arthur Hinsvark, Arun Rao, Aston Zhang, Aurelien Rodriguez, Austen Gregerson, Ava Spataru, Baptiste Roziere, Bethany Biron, Binh Tang, Bobbie Chern, Charlotte Caucheteux, Chaya Nayak, Chloe Bi, Chris Marra, Chris McConnell, Christian Keller, Christophe Touret, Chunyang Wu, Corinne Wong, Cristian Canton Ferrer, Cyrus Nikolaidis, Damien Allonsius, Daniel Song, Danielle Pintz, Danny Livshits, Danny Wyatt, David Esiobu, Dhruv Choudhary, Dhruv Mahajan, Diego Garcia-Olano, Diego Perino, Dieuwke Hupkes, Egor Lakomkin, Ehab AlBadawy, Elina Lobanova, Emily Dinan, Eric Michael Smith, Filip Radenovic, Francisco Guzmán, Frank Zhang, Gabriel Synnaeve, Gabrielle Lee, Georgia Lewis Anderson, Govind Thattai, Graeme Nail, Gregoire Mialon, Guan Pang, Guillem Cucurell, Hailey Nguyen, Hannah Korevaar, Hu Xu, Hugo Touvron, Iliyan Zarov, Imanol Arrieta Ibarra, Isabel Kloumann, Ishan Misra, Ivan Evtimov, Jack Zhang, Jade Copet, Jaewon Lee, Jan Geffert, Jana Vranes, Jason Park, Jay Mahadeokar, Jeet Shah, Jelmer van der Linde, Jennifer Billock, Jenny Hong, Jenya Lee, Jeremy Fu, Jianfeng Chi, Jianyu Huang, Jiawen Liu, Jie Wang, Jiecao Yu, Joanna Bitton, Joe Spisak, Jongsoo Park, Joseph Rocca, Joshua Johnstun, Joshua Saxe, Junteng Jia, Kalyan Vasuden Alwala, Karthik Prasad, Kartikeya Upasani, Kate Plawiak, Ke Li, Kenneth Heafield, Kevin Stone, Khalid El-Arini, Krithika Iyer, Kshitiz Malik, Kuenley Chiu, Kunal Bhalla, Kushal Lakhotia, Lauren Rantala-Yeary, Laurens van der Maaten, Lawrence Chen, Liang Tan, Liz Jenkins, Louis Martin, Lovish Madaan, Lubo Malo, Lukas Blecher, Lukas Landzaat, Luke de Oliveira, Madeline Muzzi, Mahesh Pasupuleti, Mannat Singh, Manohar Paluri, Marcin Kardas, Maria Tsimpoukelli, Mathew Oldham, Mathieu Rita, Maya Pavlova, Melanie Kambadur, Mike Lewis, Min Si, Mitesh Kumar Singh, Mona Hassan, Naman Goyal, Narjes Torabi, Nikolay Bashlykov, Nikolay Bogoychev, Niladri Chatterji, Ning Zhang, Olivier Duchenne, Onur Çelebi, Patrick Alrassy, Pengchuan Zhang, Pengwei Li, Petar Vasic, Peter Weng, Prajjwal Bhargava, Pratik Dubal, Praveen Krishnan, Punit Singh Koura, Puxin Xu, Qing He, Qingxiao Dong, Ragavan Srinivasan, Raj Ganapathy, Ramon Calderer, Ricardo Silveira Cabral, Robert Stojnic, Roberta Raileanu, Rohan Maheswari, Rohit Girdhar, Rohit Patel, Romain Sauvestre, Ronnie Polidoro, Roshan Sumbaly, Ross Taylor, Ruan Silva, Rui Hou, Rui Wang, Saghar Hosseini, Sahana Chennabasappa, Sanjay Singh, Sean Bell, Seohyun Sonia Kim, Sergey Edunov, Shaoliang Nie, Sharan Narang, Sharath Raparthy, Sheng Shen, Shengye Wan, Shruti Bhosale, Shun Zhang, Simon Vandenhende, Soumya Batra, Spencer Whitman, Sten Sootla, Stephane Collot, Suchin Gururangan, Sydney Borodinsky, Tamar Herman, Tara Fowler, Tarek Sheasha, Thomas Georgiou, Thomas Scialom, Tobias Speckbacher, Todor Mihaylov, Tong Xiao, Ujjwal Karn, Vedanuj Goswami, Vibhor Gupta, Vignesh Ramanathan, Viktor Kerkez, Vincent Gonguet, Virginie Do, Vish Vogeti, Vítor Albiero, Vladan Petrovic, Weiwei Chu, Wenhan Xiong, Wenyin Fu, Whitney Meers, Xavier Martinet, Xiaodong Wang, Xiaofang Wang, Xiaoqing Ellen Tan, Xide Xia, Xinfeng Xie, Xuchao Jia, Xuewei Wang, Yaelle Goldschlag, Yashesh Gaur, Yasmine Babaei, Yi Wen, Yiwen Song, Yuchen Zhang, Yue Li, Yuning Mao, Zacharie Delpierre Coudert, Zheng Yan, Zhengxing Chen, Zoe Papakipos, Aaditya Singh, Aayushi Srivastava, Abha Jain, Adam Kelsey, Adam Shajnfeld, Adithya Gangidi, Adolfo Victoria, Ahuva Goldstand, Ajay Menon, Ajay Sharma, Alex Boesenberg, Alexei Baevski, Allie Feinstein, Amanda Kallet, Amit Sangani, Amos Teo, Anam Yunus, Andrei Lupu, Andres Alvarado, Andrew Caples, Andrew Gu, Andrew Ho, Andrew Poulton, Andrew Ryan, Ankit Ramchandani, Annie Dong, Annie Franco, Anuj Goyal, Aparajita Saraf, Arkabandhu Chowdhury, Ashley Gabriel, Ashwin Bharambe, Assaf Eisenman, Azadeh Yazdan, Beau James, Ben Maurer, Benjamin Leonhardi, Bernie Huang, Beth Loyd, Beto De Paola, Bhargavi Paranjape, Bing Liu, Bo Wu, Boyu Ni, Braden Hancock, Bram Wasti, Brandon Spence, Brani Stojkovic, Brian Gamido, Britt Montalvo, Carl Parker, Carly Burton, Catalina Mejia, Ce Liu, Changhan Wang, Changkyu Kim, Chao Zhou, Chester Hu, Ching-Hsiang Chu, Chris Cai, Chris Tindal, Christoph Feichtenhofer, Cynthia Gao, Damon Civin, Dana Beaty, Daniel Kreymer, Daniel Li, David Adkins, David Xu, Davide Testuggine, Delia David, Devi Parikh, Diana Liskovich, Didem Foss, Dingkang Wang, Duc Le, Dustin Holland, Edward Dowling, Eissa Jamil, Elaine Montgomery, Eleonora Presani, Emily Hahn, Emily Wood, Eric-Tuan Le, Erik Brinkman, Esteban Arcaute, Evan Dunbar, Evan Smothers, Fei Sun, Felix Kreuk, Feng Tian, Filippos Kokkinos, Firat Ozgenel, Francesco Caggioni, Frank Kanayet, Frank Seide, Gabriela Medina Florez, Gabriella Schwarz, Gada Badeer, Georgia Swee, Gil Halpern, Grant Herman, Grigory Sizov, Guangyi, Zhang, Guna Lakshminarayanan, Hakan Inan, Hamid Shojanazeri, Han Zou, Hannah Wang, Hanwen Zha, Haroun Habeeb, Harrison Rudolph, Helen Suk, Henry Aspegren, Hunter Goldman, Hongyuan Zhan, Ibrahim Damlaj, Igor Molybog, Igor Tufanov, Ilias Leontiadis, Irina-Elena Veliche, Itai Gat, Jake Weissman, James Geboski, James

Kohli, Janice Lam, Japhet Asher, Jean-Baptiste Gaya, Jeff Marcus, Jeff Tang, Jennifer Chan, Jenny Zhen, Jeremy Reizenstein, Jeremy Teboul, Jessica Zhong, Jian Jin, Jingyi Yang, Joe Cummings, Jon Carvill, Jon Shepard, Jonathan McPhie, Jonathan Torres, Josh Ginsburg, Junjie Wang, Kai Wu, Kam Hou U, Karan Saxena, Kartikay Khandelwal, Katayoun Zand, Kathy Matosich, Kaushik Veeraraghavan, Kelly Michelena, Keqian Li, Kiran Jagadeesh, Kun Huang, Kunal Chawla, Kyle Huang, Lailin Chen, Lakshya Garg, Lavender A, Leandro Silva, Lee Bell, Lei Zhang, Liangpeng Guo, Licheng Yu, Liron Moshkovich, Luca Wehrstedt, Madian Khabsa, Manav Avalani, Manish Bhatt, Martynas Mankus, Matan Hasson, Matthew Lennie, Matthias Reso, Maxim Groshev, Maxim Naumov, Maya Lathi, Meghan Keneally, Miao Liu, Michael L. Seltzer, Michal Valko, Michelle Restrepo, Mihir Patel, Mik Vyatskov, Mikayel Samvelyan, Mike Clark, Mike Macey, Mike Wang, Miquel Jubert Hermoso, Mo Metanat, Mohammad Rastegari, Munish Bansal, Nandhini Santhanam, Natascha Parks, Natasha White, Navyata Bawa, Nayan Singhal, Nick Egebo, Nicolas Usunier, Nikhil Mehta, Nikolay Pavlovich Laptev, Ning Dong, Norman Cheng, Oleg Chernoguz, Olivia Hart, Omkar Salpekar, Ozlem Kalinli, Parkin Kent, Parth Parekh, Paul Saab, Pavan Balaji, Pedro Rittner, Philip Bontrager, Pierre Roux, Piotr Dollar, Polina Zvyagina, Prashant Ratanchandani, Pritish Yuvraj, Qian Liang, Rachad Alao, Rachel Rodriguez, Rafi Ayub, Raghotham Murthy, Raghu Nayani, Rahul Mitra, Rangaprabhu Parthasarathy, Raymond Li, Rebekkah Hogan, Robin Battey, Rocky Wang, Russ Howes, Ruty Rinott, Sachin Mehta, Sachin Siby, Sai Jayesh Bondu, Samyak Datta, Sara Chugh, Sara Hunt, Sargun Dhillon, Sasha Sidorov, Satadru Pan, Saurabh Mahajan, Saurabh Verma, Seiji Yamamoto, Sharadh Ramaswamy, Shaun Lindsay, Shaun Lindsay, Sheng Feng, Shenghao Lin, Shengxin Cindy Zha, Shishir Patil, Shiva Shankar, Shuqiang Zhang, Shuqiang Zhang, Sinong Wang, Sneha Agarwal, Soji Sajuyigbe, Soumith Chintala, Stephanie Max, Stephen Chen, Steve Kehoe, Steve Satterfield, Sudarshan Govindaprasad, Sumit Gupta, Summer Deng, Sungmin Cho, Sunny Virk, Suraj Subramanian, Sy Choudhury, Sydney Goldman, Tal Remez, Tamar Glaser, Tamara Best, Thilo Koehler, Thomas Robinson, Tianhe Li, Tianjun Zhang, Tim Matthews, Timothy Chou, Tzook Shaked, Varun Vontimitta, Victoria Ajayi, Victoria Montanez, Vijai Mohan, Vinay Satish Kumar, Vishal Mangla, Vlad Ionescu, Vlad Poenaru, Vlad Tiberiu Mihailescu, Vladimir Ivanov, Wei Li, Wenchen Wang, Wenwen Jiang, Wes Bouaziz, Will Constable, Xiaocheng Tang, Xiaojian Wu, Xiaolan Wang, Xilun Wu, Xinbo Gao, Yaniv Kleinman, Yanjun Chen, Ye Hu, Ye Jia, Ye Qi, Yenda Li, Yilin Zhang, Ying Zhang, Yossi Adi, Youngjin Nam, Yu, Wang, Yu Zhao, Yuchen Hao, Yundi Qian, Yunlu Li, Yuzi He, Zach Rait, Zachary DeVito, Zef Rosnbrick, Zhaoduo Wen, Zhenyu Yang, Zhiwei Zhao, and Zhiyu Ma. The llama 3 herd of models, 2024. URL `https://arxiv.org/abs/2407.21783`.

Tianrui Guan, Fuxiao Liu, Xiyang Wu, Ruiqi Xian, Zongxia Li, Xiaoyu Liu, Xijun Wang, Lichang Chen, Furong Huang, Yaser Yacoob, et al. Hallusionbench: an advanced diagnostic suite for entangled language hallucination and visual illusion in large vision-language models. In *2024 IEEE/CVF Conference on Computer Vision and Pattern Recognition (CVPR)*, pp. 14375–14385. IEEE, 2024.

Liangke Gui, Borui Wang, Qiuyuan Huang, Alexander G Hauptmann, Yonatan Bisk, and Jianfeng Gao. Kat: A knowledge augmented transformer for vision-and-language. In *Proceedings of the 2022 conference of the North American chapter of the association for computational linguistics: human language technologies*, pp. 956–968, 2022.

Jiaxian Guo, Junnan Li, Dongxu Li, Anthony Meng Huat Tiong, Boyang Li, Dacheng Tao, and Steven Hoi. From images to textual prompts: Zero-shot visual question answering with frozen large language models. In *2023 IEEE/CVF Conference on Computer Vision and Pattern Recognition (CVPR)*, pp. 10867–10877. IEEE, 2023.

Danna Gurari, Qing Li, Abigale J Stangl, Anhong Guo, Chi Lin, Kristen Grauman, Jiebo Luo, and Jeffrey P Bigham. Vizwiz grand challenge: Answering visual questions from blind people. In *2018 IEEE/CVF Conference on Computer Vision and Pattern Recognition*, pp. 3608–3617. IEEE, 2018.

Dongze Hao, Jian Jia, Longteng Guo, Qunbo Wang, Te Yang, Yan Li, Yanhua Cheng, Bo Wang, Quan Chen, Han Li, et al. Knowledge condensation and reasoning for knowledge-based vqa. *arXiv preprint arXiv:2403.10037*, 2024.

Wenyi Hong, Wenmeng Yu, Xiaotao Gu, Guo Wang, Guobing Gan, Haomiao Tang, Jiale Cheng, Ji Qi, Junhui Ji, Lihang Pan, et al. Glm-4.1 v-thinking: Towards versatile multimodal reasoning with scalable reinforcement learning. *arXiv e-prints*, pp. arXiv–2507, 2025.

Bairu Hou, Yujian Liu, Kaizhi Qian, Jacob Andreas, Shiyu Chang, and Yang Zhang. Decomposing uncertainty for large language models through input clarification ensembling. *arXiv preprint arXiv:2311.08718*, 2023.

Yushi Hu, Hang Hua, Zhengyuan Yang, Weijia Shi, Noah A Smith, and Jiebo Luo. Promptcap: Prompt-guided image captioning for vqa with gpt-3. In *2023 IEEE/CVF International Conference on Computer Vision (ICCV)*, pp. 2951–2963. IEEE, 2023a.

Yushi Hu, Otilia Stretcu, Chun-Ta Lu, Krishnamurthy Viswanathan, Kenji Hata, Enming Luo, Ranjay Krishna, and Ariel Fuxman. Visual program distillation: Distilling tools and programmatic reasoning into vision-language models. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, pp. 9590–9601, 2024.

Ziniu Hu, Ahmet Iscen, Chen Sun, Kai-Wei Chang, Yizhou Sun, David Ross, Cordelia Schmid, and Alireza Fathi. Avis: Autonomous visual information seeking with large language model agent. *Advances in Neural Information Processing Systems*, 36:867–878, 2023b.

Xinyu Huang, Yi-Jie Huang, Youcai Zhang, Weiwei Tian, Rui Feng, Yuejie Zhang, Yanchun Xie, Yaqian Li, and Lei Zhang. Open-set image tagging with multi-grained text supervision. In *Proceedings of the 33rd ACM International Conference on Multimedia*, pp. 4117–4126, 2025.

Drew A Hudson and Christopher D Manning. Gqa: A new dataset for real-world visual reasoning and compositional question answering. In *2019 IEEE/CVF Conference on Computer Vision and Pattern Recognition (CVPR)*, pp. 6693–6702. IEEE, 2019.

Eyke Hüllermeier and Willem Waegeman. Aleatoric and epistemic uncertainty in machine learning: An introduction to concepts and methods. *Machine learning*, 110(3):457–506, 2021.

Zhengbao Jiang, Frank F Xu, Luyu Gao, Zhiqing Sun, Qian Liu, Jane Dwivedi-Yu, Yiming Yang, Jamie Callan, and Graham Neubig. Active retrieval augmented generation. In *Proceedings of the 2023 conference on empirical methods in natural language processing*, pp. 7969–7992, 2023.

Saurav Kadavath, Tom Conerly, Amanda Askell, Tom Henighan, Dawn Drain, Ethan Perez, Nicholas Schiefer, Zac Hatfield-Dodds, Nova DasSarma, Eli Tran-Johnson, et al. Language models (mostly) know what they know. *arXiv preprint arXiv:2207.05221*, 2022.

Zaid Khan and Yun Fu. Consistency and uncertainty: Identifying unreliable responses from black-box vision-language models for selective visual question answering. In *2024 IEEE/CVF Conference on Computer Vision and Pattern Recognition (CVPR)*, pp. 10854–10863. IEEE, 2024.

Mahyar Khayatkhoei, Prateek Chhikara, Filip Ilievski, et al. Mllms know where to look: Training-free perception of small visual details with multimodal llms. In *International Conference on Learning Representations*, volume 2025, pp. 68194–68213, 2025.

Jieun Kim, Yujin Jeong, and Sung-Bae Cho. Visual–linguistic abductive reasoning with llms for knowledge-based visual question answering. In *Findings of the Association for Computational Linguistics: EACL 2026*, pp. 6529–6544, 2026.

Jannik Kossen, Jiatong Han, Muhammed Razzak, Lisa Schut, Shreshth Malik, and Yarin Gal. Semantic entropy probes: Robust and cheap hallucination detection in llms. In *ICML Workshop on Foundation Models in the Wild*, 2024. URL https://openreview.net/forum?id=Zd0XLr6JKn.

Lorenz Kuhn, Yarin Gal, and Sebastian Farquhar. Semantic uncertainty: Linguistic invariances for uncertainty estimation in natural language generation. *arXiv preprint arXiv:2302.09664*, 2023.

Sicong Leng, Hang Zhang, Guanzheng Chen, Xin Li, Shijian Lu, Chunyan Miao, and Lidong Bing. Mitigating object hallucinations in large vision-language models through visual contrastive decoding. In *2024 IEEE/CVF Conference on Computer Vision and Pattern Recognition (CVPR)*, pp. 13872–13882. IEEE, 2024.

Mingfu Liang, Ying Wu, et al. Toa: Task-oriented active vqa. *Advances in Neural Information Processing Systems*, 36:54061–54074, 2023.

Weizhe Lin and Bill Byrne. Retrieval augmented visual question answering with outside knowledge. In *Proceedings of the 2022 conference on empirical methods in natural language processing*, pp. 11238–11254, 2022.

Weizhe Lin, Jinghong Chen, Jingbiao Mei, Alexandru Coca, and Bill Byrne. Fine-grained late-interaction multi-modal retrieval for retrieval augmented visual question answering. *Advances in Neural Information Processing Systems*, 36:22820–22840, 2023.

Weizhe Lin, Jingbiao Mei, Jinghong Chen, and Bill Byrne. Preflmr: Scaling up fine-grained late-interaction multi-modal retrievers. In *Proceedings of the 62nd Annual Meeting of the Association for Computational Linguistics (Volume 1: Long Papers)*, pp. 5294–5316, 2024.

Yuanze Lin, Yujia Xie, Dongdong Chen, Yichong Xu, Chenguang Zhu, and Lu Yuan. Revive: Regional visual representation matters in knowledge-based visual question answering. *Advances in neural information processing systems*, 35:10560–10571, 2022.

Weitang Liu, Xiaoyun Wang, John D. Owens, and Yixuan Li. Energy-based out-of-distribution detection. In *Advances in Neural Information Processing Systems*, volume 33, 2020.

Pan Lu, Baolin Peng, Hao Cheng, Michel Galley, Kai-Wei Chang, Ying Nian Wu, Song-Chun Zhu, and Jianfeng Gao. Chameleon: Plug-and-play compositional reasoning with large language models. *Advances in Neural Information Processing Systems*, 36:43447–43478, 2023.

Kenneth Marino, Mohammad Rastegari, Ali Farhadi, and Roozbeh Mottaghi. Ok-vqa: A visual question answering benchmark requiring external knowledge. In *2019 IEEE/CVF conference on computer vision and pattern recognition (CVPR)*, pp. 3190–3199. IEEE, 2019.

Timo Schick, Jane Dwivedi-Yu, Roberto Dessì, Roberta Raileanu, Maria Lomeli, Eric Hambro, Luke Zettlemoyer, Nicola Cancedda, and Thomas Scialom. Toolformer: Language models can teach themselves to use tools. *Advances in neural information processing systems*, 36:68539–68551, 2023.

Dustin Schwenk, Apoorv Khandelwal, Christopher Clark, Kenneth Marino, and Roozbeh Mottaghi. A-okvqa: A benchmark for visual question answering using world knowledge. In *European conference on computer vision*, pp. 146–162. Springer, 2022.

Zhenwei Shao, Zhou Yu, Meng Wang, and Jun Yu. Prompting large language models with answer heuristics for knowledge-based visual question answering. In *2023 IEEE/CVF Conference on Computer Vision and Pattern Recognition (CVPR)*, pp. 14974–14983. IEEE, 2023.

Dídac Surís, Sachit Menon, and Carl Vondrick. Vipergpt: Visual inference via python execution for reasoning. In *2023 IEEE/CVF International Conference on Computer Vision (ICCV)*, pp. 11854–11864. IEEE, 2023.

Haibo Wang and Weifeng Ge. Q&a prompts: Discovering rich visual clues through mining question-answer prompts for vqa requiring diverse world knowledge. In *European conference on computer vision*, pp. 274–292. Springer, 2024.

Weiyun Wang, Zhangwei Gao, Lixin Gu, Hengjun Pu, Long Cui, Xingguang Wei, Zhaoyang Liu, Linglin Jing, Shenglong Ye, Jie Shao, Zhaokai Wang, Zhe Chen, Hongjie Zhang, Ganlin Yang, Haomin Wang, Qi Wei, Jinhui Yin, Wenhao Li, Erfei Cui, Guanzhou Chen, Zichen Ding, Changyao Tian, Zhenyu Wu, Jingjing Xie, Zehao Li, Bowen Yang, Yuchen Duan, Xuehui Wang, Zhi Hou, Haoran Hao, Tianyi Zhang, Songze Li, Xiangyu Zhao, Haodong Duan, Nianchen Deng, Bin Fu, Yinan He, Yi Wang, Conghui He, Botian Shi, Junjun He, Yingtong Xiong, Han Lv, Lijun Wu, Wenqi Shao, Kaipeng Zhang, Huipeng Deng, Biqing Qi, Jiaye Ge, Qipeng Guo, Wenwei Zhang, Songyang Zhang, Maosong Cao, Junyao Lin, Kexian Tang, Jianfei Gao, Haian Huang, Yuzhe Gu, Chengqi Lyu, Huanze Tang, Rui Wang, Haijun Lv, Wanli Ouyang, Limin Wang, Min Dou, Xizhou Zhu, Tong Lu, Dahua Lin, Jifeng Dai, Weijie Su, Bowen Zhou, Kai Chen, Yu Qiao, Wenhai Wang, and Gen Luo. Internvl3.5: Advancing open-source multimodal models in versatility, reasoning, and efficiency, 2025. URL https://arxiv.org/abs/2508.18265.

ZhengXian Wu, Hangrui Xu, Kai Shi, Zhuohong Chen, Yunyao Yu, Chuanrui Zhang, Zirui Liao, Jun Yang, Zhenyu Yang, Haonan Lu, et al. Promsa: Progressive multimodal search agents for knowledge-based visual question answering. In *European Conference on Computer Vision*, pp. 127–143. Springer, 2026.

Zhen Yang, Zhuo Tao, Qi Chen, Liang Li, Yuankai Qi, Anton Van Den Hengel, and Qingming Huang. Separation of powers: On segregating knowledge from observation in llm-enabled knowledge-based visual question answering. In *2025 IEEE/CVF Conference on Computer Vision and Pattern Recognition (CVPR)*, pp. 24753–24762. IEEE, 2025.

Zhengyuan Yang, Zhe Gan, Jianfeng Wang, Xiaowei Hu, Yumao Lu, Zicheng Liu, and Lijuan Wang. An empirical study of gpt-3 for few-shot knowledge-based vqa, 2022. URL https://arxiv.org/abs/2109.05014.

Shunyu Yao, Jeffrey Zhao, Dian Yu, Nan Du, Izhak Shafran, Karthik Narasimhan, and Yuan Cao. React: Synergizing reasoning and acting in language models. *arXiv preprint arXiv:2210.03629*, 2022.

A Preliminary Study Details

A.1 Evaluation Protocol

The preliminary studies use Qwen2.5-VL-7B-Instruct (Bai et al., 2025) with greedy decoding and a short answer instruction. They evaluate 5,046 OK-VQA validation questions and 1,075 A-OKVQA validation questions. In the notation of Section 3.1, let $s_i^{(0)} = R(z_0; x_i) \in [0, 1]$ denote the official VQA score of the initial prediction and let $s_i^{(j)} = R(z_j; x_i)$ denote the score after intervention j. We use $\gamma = 0.5$ for the diagnostic analysis. This gives 1,404 initial errors on OK-VQA and 325 on A-OKVQA.

A repair crosses from $s_i^{(0)} < \gamma$ to $s_i^{(j)} \geq \gamma$. A corruption crosses in the opposite direction. The accuracy change is

$$\Delta\mathrm{Acc}_j = \frac{100}{n} \sum_{i=1}^{n} \big(s_i^{(j)} - s_i^{(0)}\big). \tag{22}$$

The score difference measures graded changes in answer quality. Repairs and corruptions record threshold crossings. Reliability signals are evaluated against the original greedy answer. Sampled answers are used only to probe the behavior of the VLM around that answer.

A.2 Detecting Risky Answers

We examine token likelihood, agreement across samples, semantic entropy (Kuhn et al., 2023; Farquhar et al., 2024), hidden state dispersion, and self evaluation (Kadavath et al., 2022). Semantic entropy uses ten sampled answers. The combined estimator uses 21 signals, and 13 individual signals exceed 0.65 AUROC on both benchmarks. Sequence NLL reaches 0.80 AUROC on OK-VQA and 0.78 on A-OKVQA. Semantic entropy reaches 0.78 and 0.79, respectively. The combined out of fold logistic estimator reaches 0.82 and 0.80, with reported intervals $[0.81, 0.83]$ and $[0.77, 0.83]$.

The detector family experiment in Table 1 provides the current detailed comparison using AUROC, precision, and recall. We do not repeat an older detector table here because it used a different reporting rule and would not be directly comparable. The preliminary result needed for the argument is that behavior around the initial answer provides a useful signal for identifying risky predictions.

Ranking, calibration, and residual errors. Calibration is distinct from discriminative ranking. A score can rank errors without being a calibrated probability. Measured ECE is $0.09/0.05$ for sequence probability, $0.08/0.08$ for self consistency, $0.17/0.17$ for $P(\mathrm{True})$, and $0.13/0.15$ for verbalized confidence on OK-VQA and A-OKVQA. These values depend on the probability mapping and calibration bins. A separate probe prompt self consistency variant has ECE from 0.38 to 0.42 while retaining error AUROC from 0.66 to 0.68. These results show that useful ranking does not eliminate false positives or false negatives. The decision process must therefore preserve the initial answer as an explicit option.

A.3 Testing Whether Interventions Help

Six interventions are evaluated separately on the initial VLM. They are VCD (Leng et al., 2024), DoLa (Chuang et al., 2024), RAM++ tags (Huang et al., 2025), EasyOCR text, PreFLMR passages (Lin et al., 2024), and five crop majority voting. VCD uses a diffusion noise contrast with $\alpha = 1$, $\beta = 0.1$, and noise step 500. DoLa uses the Transformers implementation of layer contrastive decoding. Tags and OCR outputs are inserted into the prompt. Retrieval inserts the top three passages. Crop voting aggregates predictions from five crops. This study evaluates configured interventions rather than six newly trained models.

Every intervention repairs some initial predictions but also corrupts others. All six reduce aggregate VQA score on both benchmarks when applied to the full evaluation set. The gated experiment in Figure 3 instead applies each tool to the most uncertain 30% of examples by semantic entropy. That gate changes the magnitude of the losses but does not consistently make a single intervention beneficial.

Table 6: Single interventions applied to all examples at 100% coverage. Repair and corrupt are percentages of the full evaluation set. The VQA score change Δ is measured against the same initial answers.

	OK-VQA		A-OKVQA	
Intervention	Repair/Corrupt	Δ	Repair/Corrupt	Δ
VCD	3.55/8.86	-4.97	3.72/8.09	-3.72
DoLa	5.17/17.66	-11.71	5.49/15.26	-10.05
Five-crop voting	4.14/8.82	-4.44	4.74/13.58	-9.15
EasyOCR	1.19/2.30	-1.01	2.14/2.23	-0.47
PreFLMR	4.08/31.79	-25.89	5.40/24.37	-18.54
RAM++ tags	5.21/14.19	-8.71	4.93/11.72	-6.57

Implementation controls. Backend differences can affect small score changes; all intervention changes are measured against the same initial-answer run within each benchmark. The reported repair and corruption percentages count threshold crossings, whereas Δ uses the graded VQA score.

VCD targets visual hallucination, while DoLa was developed for language model factuality. Neither was optimized for this short answer setting. Inserting PreFLMR passages into a VLM is also not a reproduction of a trained retrieval augmented reader. The retrieval loss describes this integration rather than the retriever in its intended system. OPERA, SpatialRGPT, and REVIVE are not included because their required attention, region, depth, or trained feature pipelines are incompatible with this evaluation.

A.4 Why Reliability Alone Cannot Select an Intervention

The study next tests whether observations available before intervention execution can select a useful action. The tested observations include initial uncertainty, self assessment, prompt and model ensemble decompositions, and modality sensitivity. They identify some risky answers, but they do not consistently separate interventions that repair an answer from interventions that corrupt it. This is an empirical result for the evaluated signals and interventions. It does not claim a universal limit for every possible selector.

For the individual intervention analysis, let $R_j = R(z_j; x)$ and let $e_j = \mathbf{1}[R_j < \gamma]$. Index 0 denotes the initial prediction. For a behavioral summary u, define $p_j^+(u) = \Pr(e_j = 0 \mid e_0 = 1, u_0 = u)$ and $p_j^-(u) = \Pr(e_j = 1 \mid e_0 = 0, u_0 = u)$. The expected thresholded change is

$$g_j(u) = \mathbb{E}[e_0 - e_j \mid u_0 = u] \tag{23}$$

$$= [1 - \rho(u)]p_j^+(u) - \rho(u)p_j^-(u). \tag{24}$$

Reliability $\rho(u)$ is only one factor. Intervention value also depends on the repair rate, the corruption rate, and any execution cost. For the graded VQA score, the corresponding diagnostic target is

$$\Delta_j(u) = \mathbb{E}[R_j - R_0 \mid u_0 = u] = \mathbb{E}[\delta_j(x) \mid u_0 = u], \qquad \delta_j(x) = R_j - R_0. \tag{25}$$

This quantity averages over inputs with the same behavioral summary and over the responses returned by the intervention.

Reliability based gating. Each intervention is applied to the most uncertain tested fractions of the evaluation set using initial sequence NLL. The base prediction is retained elsewhere. Table 7 reports the most favorable tested nonzero budget on OK-VQA. In this separate sequence-NLL-gating analysis, every best observed change remains negative. Initial uncertainty can prioritize both beneficial and harmful changes, so identifying a risky answer is not sufficient for choosing an intervention.

Source attribution. Epistemic and aleatoric uncertainty distinguish uncertainty about a model from variation in outcomes conditional on available information (Hüllermeier & Waegeman, 2021). Input clarification provides an alternative ensemble construction for language models (Hou et al., 2023). The preliminary study tests whether prompt and model ensemble estimates predict reference

Table 7: Reliability gated intervention outcomes on OK-VQA. The best reliability gate is the most favorable tested nonzero sequence NLL threshold for each intervention. The hindsight gate uses reference answers for selection.

	Uncertainty AUROC		ΔAcc in points	
Intervention	Helps	Hurts	Best reliability gate	Hindsight gate
VCD	0.744	0.740	−0.81	+3.54
DoLa	0.669	0.670	−1.09	+5.25
Five-crop voting	0.749	0.644	−0.31	+4.02
EasyOCR text	0.750	0.722	−0.17	+1.20
PreFLMR passages	0.656	0.513	−1.07	+4.21

Table 8: Uncertainty decomposition diagnostics. Correlations and AUROC values depend on the sampled prompt or model ensemble.

Diagnostic	OK-VQA	A-OKVQA
$\mathrm{corr}(U_{\mathrm{alea}}, U_{\mathrm{epis}})$ with prompt ensemble	0.91	0.88
$\mathrm{corr}(U_{\mathrm{alea}}, U_{\mathrm{epis}})$ with six model ensemble	0.87	0.81
corr(component, total uncertainty)	0.95 to 0.99	0.95 to 0.99
Reference disagreement AUROC for alea./epis./total	0.74/0.74/0.75	0.73/0.73/0.74
Recoverability AUROC for alea./epis./total	0.35/0.38/0.35	0.32/0.36/0.33
Mean epistemic estimate with 2 to 6 members	$0.37 \rightarrow 0.85$	$0.34 \rightarrow 0.76$
Prompt and model ensemble rank agreement	0.63	0.57

disagreement or recovery by the available alternatives. These outcomes are operational measurements rather than reference error causes.

The components overlap strongly and depend on ensemble composition. The recoverability results contradict the intended rule that a larger epistemic estimate should prioritize cases repaired by these particular alternatives. Modality ablation, changes induced by perturbations, log probability changes, and answer changes yield intervention discrimination AUROC from 0.55 to 0.59 on OK-VQA and from 0.41 to 0.59 on A-OKVQA. As error detectors, these attribution scores reach 0.48 to 0.52 and 0.45 to 0.47. Sensitivity to an input channel does not establish that the channel is erroneous or that strengthening it will help.

More than half of the initial errors survive both tested interventions. This limits the evaluated menu and configuration, but it does not show that the errors are irreducible. A successful crop or retrieval response should therefore not be interpreted as a definitive diagnosis of the original error.

Selection headroom. Table 9 shows that crop or retrieval repairs approximately 46.3% of initial errors on OK-VQA and 47.4% on A-OKVQA. The percentages are sums of the disjoint repair categories and are subject to rounding. Applied to the initial error counts reported in Section A.1, these proportions correspond to approximately 650 and 154 questions. In a separate answer selection analysis, a label informed chooser reaches VQA scores of 79.55 and 82.42, compared with base scores of 67.22 and 69.74 under that analysis's protocol. Those base scores differ from the random few shot baselines in Table 2. The chooser uses reference answers to select among predictions, so its results indicate possible selection headroom rather than inference performance.

A.5 Transfer Across Benchmarks and Models

Figure 5 groups initial predictions into semantic entropy deciles for five VLMs. On both benchmarks, higher deciles generally contain larger proportions of incorrect answers for each model. This pattern supports semantic entropy as a reliability ranking across the evaluated VLMs, without implying that their error rates are calibrated alike or that the same intervention will benefit them equally.

The intervention results also show the same qualitative pattern on OK-VQA and A-OKVQA. Every substantive uniformly applied intervention both repairs and corrupts answers, and aggregate accuracy

Table 9: Repair outcomes for crop voting and retrieval among initially incorrect answers. The four categories are disjoint and cover the initial errors.

Outcome among initial errors in percent	OK-VQA	A-OKVQA
Crop repairs and retrieval does not	27.1	23.1
Retrieval repairs and crop does not	9.1	10.8
Both independently repair	10.1	13.5
Neither repairs	53.6	52.6

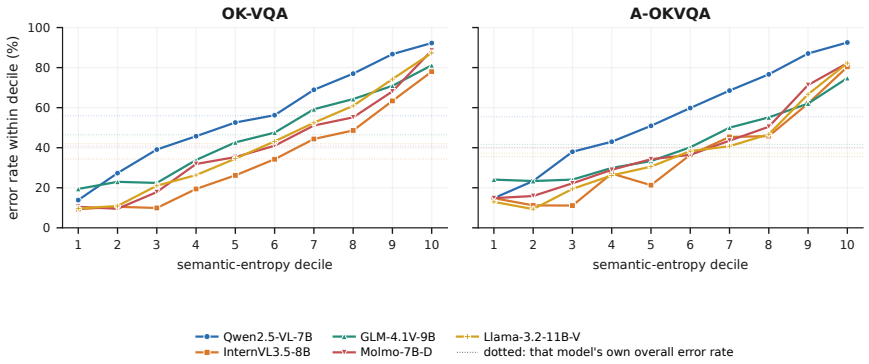


Figure 5: Initial error rate by semantic entropy decile for five VLMs on OK-VQA and A-OKVQA. Decile 1 has the lowest entropy and decile 10 the highest. Each dotted horizontal line gives the corresponding model's overall error rate on that benchmark.

decreases on both benchmarks. The magnitudes differ across benchmarks, so the result supports transfer of the decision problem rather than a universal intervention ranking.

B METHOD DETAILS

We first specify the uncertainty state and learned policy, then give the inference procedure and the value analysis that motivates sequential decisions.

B.1 UNCERTAINTY STATE AND HISTORYUQ

Behavioral coordinates. U_{between} in Equation 7 is the mutual information under the context-indexed mixture; it does not assume that prompts are Bayesian parameter samples. Neither disagreement nor agreement with the visually ablated model establishes correctness by itself. The cache $\mathcal{B}(x)$ describes how the initial predictor behaves, not which source will correct it; detailed probe outputs remain in the cache and can be revealed later as source-attributed responses, so the compact state u_0 is not mistaken for the full evidence record.

HISTORYUQ. The rendering $\varrho(h_t)$ contains the evidence E_t but excludes the uncertainty summary being computed. Because u_0 and u_t^{hist} are produced by different measurement procedures, their values are not directly comparable; only u_t^{hist} is compared across steps. Its statistics describe how settled the fixed assessor is under the current evidence and are not a calibrated probability of correctness. Keeping HISTORYUQ fixed separates policy changes from changes in the measurement at a *fixed history*; it does not prevent the policy from visiting new histories, nor guarantee that more evidence reduces uncertainty. The decision-theoretic formulation uses the full history and does not assume that a compressed feature vector is a sufficient Markov state.

Reliability-based routing. PROBEDETECTOR approximates initial reliability, not the continuation advantage. Routing is an instance-level allocation of computation, distinct from the agent's subsequent source selection; a weak predicted reliability does not establish that any available source will help.

B.2 POLICY AND LEARNING DETAILS

Construction of execution verified data. All supervision comes from training questions. An initial chooser is trained from probe generated candidates and successful training trajectories. Cross fitting then supplies held out training histories for evaluating that chooser. For each fold, the chooser fitted on the other folds is frozen. At each collected history, it answers from the evidence already available, giving the stopping score $\widehat{S}(h)$. For every admissible tool, one request is executed from the same history and its report is appended. The frozen chooser then runs to an answer. It may call further tools before stopping, subject to the remaining request horizon. The final answer score gives $\widehat{Q}(h, a)$. Only the first tool is forced, while subsequent choices are made by the same chooser used for the comparison. The largest positive observed advantage supplies the tool target. If no tested tool helps, the target is `STOP`. Arguments of beneficial requests supply targets for the `request` role. Visited continuation histories also supply targets for answer arbitration. Ground truth answers are used for offline scoring and are excluded from inference prompts and policy states. Validation examples are used only for reporting results.

Roles. The `request` role is invoked only when the selected source has a nontrivial argument space; cache lookups and deterministic operators have none, and already exposed cached channels are removed from the menu. The admissible menu contains only requests feasible within the remaining request horizon. Auxiliary assessment or rewriting functions may be used by the terminal procedure. Neither shared parameterization nor role labels alone ensures positive transfer between roles; this is evaluated empirically.

Execution cache. Executions are cached by input, ordered evidence history, tool arguments, tool-model configuration, and evaluator configuration, since a tool's return at the initial state cannot in general substitute for its return after other evidence has been acquired. Cached observations and returns are reusable only under matching generating and scoring configurations. Caching makes a realized return reproducible; finitely many sampled executions do not make the expected action value exact.

Algorithm 1 U-TRACE inference with shared answer generation and final rephrasing

Require: input x, VLM, PROBESET configuration, TOOLVLM, information sources
Require: language backbone ψ, adapter ϕ
Require: PROBEDETECTOR d_η, routing threshold κ, request horizon H
Require: final answer rephraser F and few shot examples
Run PROBESET to obtain behavioral cache $\mathcal{B}(x)$ and incumbent z_0
Form the initial U-STATE $u_0 = \Phi(\mathcal{B}(x))$
Set $r_0 \leftarrow d_\eta(u_0)$, $E_0 \leftarrow \varnothing$, and $t \leftarrow 0$
Set $h_0 \leftarrow (q, z_0, \mathcal{B}(x), u_0, r_0, E_0)$
if $r_0 < \kappa$ **then**
    Obtain $z = D_\phi(h_0)$ and basis b from the `answer` role
else
    while $t < H$ and an admissible source remains **do**
        Sample $\hat{p}_t$ with HISTORYUQ from $\varrho(h_t)$
        Set $u_t^{\text{hist}} \leftarrow \Psi(\hat{p}_t)$ and render s_t
        Obtain $P(\text{tool})$ and $P(\text{answer})$ from the `select` role
        if $P(\text{tool})/[P(\text{tool}) + P(\text{answer})] < 0.5$ **then**
            break
        end if
        Choose an available source j using the `select` role
        Generate $\omega \sim \pi_\phi^{\text{request}}(\cdot \mid s_t, j)$ if required
        Obtain $o_{t+1} \sim P_j(\cdot \mid x, h_t, \omega)$, via TOOLVLM when VLM-mediated
        $E_{t+1} \leftarrow E_t \oplus ((j, \omega), o_{t+1})$ and update h_{t+1}
        Update the admissible source menu; $t \leftarrow t + 1$
    end while
    Obtain $z = D_\phi(h_t)$ and basis b from the `answer` role
end if
Obtain rephrasing decision v from F using the question, answer, basis, reports, and few shot examples
return z if v is `keep`, otherwise the answer in `switch`

Hindsight labels. A beneficial request is not paired with a contradictory stopping label at the same history. Each sampled continuation gives a realized return, which may differ from the expected return under repeated tool responses. Cross fitted comparisons use the same frozen chooser and tool configuration. Their targets account for later tool calls made during the sampled continuation, but they do not provide exact optimal values for every possible future history.

Weights and leakage control. Auxiliary roles use task-appropriate supervision. When several terminal targets receive credit, weighted rows implement a normalized soft target within that state; these targets express supervision preferences, not optimal expected-utility decoding. Reference labels are used only to construct offline targets and evaluate trajectories, and are excluded from policy states. When solved training exemplars are exposed as evidence, a training query excludes itself and other examples from the same image.

B.3 INFERENCE PROCEDURE

Algorithm 1 summarizes inference. Both routing branches use the ROLEAGENT answer role, followed by the same final answer rephraser. The evidence acquisition loop ends when the VOI GATE chooses to answer, no admissible source remains, or the request horizon is reached.

B.4 EVIDENCE VALUE, COSTS, AND OPTION VALUE

Evidence value. The value of request $a = (j, \omega)$ relative to evidence E is

$$\delta_a(x; E) = R\big(z(E \oplus (a, o)); x\big) - R\big(z(E); x\big), \qquad o \sim P_j(\cdot \mid x, E, \omega). \tag{26}$$

It is random through the response o, can be negative when a response is irrelevant or misleading, and depends on E. The same request can be decisive under one evidence set and redundant under

another. The single-request gains in Section 3.2 correspond to $E = \varnothing$. If the terminal procedure answers from the evidence as the VLM does, $D_\phi(h) = z(E)$ for a history with evidence E, then by the tower property

$$A_1(h, a) = -\lambda c(h, a) + \mathbb{E}\big[\delta_a(x; E) \mid h\big], \tag{27}$$

whereas the preliminary quantities average δ_a over all inputs with the same u_0 and over all responses the source may return.

Costs. Costs in Equation 13 are optional terms in the decision formulation and could represent tool calls, generated tokens, latency, or repeated HISTORYUQ sampling. Initial probing is incurred before routing and is omitted from that conditional objective. We do not claim an end-to-end efficiency gain from this formulation; such a comparison would also need to account for probing and detector inference.

Immediate gain and option value. Fix the terminal procedure, tool-response kernels, and remaining request budget $b \geq 1$. With $h' = h \oplus (a, o)$, the history whose evidence sequence is extended by the new response, the Bellman equations imply

$$A_b^*(h, a) = -\lambda c(h, a) + \mathbb{E}_{o|h,a}[V_{b-1}^*(h')] - S(h) \tag{28}$$

$$= A_1(h, a) + \mathbb{E}_{o|h,a}[V_{b-1}^*(h') - S(h')]. \tag{29}$$

The second equality adds and subtracts $\mathbb{E}_{o|h,a}[S(h')]$. Hence

$$A_b^*(h, a) = A_1(h, a) + \mathbb{E}_{o|h,a}\big[V_{b-1}^*(h') - S(h')\big]. \tag{30}$$

Because stopping remains available, $V_{b-1}^*(h') \geq S(h')$; the option term is nonnegative and vanishes at $b = 1$. It can make a request valuable even when its immediate gain is negative, but only if a sufficiently useful continuation exists. This is a statement about optimal expected returns under the specified configuration, not guaranteed benefit from a learned policy. The term need not reflect lower uncertainty. Subsequent evidence may instead change which answer or tool is preferable.

This decomposition ties the preliminary decision problem to the method's history-conditioned state, explicit stopping action, and history-specific action credit. An initial reliability summary cannot track the changed responses in h'; the controller therefore retains their content, while HISTORYUQ measures uncertainty under that evidence. Executed continuations provide observations of their value under a particular terminal procedure, rather than assuming that uncertainty reduction is utility. Keeping HISTORYUQ fixed makes uncertainty comparable across policy updates at a fixed history, but does not make value estimates exact or ensure that uncertainty decreases along a trajectory. The single-intervention experiments motivate these design choices; response-conditioned gains require separate evaluation.

Interpretation. The Bellman quantities are decision targets under a specified tool configuration, observation kernels, and terminal procedure, not observable scores. For $H = 1$, $A_1^* = A_1$ and the problem reduces to selection with one request remaining. Uncertainty reduction alone is not the reward. A more concentrated incorrect prediction can have lower utility than an uncertain but recoverable state. Conditioning the next action on observations helps only when they distinguish which action has the greatest expected value; if one action is optimal across all observable states, a fixed choice suffices. Appendix C analyzes how value-approximation error accumulates over the request horizon.

B.5 VALUE OF CONDITIONING ON RESPONSES

The static gain $\Delta_j(u)$ in Equation 25 averages over every response that source j may return. A decision made after the response is observed can instead condition on it. Fix an evidence source j with request a_j and a set of inputs on which it is invoked, so that the invocation cost is the same for the rules compared below. Write $\Delta = R(z_j; x) - R(z_0; x)$ and consider two information sets, the pre-evidence state $\mathcal{I}_{\text{pre}} = \sigma(u_0)$ and the post-evidence state $\mathcal{I}_{\text{post}} = \sigma(u_0, z_0, o_j, z_j) \supseteq \mathcal{I}_{\text{pre}}$. An *adoption rule* is a random variable $\pi \in [0, 1]$, measurable with respect to the information it may use, giving the probability of answering z_j instead of z_0; its expected utility is $\mathbb{E}[R(z_0; x)] + \mathbb{E}[\pi\Delta]$.

Proposition 1 (Value of conditioning on the response)**.** *Let* $Y = \mathbb{E}[\Delta \mid \mathcal{I}_{\mathrm{post}}]$*, and write* $Y_+ = \max\{Y, 0\}$ *and* $Y_- = \max\{-Y, 0\}$*. Then*

$$\sup_{\pi \in \mathcal{I}_{\mathrm{pre}}} \mathbb{E}[\pi\Delta] = \mathbb{E}\big[(\mathbb{E}[\Delta \mid \mathcal{I}_{\mathrm{pre}}])_+\big] \;\le\; \mathbb{E}[Y_+] = \sup_{\pi \in \mathcal{I}_{\mathrm{post}}} \mathbb{E}[\pi\Delta] \;\le\; \mathbb{E}[\Delta_+], \tag{31}$$

where $\pi \in \mathcal{I}_{\mathrm{pre}}$ *denotes a* $\mathcal{I}_{\mathrm{pre}}$*-measurable rule. The first gap equals* $\mathbb{E}\big[\min\{\mathbb{E}[Y_+ \mid \mathcal{I}_{\mathrm{pre}}], \mathbb{E}[Y_- \mid \mathcal{I}_{\mathrm{pre}}]\}\big]$*, which is positive if and only if, with positive probability, inputs sharing the same* u_0 *include post-evidence states with positive expected gain and post-evidence states with negative expected gain.*

Proof. For any $\mathcal{I}_{\mathrm{pre}}$-measurable $\pi \in [0, 1]$, the tower property gives $\mathbb{E}[\pi\Delta] = \mathbb{E}[\pi\,\mathbb{E}[\Delta \mid \mathcal{I}_{\mathrm{pre}}]] \le \mathbb{E}[(\mathbb{E}[\Delta \mid \mathcal{I}_{\mathrm{pre}}])_+]$, with equality for $\pi = \mathbf{1}[\mathbb{E}[\Delta \mid \mathcal{I}_{\mathrm{pre}}] > 0]$. The same argument with $\mathcal{I}_{\mathrm{post}}$ gives the second equality. The final inequality follows from conditional Jensen's inequality, $Y_+ \le \mathbb{E}[\Delta_+ \mid \mathcal{I}_{\mathrm{post}}]$; $\mathbb{E}[\Delta_+]$ is attained by the label-informed rule $\mathbf{1}[\Delta > 0]$. Let $m = \mathbb{E}[\Delta \mid \mathcal{I}_{\mathrm{pre}}] = \mathbb{E}[Y \mid \mathcal{I}_{\mathrm{pre}}] = \mathbb{E}[Y_+ \mid \mathcal{I}_{\mathrm{pre}}] - \mathbb{E}[Y_- \mid \mathcal{I}_{\mathrm{pre}}]$. If $m \ge 0$, then $\mathbb{E}[Y_+ \mid \mathcal{I}_{\mathrm{pre}}] - m_+ = \mathbb{E}[Y_- \mid \mathcal{I}_{\mathrm{pre}}]$; if $m < 0$, then $\mathbb{E}[Y_+ \mid \mathcal{I}_{\mathrm{pre}}] - m_+ = \mathbb{E}[Y_+ \mid \mathcal{I}_{\mathrm{pre}}]$. In both cases the difference is $\min\{\mathbb{E}[Y_+ \mid \mathcal{I}_{\mathrm{pre}}], \mathbb{E}[Y_- \mid \mathcal{I}_{\mathrm{pre}}]\} \ge 0$, and taking expectations gives the first inequality and the stated gap. □

The three terms in Equation 31 correspond, as population targets, to the reliability gate, response-conditioned integration, and the label-informed hindsight gate of Table 7. The empirical entries of that table are not these suprema. The reliability gate is restricted to sequence NLL thresholds over five budgets and is selected on the evaluation split. The proposition also does not show that invoking j is worthwhile, since both rules pay the same invocation cost; that decision is the stopping problem of Section 4.2. Within the method, the `answer` role generalizes an $\mathcal{I}_{\mathrm{post}}$ measurable rule. It observes the whole evidence sequence and may retain z_0 or adopt an answer supported by the responses.

C FINITE-HORIZON POLICY APPROXIMATION ANALYSIS

The value-of-information formulation defines an ideal policy, while the learned agent acts through finite observations and imperfect supervision. The following standard finite-horizon argument makes the gap between these objects explicit. It applies to a specified tool configuration (including its observation kernels), a fixed terminal procedure, and a fixed admissible action menu at each history. These quantities define the configuration under which the policies are compared. If open-ended invocations are reduced to a finite candidate set, the comparator is the optimum over that candidate set; the result does not account for beneficial invocations omitted by the generator.

Extend Equation 17 by setting $Q_b^*(h, \texttt{STOP}) = S(h)$. Let $\widehat{Q}_b(h, a)$ be an approximation of the optimal continuation value, and let $\widehat{\pi}$ choose an action maximizing $\widehat{Q}_b$ at each history. At zero remaining requests, both policies terminate with the same D_ϕ.

Proposition C.2 (Accumulation of control error)**.** *Assume the relevant maxima are attained and, for every feasible history, action, and remaining request horizon $b \in \{1, \dots, H\}$,*

$$|\widehat{Q}_b(h, a) - Q_b^*(h, a)| \le \epsilon. \tag{32}$$

Then the net utility of the induced policy satisfies

$$0 \le V_b^*(h) - V_b^{\widehat{\pi}}(h) \le 2b\epsilon. \tag{33}$$

Proof. Let a^* maximize $Q_b^*(h, \cdot)$ and $\widehat{a}$ maximize $\widehat{Q}_b(h, \cdot)$. By Equation 32,

$$Q_b^*(h, a^*) \le \widehat{Q}_b(h, a^*) + \epsilon \tag{34}$$

$$\le \widehat{Q}_b(h, \widehat{a}) + \epsilon \le Q_b^*(h, \widehat{a}) + 2\epsilon. \tag{35}$$

If $\widehat{a} = \texttt{STOP}$, the chosen action has no continuation error and the bound is at most 2ϵ. Otherwise, the immediate cost is identical in the two evaluations of $\widehat{a}$. The remaining discrepancy is the expected value loss after its observation, with at most $b - 1$ requests remaining. Induction from the common terminal value at $b = 0$ yields $2\epsilon + 2(b - 1)\epsilon = 2b\epsilon$. □

This result explains the role of a bounded request horizon. Under a uniform value approximation assumption, additional opportunities to gather evidence also introduce additional opportunities for control error. It does not establish that the trained adapter satisfies Equation 32. In particular, the error probability from PROBEDETECTOR, a HISTORYUQ statistic, and a return measured under a different terminal evaluator do not provide that guarantee. Verification on histories visited by the current policy is intended to improve the relevance of supervision to deployment; empirical evaluation is still needed to assess its accuracy and coverage.

One request remaining. With one request remaining, a tested action is followed immediately by the fixed terminal procedure.

$$Q_1^*(h, a) = -\lambda c(h, a) + \mathbb{E}_{o|h,a}[S(h \oplus (a, o))]. \tag{36}$$

This recovers selection with one remaining request as a special case. At longer horizons, replacing V_{b-1}^* with S evaluates a request followed by an immediate answer and may miss evidence that enables a useful later action. The training comparisons instead execute the first tool and then allow the frozen chooser to continue within the remaining horizon. These sampled returns include later tool choices, although they are not exact estimates of the optimal values in Equation 17.

Table 10: Full PROBEDETECTOR results. Precision and recall measure initial-error detection at the reported operating point. The Qwen2.5-VL-7B A-OKVQA row uses the corrected PROBEDETECTOR result reported in Table 1.

Data	Base VLM	AUROC	Precision	Recall
OK-VQA	Qwen2.5-VL-7B	.845	.614	.754
	InternVL3.5-8B	.862	.662	.770
	GLM-4.1V-9B	.893	.845	.728
	Llama-3.2-11B-Vision	.874	.770	.732
	Molmo-7B-D	.856	.733	.717
A-OKVQA	Qwen2.5-VL-7B	.863	.733	.732
	InternVL3.5-8B	.867	.658	.741
	GLM-4.1V-9B	.864	.802	.723
	Llama-3.2-11B-Vision	.843	.693	.743
	Molmo-7B-D	.832	.712	.680

Table 11: VQA score on unflagged inputs. Initial and final reproduce the unflagged scores in Table 3. Gain is final minus initial in percentage points, calculated from the displayed scores.

Data	Base VLM	Initial raw VLM	Final U-TRACE	Gain
OK-VQA	Qwen2.5-VL-7B	81.4	83.9	+2.5
	InternVL3.5-8B	82.0	83.4	+1.4
	GLM-4.1V-9B	74.0	79.1	+5.1
	Llama-3.2-11B-Vision	76.4	81.2	+4.8
A-OKVQA	Qwen2.5-VL-7B	83.0	84.1	+1.1
	InternVL3.5-8B	83.9	84.0	+0.1
	GLM-4.1V-9B	79.7	82.6	+2.9
	Llama-3.2-11B-Vision	83.0	83.8	+0.8

D ADDITIONAL EXPERIMENTAL RESULTS

This appendix groups analyses of routing and uncertainty, tools and policy learning, generalization, and runtime. Implementation and reproducibility details appear in the final appendix section.

D.1 PROBEDETECTOR DIAGNOSTICS

PROBEDETECTOR ranks initial answers by predicted reliability deficit using PROBESET-derived U-STATE features. At the reported operating point, precision measures the proportion of selected examples that are errors, while recall measures the proportion of all errors captured by the selected set. Table 10 reports detector discrimination and operating point precision and recall across five VLMs. High AUROC is consistent across VLMs, but does not by itself establish calibration or identify which intervention will help.

D.2 ANSWER GENERATION ON UNFLAGGED INPUTS

Table 11 derives the gains from the unflagged scores in Table 3. Initial is the raw direct answer generated by the base VLM. Final is produced by U-TRACE using the ROLEAGENT answer role conditioned on the initial answer and available probe information, followed by answer rephrasing. Both scores use the same unflagged subset, with no tool calls. The gains of 0.1 to 5.1 VQA points reflect the combined contribution of probe conditioned answer generation and rephrasing.

D.3 ROUTING CONTROLS

Table 12 compares overall VQA accuracy when routing is based on PROBEDETECTOR or individual uncertainty signals. The PROBEDETECTOR row reproduces the primary pipeline results from Table 2.

Table 12: Routing-signal comparison on OK-VQA and A-OKVQA. Values are overall direct-answer VQA accuracy. The PROBEDETECTOR results are the primary pipeline scores reported in Table 2.

Routing signal	OK-VQA	A-OKVQA
PROBEDETECTOR	70.5	73.9
Semantic entropy	67.1	71.5
Sequence NLL	68.3	70.9
Token entropy	67.8	71.7
Sample disagreement	67.8	71.6
Eigenscore	68.0	72.0
Energy score	66.8	70.4
Random	63.8	69.5

Table 13: VQA score at different routing thresholds. Data percentage is the share routed to adaptive evidence acquisition at each threshold.

Threshold	Data percentage	Overall
OK-VQA		
0.8	4%	0.6322
0.5	39%	**0.7052**
0.2	59%	0.6989
0	100%	0.6895
A-OKVQA		
0.8	9%	0.6943
0.5	37%	**0.7391**
0.2	54%	0.7258
0	100%	0.7302

D.4 ROUTING THRESHOLD SENSITIVITY

Table 13 shows how the detector operating point changes both the fraction of data sent to evidence acquisition and the final accuracy.

D.5 DETECTOR BACKBONE AND PROMPT SENSITIVITY

With features, labels, prompt window, and routing rule fixed on OK-VQA, the reported language-backbone result for PROBEDETECTOR is 0.845 AUROC. The comparison covers Phi-3.5-mini, Phi-4, Qwen2.5-7B/14B, Qwen3-8B, Llama-3.1-8B, and Mistral-NeMo. Prompt variants are similarly stable. Only Mistral-NeMo improves beyond the measured detector noise, by 0.012 AUROC, while adding a second VLM opinion reduces AUROC for six of seven backbones. Supplying more numeric features also does not help on average. Across the seven backbones, the evaluation prompt, a text-only terse prompt, 25 selected numeric signals, and a compressed score over all 350 features obtain mean AUROC of 0.8399, 0.8412, 0.8382, and 0.8406, respectively.

D.6 INITIAL AND HISTORY UNCERTAINTY ABLATIONS

Table 14 separates the contribution of uncertainty available before evidence acquisition from uncertainty updated over the evidence history. Scores are measured on the flagged subsets, where the raw base predictions obtain 0.345 on OK-VQA and 0.432 on A-OKVQA.

Both forms of uncertainty improve tool-use decisions over removing evidence uncertainty. The combined initial-and-history state performs best on both benchmarks, while history uncertainty alone retains most of the gain. This supports updating uncertainty as evidence arrives while preserving the initial behavioral state.

Table 14: Uncertainty-state ablations on the flagged subsets. The full state combines the initial uncertainty representation with history-conditioned uncertainty updates.

Policy state	OK-VQA	A-OKVQA
Raw base prediction	0.345	0.432
Full state with initial and history	**0.496**	**0.565**
No evidence uncertainty	0.452	0.512
Initial uncertainty only	0.474	0.554
History uncertainty only	0.484	0.562

Table 15: Behavior of the uncertainty update after inserting evidence reports. Percentages are computed within each report category.

Inserted report	n	Disagreement falls	Incumbent improves
Helpful	963	31.0	34.5
Supporting	1,075	22.0	4.5
Non-diagnostic	465	28.8	0.6
Misleading	1,522	29.3	1.8

D.7 Uncertainty Updates after Evidence

Table 15 measures 4,025 report insertions from logged request trajectories. The rank correlation between uncertainty change and score change is -0.137, and uncertainty decreases after 42.9% of updates that turn a correct incumbent into an incorrect one. This conditional percentage has a different denominator from the report categories in Table 15. Thus, reduced uncertainty after a response does not establish that the answer improved.

This finding supports retaining the response content and its provenance in the history. It does not support treating a decrease in scalar uncertainty as the optimization target.

D.8 Single-Tool Outcomes and RoleAgent Ablations

The five rows in each benchmark block of Table 16 are the five tool modes available to RoleAgent. They are Multiple Choice, Caption, Interpretation, Retrieval, and Few Shot. Each tool is evaluated independently on the flagged subset. The final column reports accuracy when the listed tool is always used, alongside the adaptive all-tool result.

D.9 Knowledge Source

Table 17 places published systems alongside two Qwen2.5-VL knowledge source variants. For the two U-TRACE rows, the ProbeSet configuration, ProbeDetector, and evidence requests are held fixed. Wikipedia passages are retrieved with FLMR (Lin et al., 2023); the alternative run uses knowledge generated by GPT-5. The published systems use different backbones and evaluation settings, so their rows provide context rather than a controlled comparison.

Both U-TRACE knowledge sources exceed the matched base VLM scores. Retrieval with FLMR reaches 70.5 on OK-VQA and 74.0 on A-OKVQA, while GPT-5 generated knowledge gives 70.0 and 74.2, respectively.

D.10 Tool Adaptation and Self-Correction

The shared tool adapter changes both answer quality and abstention behavior. On routed OK-VQA inputs, adapted Verify answers 91% of requests, repairs 21.3% of examples, corrupts 8.5%, and has a net repair rate of 12.8 points. The unadapted VLM Verify always answers and has a net rate of 8.1 points. The gap is larger for operations that require specialized prompting. Adapted Interpretation and Caption have net rates of +11.4 and +11.8, whereas their unadapted VLM

Table 16: Single-tool outcomes on the flagged subsets. Fixes and breaks are percentages of the flagged set; net is their difference in percentage points. The last column reports the resulting flagged-set accuracy.

Tool	Fixes	Breaks	Net	Always-tool or system accuracy
OK-VQA				
MULTIPLE CHOICE	20.3%	9.5%	+10.8	0.449
CAPTION	23.4%	11.6%	+11.8	0.455
INTERPRETATION	23.3%	11.9%	+11.4	0.453
RETRIEVAL	8.8%	3.4%	+5.4	0.397
FEW SHOT	7.5%	2.5%	+5.0	0.394
ROLEAGENT *with all tools*	-	-	-	*0.496*
A-OKVQA				
CAPTION	16.9%	10.0%	+6.9	0.527
MULTIPLE CHOICE	15.8%	11.6%	+4.2	0.502
INTERPRETATION	10.6%	7.6%	+3.0	0.482
FEW SHOT	7.4%	4.0%	+3.5	0.490
RETRIEVAL	3.7%	3.5%	+0.2	0.465
ROLEAGENT *with all tools*	-	-	-	*0.565*

Table 17: Knowledge sources and reported direct answer results. The U-TRACE rows use Qwen2.5-VL-7B and the VQA score specified in Section 5.1.

Method	Knowledge source	OK-VQA	A-OKVQA
Published systems			
KAT (Gui et al., 2022)	Wikidata and GPT-3 knowledge	54.4	-
PromptCap (Hu et al., 2023a)	GPT-3 knowledge with question guided captions	60.4	56.3
TOA (Liang et al., 2023)	GPT-3.5-turbo knowledge and visual tools	60.6	61.2
RA-VQA-v2 (Lin et al., 2023)	Google Search retrieval	62.1	-
LION (Chen et al., 2024b)	Visual tags and region features	57.3	60.9
QACap (Yang et al., 2025)	Claude 3.5 knowledge with question aware captions	68.2	66.3
ViLA (Kim et al., 2026)	GPT-4 knowledge with LLaVA-NeXT-13B visual premises	68.9	70.0
U-TRACE *knowledge sources*			
U-TRACE with FLMR	Retrieved Wikipedia passages	70.5	74.0
U-TRACE with GPT-5	GPT-5 generated knowledge	70.0	74.2

versions have net rates of -14.8 and -14.2. Thus, the adapter's learned abstention is an important part of safe tool execution.

Intrinsic self-correction does not provide the same benefit. On routed inputs, the VLM starts at 34.5; greedy self-reflection remains at 34.5, reflection with a chain of thought falls to 25.4, and reflection supplied with a retrieved passage falls to 24.8. Greedy reflection repairs 4.9% and corrupts 5.1% of answers; the latter two variants corrupt 16.5% and 18.9%, respectively. These controls indicate that merely prompting the base VLM to reconsider its answer is not a substitute for tool adaptation and conditional arbitration.

D.11 POLICY TOOL USE

Table 18 reports how often the adaptive policy uses each tool or answer action. Because an input may trigger more than one action, percentages need not sum to 100.

Table 18: Tool and answer-action use rates by benchmark.

Benchmark	Retrieval	Few shot	Interpretation	Caption	Multiple choice	Final answer
OK-VQA	10%	8%	23%	12%	29%	24%
A-OKVQA	4%	11%	9%	24%	37%	28%

Table 19: Execution-verified learning ablations on the flagged subsets.

Training variant	OK-VQA	A-OKVQA
Full execution-verified learning	**0.496**	**0.565**
Without verification	0.399	0.489
Self-diagnosed error labels	0.374	0.466
No tool-selection training	0.472	0.544
No request-generation training	0.485	0.553
Prompted agent	0.435	0.527

D.12 Execution-Verified Learning Ablations

Table 19 isolates how execution-verified learning contributes to agent training. All scores are measured on the same flagged subsets as Table 14.

The variant without verification records agent runs, keeps trajectories that end with the correct answer, and trains the agent to reproduce them. It therefore sees only what worked and never observes what an alternative tool would have produced from the same state. The self-diagnosed-error variant instead asks the frozen model to infer why its answer is wrong and maps that diagnosis to a tool, without execution feedback or fine-tuning. Both alternatives trail execution-verified learning on both benchmarks.

The component ablations show that the learned policy is not merely memorizing a fixed tool sequence. Removing tool-selection training reduces performance to 0.472/0.544, whereas removing request-generation training retains 0.485/0.553. Thus, the agent can learn both which tool to use and how to formulate its request from executed downstream outcomes, without requiring a separately constructed benchmark for each skill. Even with no agent training, a prompted agent without a trained adapter improves over the raw flagged-set predictions, but the full results show a clear additional benefit from training. Unlike the broader ablation in Table 4, this row retains agent based tool selection and answer generation.

D.13 Policy Seed Sensitivity

Across three policy-training seeds, routed accuracy is 48.9 ± 0.4, fixed-order accuracy is 47.7 ± 0.2, and the final score is 70.5 ± 0.2. The learned policy outperforms the fixed Verify→Retrieval order for every seed.

Table 20: Overall VQA score with different agent LLMs. Parameters denote the approximate size of the agent backbone.

Agent LLM	Parameters	OK-VQA	A-OKVQA
Phi-3.5-mini	3.8B	0.680	0.720
Qwen2.5-7B	7.6B	0.693	0.728
Llama-3.1-8B	8.0B	0.705	**0.739**
Qwen3-8B	8.2B	**0.707**	0.736
Mistral-NeMo	12.2B	0.693	0.724
Phi-4	14.7B	0.696	0.725

Table 21: Transfer to two additional VQA benchmarks. AUROC, precision, and recall evaluate initial error detection at the reported operating point. Calls are executed tool calls per routed input. No component is tuned for either dataset.

Dataset	n	Initial	U-TRACE	Δ	AUROC	Precision	Recall	Calls
VQA v2	5,033	81.6	**86.4**	+4.8	0.915	0.436	0.918	1.02
GQA	5,000	60.1	**66.0**	+5.9	0.825	0.699	0.701	0.68

D.14 Agent Backbone and Policy Transfer

Table 20 varies the agent LLM while retaining the remaining pipeline configuration. Overall VQA scores span 0.680 to 0.707 on OK-VQA and 0.720 to 0.739 on A-OKVQA. The differences are modest across these six agents, and parameter count does not order performance. The 8.2B Qwen3 model has the highest OK-VQA result, while the 8.0B Llama model has the highest A-OKVQA result. This comparison supports robustness within the evaluated agent choices without implying that all language models will behave alike. The Llama-3.1-8B row reproduces the primary 70.5 and 73.9 results after expressing them on the unit scale.

In a separate base VLM transfer study, direct transfer of a trained agent policy is weak. The policy trained for Qwen2.5-VL-7B obtains 40.1 on flagged GLM inputs versus 50.9 for GLM's own policy, and 40.3 on flagged Llama-3.2-11B-Vision inputs versus 49.4 for its own policy. A pooled policy narrows the gap on held out InternVL inputs, with 44.1 versus 44.6, and on Molmo inputs, with 43.4 versus 46.3.

D.15 Transfer to Additional VQA Benchmarks

We additionally run the complete pipeline on VQA v2 (Antol et al., 2015) and GQA (Hudson & Manning, 2019) without dataset specific tuning. VQA v2 uses one validation question per available COCO image with 5,033 questions. GQA uses 5,000 questions from testdev balanced. Because GQA provides one reference answer per question, we replicate it ten times solely to use the same VQA scoring implementation.

The gains on both transfer benchmarks show that the routing and acquisition pattern extends beyond knowledge-intensive VQA. The smaller VQA v2 gain is consistent with its higher initial accuracy and more limited remaining headroom.

D.16 A-OKVQA Multiple Choice

A-OKVQA also evaluates a choice among four options. For this setting, we fine tune TOOLVLM and ROLEAGENT with the options present, and the final answer is one of the four choices. The results demonstrate that our model achieves the best performance compared to previous baselines.

D.17 Runtime Profile

Table 23 gives component times for an illustrative flagged question with one tool call using the Qwen2.5-VL-7B probe and the Llama-3.1-8B agent. The probe produces ten sampled completions

Table 22: A-OKVQA multiple choice accuracy. Published model results are from Hu et al. (2024); the last two rows use the same base VLM.

Method	Backbone	Accuracy
InstructBLIP (Hu et al., 2024)	Vicuna 7B	75.7
PaLI 3 VPD specialist (Hu et al., 2024)	PaLI 3 5B	79.7
PaLI X VPD specialist (Hu et al., 2024)	PaLI X 55B	85.2
U-TRACE	Qwen2.5-VL-7B	**89.5**

Table 23: Indicative runtime for a flagged input with one tool call. Component times depend on the profiling environment and are not end to end averages over either benchmark.

Component	Operation	Work	Time
Probe VLM	Four prompts with ten samples each	≈ 40 completions	≈ 1.29 s
PROBEDETECTOR	Reliability readout $P(\text{yes})$	1 forward	≈ 0.05 s
ROLEAGENT	Generate a request with updated uncertainty	≈ 15 forwards	≈ 0.85 s
TOOLVLM	Execute one tool	≈ 1 generation	≈ 0.20 s
ROLEAGENT	Select an answer or stop	≈ 5 forwards	≈ 0.45 s
ROLEAGENT	Generate the final answer	≈ 1 forward	≈ 0.05 s
Total	One illustrative flagged question		≈ 2.9 s

under each of four prompts. These are batched with vLLM, so forty completions do not imply forty sequential requests. Agent stages and tool execution occur only after an input is flagged. A trajectory with additional tool calls can take longer.

The probe and detector account for approximately 1.34 seconds, while the agent and tool stages account for approximately 1.55 seconds in this profile. Thus, each part occupies roughly half of the shown time. Repeated generation remains a practical limitation because multiple answers are needed to characterize the uncertainty profile. vLLM batches these samples, but the profile does not establish an overall efficiency advantage. Prior work also studies the sampling burden of semantic uncertainty and cheaper readouts (Farquhar et al., 2024; Kossen et al., 2024). We intend to investigate adaptive sample allocation and lighter uncertainty estimators in future work, while checking whether they preserve downstream evidence selection and VQA score.

E Implementation and Reproducibility Details

E.1 Models, Protocol, and Comparison Systems

Benchmarks and metric. The OK-VQA train and validation splits contain 9,009 and 5,046 questions, and the A-OKVQA splits contain 17,056 and 1,075. We follow each dataset's official direct answer metric. OK-VQA follows the original VQA evaluation protocol (Marino et al., 2019)[1], which normalizes answers and averages the score over the ten leave one annotator out reference sets. For a prediction matching m of ten answers, its score is

$$R_{\mathrm{OK}}(m) = \frac{1}{10}\sum_{i=1}^{10} \min\left(1, \frac{m - \mathbf{1}\{a_i = \hat{a}\}}{3}\right). \tag{37}$$

Thus three matches receive 0.9, rather than 1.0. The official A-OKVQA direct answer evaluator instead uses $R_{\mathrm{AOK}}(m) = \min(1, m/3)$ with exact matching against the direct answers and excludes questions marked difficult for direct answer evaluation (Schwenk et al., 2022)[2]. Its multiple choice metric is exact choice accuracy. A dash in a table denotes an unreported or inapplicable value.

Evaluation overview. The fixed ProbeSet suite constructs the initial U-State representation for each input. ProbeDetector uses this representation to curate uncertain training examples and to make the initial routing decision at inference. We evaluate the same uncertainty-guided pipeline across multiple base VLMs and use the corresponding unaugmented prediction as the control in every within-framework comparison. ProbeDetector routes examples whose predicted reliability deficit exceeds its operating point to evidence acquisition. RoleAgent answers on both routes, using ToolVLM for VLM-mediated requests when further evidence is selected. The remainder of this section gives backbone specifications, probe and routing settings, implementation details, and baseline descriptions.

Implementation details. The Qwen2.5-VL-7B-Instruct main configuration uses a frozen probe (Bai et al., 2025). We apply the same onboarding recipe to InternVL3.5-8B, LLaVA-OneVision-7B, Molmo-7B-D, GLM-4.1V-9B, Llama-3.2-11B-Vision, and Qwen2-VL (Wang et al., 2025; Deitke et al., 2024; Hong et al., 2025). For each question, ProbeSet typically draws ten samples from each of four prompts, including in the main results. These are direct answer, caption then answer, rationale then answer, and the direct prompt with a blank image. Sampling uses temperature 1.0 and top-p 0.95 with batched vLLM generation. The rendered templates appear in Appendix E.2.

ProbeDetector is a fine-tuned Llama-3.1-8B LoRA (Grattafiori et al., 2024) that reads the shared numerical signals, four answer histograms, sampled answer text, one caption, and one rationale within a 640 token window. The MLP, FT-Transformer, and Set Transformer use the same numerical signal vector without these textual probe outputs. A single readout token provides $P(\mathrm{yes})$; the detector is trained for two epochs on the training split with learning rate 10^{-4} and batch size 32, using binary labels derived from the VQA score. Its LoRA rank is 16, with $\alpha = 32$ and dropout 0.05. Questions are ranked for routing by $1 - P(\mathrm{yes})$.

Detector input signals. The shared numerical vector summarizes the frozen VLM's direct generations and variation across probe contexts. For ten sampled direct answers, semantic entropy is the Shannon entropy of the empirical distribution over distinct meaning equivalent answer groups (Kuhn et al., 2023; Farquhar et al., 2024). The modal answer share is the fraction of those ten samples giving the most common answer, and sample disagreement is one minus that share. The number of distinct answers counts unique normalized direct answers. Sequence NLL is the mean negative log likelihood of the generated answer's tokens, so a larger value indicates a less probable answer under the model. Token entropy averages next token distribution entropy before each sampled token over positions in each answer and then over the ten sampled answers. EigenScore measures dispersion through the eigenvalues of the covariance matrix of sampled answers' final hidden state embeddings (Chen et al., 2024a). Energy score averages the negative log sum exp of the output logits over generated positions,

[1] https://visualqa.org/evaluation.html

[2] https://github.com/allenai/aokvqa/blob/main/evaluation/eval_predictions.py

adapting a confidence measure from out of distribution detection (Liu et al., 2020). The epistemic component is the between context entropy term in Equation 7. It is an operational proxy for epistemic uncertainty, recording variation between the sighted probe contexts, but does not by itself diagnose why an answer is wrong.

TOOLVLM uses one jointly trained LoRA adapter on the probe's language layers, while the vision tower and base parameters remain frozen. Its five modes are MULTIPLE CHOICE over leading candidates, RETRIEVAL with the top two encyclopedia passages, FEW SHOT with dense-nearest solved training questions, INTERPRETATION through an LLM-written sub-question, and CAPTION before re-answering. The adapter has rank 16, $\alpha = 32$, and dropout 0.05, and cross-split adapters provide out-of-sample tool responses for agent training.

ROLEAGENT is a second Llama-3.1-8B LoRA with rank 16, $\alpha = 32$, and dropout 0.05. It first learns a chooser from probe generated candidates and successful training trajectories. Cross fitting on held out training histories then compares stopping with each tool followed by a full continuation under the frozen chooser. These comparisons provide tool choice and answer arbitration targets. The agent trains for two epochs in total.

Answer and routing protocol. The unaugmented base VLM uses randomly selected few shot training examples. For unflagged questions, the ROLEAGENT answer role uses the question, initial prediction, and PROBESET outputs to produce an answer without invoking its selection or request roles. For flagged questions, ROLEAGENT can request evidence and uses the same initial information and any acquired reports to form its terminal answer. A final language model rephrases either answer as a concise response in the dataset's expected style. At each of at most three steps, the learned VOI GATE either answers or proposes a tool. The tool is called when $P(\text{tool})/[P(\text{tool}) + P(\text{answer})] \geq 0.5$, and its answer is accepted when arbitration confidence exceeds 0.7. Here $P(\text{tool})$ is the select role's probability for tool actions, while $P(\text{answer})$ is its probability for the stopping action. The three-step cap is motivated by the possibility that successive reports compound uncertainty or mislead the answer; behavior beyond three steps is not evaluated here.

Comparison systems. Table 2 includes representative open-domain VQA systems together with the matched unaugmented and U-TRACE base-VLM results. PromptCap learns question-guided captions (Hu et al., 2023a). KAT combines retrieved and parametric knowledge, while RA-VQA with FLMR improves fine-grained multimodal retrieval (Gui et al., 2022; Lin et al., 2023). TOA iterates between hypothesis formation and visual verification (Liang et al., 2023), while AVIS uses an LLM planner, reasoner, and working memory for sequential visual information seeking (Hu et al., 2023b).

Scope of supplementary analyses. Unless stated otherwise, the analyses in Appendix D use Qwen2.5-VL-7B, the Llama-3.1-8B agent, the reported detector operating point, and a request horizon of three. Comparisons use their stated configurations and do not replace the primary results in Section 5.

E.2 PROMPT TEMPLATES

Figures 6 to 10 give the prompt templates used in the pipeline. Braced fields denote values filled for each question. The displayed names clarify the inputs without changing the decision and output requirements. The headings inside the first figure identify templates and are not included in the model input. The direct and blind probes use identical text, but the blind probe receives a blank image. The detector sees the numerical signals defined above as well as textual probe outputs. For the agent, a tool choice and its request are shown together although the shared model produces them through its selection and request roles. The literal output `answer` represents the stopping action; it is not a VQA answer.

```
Direct answer and blind image
{question}
Answer in 1-4 words.

Caption then answer
Describe the image in at most 12 words, then answer.
Q: {question}
Reply exactly as:
Caption: <at most 12 words>
Answer: <1-4 words>

Rationale then answer
Give a brief reason, then answer.
Q: {question}
Reply exactly as:
Reasoning: <at most 12 words>
Answer: <1-4 words>
```

Figure 6: The four frozen VLM probe contexts. The direct text is used once with the image and once with a blank image. Ten samples are drawn per context.

```
Question: {question}
Proposed answer: {initial_answer}

VLM caption: {caption}
VLM reasoning: {rationale}

Sampled answers (direct): {direct_answer_counts}
Sampled answers (after captioning): {caption_answer_counts}
Sampled answers (after reasoning): {rationale_answer_counts}
Answers WITHOUT the image: {blind_answer_counts}

Signals: {numeric_signal_values}
Given this evidence, is the proposed answer correct?
Answer yes or no.
```

Figure 7: The PROBEDETECTOR input. Each answer count field summarizes the ten samples from its corresponding probe context. The numerical field contains the shared uncertainty signals described in Appendix E.1.

```
Question about an image you cannot see
{question}

Initial vision model answer
{initial_answer}

Sampled probe responses
{sampled_probe_responses}

Uncertainty profile
{uncertainty_profile}

Accumulated evidence in the order received
{evidence_history}

Tool requests used
{requests_used} of {max_requests}

Available tools
{available_tools}

Consider what each tool report concludes and what evidence
supports it. Take account of uncertainty, abstention, or
reasoning when the report provides them. A report may help,
add nothing, or mislead. Agreement alone does not establish
correctness, and disagreement alone does not identify what
information is missing.

Choose ‘answer‘ when the available information supports your best
answer or no available tool is likely to improve it. Otherwise,
choose one available tool and specify the information it should
obtain. Base this decision on the current evidence history.

Reply with exactly one line. Use ‘answer‘ to stop requesting tools,
or reply with ‘<available_tool_name> | <specific_request>‘.
For a tool that takes no request argument, use ‘none‘.
Do not write the final answer in this step.
```

Figure 8: The RoleAgent action prompt. The displayed decision includes the source selection and, when applicable, its request. The agent may receive reports and make further decisions until it selects the stopping action or reaches the request horizon. The answer role then produces the answer.

```
You cannot inspect the image directly. Answer using the
information currently available.

Question
{question}

Initial vision model answer
{initial_answer}

Predicted reliability deficit from ProbeDetector
{reliability_deficit:.2f}

Tool reports in the order received
{evidence_history}

Consider what each report supports, including any abstention or
uncertainty it expresses. Reports may be incomplete or misleading.
Agreement alone does not prove an answer, and disagreement alone
does not identify its cause. Keep the initial answer if the reports
do not support changing it.

Choose the best answer to the question. Use familiar, concise
wording at the level of detail requested. If several answers remain
plausible, give the one best supported by the available information
without inventing details.

Reply in exactly this form.
Answer: <short answer>
Basis: <one brief sentence identifying the strongest evidence>

Do not include tool requests or additional fields.
```

Figure 9: The ROLEAGENT terminal answering prompt for both routing branches. For unflagged questions, the tool report field is empty. The resulting answer and basis are passed to the final rephrasing step.

```
You are the final answer rephrasing step. Use all information
available for this question, whether or not a tool was called.

Question
{question}

Tool reports
{evidence_history}

Current answer and available basis
{candidate_answer_and_available_basis}

Examples
{few_shot_examples}

Use a common word or short phrase at the level of detail requested.
Do not add an explanation. Always provide a concrete answer, even
when the evidence is limited.

Reply with exactly one of the following.

keep
switch: <lowercase short answer>

Use keep when the answer is already suitably phrased. Otherwise
use switch with the final wording. Do not include anything else.
```

Figure 10: The final answer rephrasing prompt used for routed and unflagged questions. Both branches supply the ROLEAGENT answer and basis. The answer role uses probe information before rephrasing, while the report field is empty when no tool was called. The control words are removed before the final answer is scored.