BAYESIAN EVIDENCE FUSION FOR UNCERTAINTY QUANTIFICATION OF AI AGENTS

Anonymous authors
Paper under double-blind review

ABSTRACT

AI agents complete tasks through sequences of diverse actions: large language model (LLM) generations, tool calls, and revisions. Existing uncertainty quantification (UQ) techniques typically assess the reliability of individual LLM generations, leaving open how to aggregate and propagate uncertainty across different types of actions within an agent trajectory. We formulate UQ for LLM agents as Bayesian belief tracking over final task correctness, integrating generation-level uncertainty with tool outcomes through Bayesian updates. Our framework requires no changes to the agentic harness or LLM, and can be applied post hoc without additional agent executions, thus introducing negligible latency. We conduct an extensive evaluation across five LLMs, eight agent harnesses, and 15 agentic environments spanning coding, retrieval, and interactive tasks. The basic Bayesian fusion achieves the highest average outcome-ranking scores on coding and retrieval, while a variant that separately models trajectory length, history, and final-step evidence leads on interactive tasks. Bayesian fusion generalizes well across models, harnesses, and datasets within the same domain. For larger environment and domain shifts, adaptation with little or no labeled target data improves performance.

1 INTRODUCTION

Large language models (LLMs) can produce plausible but incorrect responses. Uncertainty quantification (UQ) aims to assess their reliability, enabling systems to flag potential errors, abstain, or request verification. For individual LLM generations, existing methods derive uncertainty from predictive probabilities, semantic disagreement across sampled responses, and model self-assessment (Malinin & Gales, 2021; Kuhn et al., 2023; Tian et al., 2023). Effectiveness of these signals depends on the task, output length, and computational budget (Vashurin et al., 2025).

UQ is more complex in AI agents, where an LLM completes a task through a sequence of generations and tool calls (Yao et al., 2023; Shinn et al., 2023). Here the reliability of individual generations does not determine the reliability of the final outcome. Generation-level uncertainty provides evidence about the reliability of LLM outputs, whereas tests, critic judgments, and environment feedback provide partial observations of task progress. For example, a coding agent may confidently generate incorrect code, yet test failures can expose its errors. However, on contrary, passing an incomplete test suite does not guarantee that the final solution is correct. These observations therefore provide imperfect evidence of final task success, and feedback from intermediate steps can play a different role from evidence at the final step.

Neither uncertainty in the final generation alone nor an unweighted average of intermediate uncertainty scores necessarily captures these distinctions. Reliable assessment of the final outcome therefore requires a principled aggregation of generation-level uncertainty and tool feedback across the trajectory. To be practical, this aggregation should also be computationally efficient, minimizing model calls and environment interactions. Recent uncertainty-propagation methods address aspects of this sequential setting (Zhao et al., 2025; Duan et al., 2025), but effectively combining generation-level uncertainty with tool feedback across various agent configurations remains an open challenge.

In this work, to address this challenge, we propose a meta-model that aggregates diverse evidence into a trajectory-level score using a Bayesian approach: observations update a belief about task cor-

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.

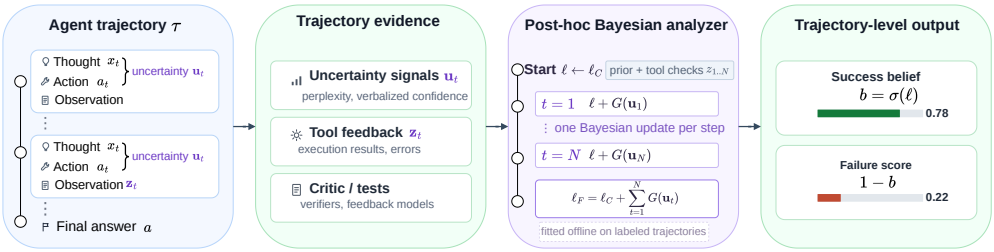

Figure 1: **Overview of our Bayesian framework** for uncertainty quantification of agent outcomes.

rectness according to their likelihood under successful and unsuccessful outcomes (Särkkä, 2013). We fit these likelihoods using trajectory-level outcome labels. Our fusion estimators – termed "Bayes Fused" – combine tool feedback with generation-level uncertainty, including variants that distinguish history from the final generation. A further variant, Structured Bayesian Fusion (SBF), takes into account trajectory length and recalibrates accumulated history evidence. The estimators operate post hoc on recorded rollouts without changing the agent's policy or requiring additional agent executions. Their scores can help determine whether to accept an agent's output, send it for human review, or perform additional verification.

We evaluate outcome ranking across 15 agentic environments: five for coding, four for retrieval, and six for interactive tasks spanning customer-service interactions, web and desktop use, and embodied planning. The evaluation covers five LLMs and 80 benchmark-model-harness configurations, including both in-distribution performance and OOD generalization (Sections 5.1 and 5.2).

Our main contributions are as follows:

- We formulate UQ for LLM agents as Bayesian belief tracking over final task correctness. Our estimators combine generation-level uncertainty with tool feedback without fine-tuning the agent model. A tool-only variant requires neither token probabilities nor additional LLM calls when tool outcomes are already recorded.
- We compare generation-level signals, trajectory summaries, and fitted predictors across 15 agentic environments spanning coding, retrieval, and interactive tasks. Bayesian fusion achieves the highest average outcome-ranking scores on coding and retrieval, while Structured Bayesian Fusion leads on interactive tasks.
- We evaluate OOD generalization across models, harnesses, datasets, environments, and domains, and introduce label-free and few-shot adaptation to address larger shifts. Averaged across the evaluated within-domain shifts, Bayes Fused largely preserves ranking performance, while logistic regression degrades substantially. Unlabeled target trajectories improve transfer between WebArena (Zhou et al., 2023) and OSWorld (Xie et al., 2024). For the more challenging transfer between ALFWorld (Shridhar et al., 2021) and DeepSWE (Huang et al., 2026), unlabeled data alone are insufficient, but approximately 10-15 labeled target trajectories enable Bayes Fused to outperform raw uncertainty scores.

2 Related Work

Uncertainty quantification for LLMs. Much work on UQ for LLMs focuses on the reliability of individual generations. Common approaches derive uncertainty from predictive probabilities (Malinin & Gales, 2021), verbalized confidence (Tian et al., 2023), or semantic disagreement across sampled responses (Kuhn et al., 2023; Lin et al., 2024). Other methods examine the factual content of a response closely. CCP estimates uncertainty about individual claim values (Fadeeva et al., 2024), while FRANQ conditions factuality estimation on faithfulness to retrieved evidence (Fadeeva et al., 2026). For agents, generation-level estimates provide evidence about individual steps, but leave open how to combine that evidence with tool observations to assess the final outcome.

Uncertainty in agent control. One way to use uncertainty in an agent is to let it guide decisions during execution. For example, KnowNo uses conformal prediction to determine when to seek help, providing task-completion guarantees under its assumptions (Ren et al., 2023). UALA similarly uses uncertainty to regulate tool use and assistance (Han et al., 2024). Uncertainty can also inform

how an agent revisits its earlier decisions. Agentic Uncertainty Quantification retains verbalized confidence in memory and uses it to trigger reflection, reporting improvements in task performance and trajectory-level calibration (Zhang et al., 2026a). In these approaches, uncertainty estimation is part of the agent's behavior and can influence the course of a rollout.

Uncertainty about agent outcomes. A related line of work asks how uncertainty at individual steps translates into uncertainty about the final outcome. SAUP addresses this question by aggregating stepwise uncertainty with situational weights (Zhao et al., 2025), while UProp uses trajectory-dependent sampling to estimate uncertainty inherited from earlier decisions (Duan et al., 2025). Holistic Trajectory Calibration (HTC) extracts temporal and structural features from token-confidence traces and fits an interpretable calibrator for trajectory success (Zhang et al., 2026b). Recent studies show that the effectiveness of generation-level UQ on tool-use trajectories depends on the aggregation method and sampling budget (Bouchard & Chauhan, 2026), and that step-level calibration does not imply trajectory-level calibration (Lin et al., 2026). Bayesian networks offer another approach to modeling uncertainty in agent workflows. Applications to agentic retrieval report limitations in calibration (Donaldson et al., 2026), while work on multi-agent workflows highlights the difficulty of modeling iterative revisions (Custers & Aslansefat, 2026).

Our work complements uncertainty propagation across generated decisions by explicitly modeling tool outcomes as evidence of final correctness, which we combine with generation-level uncertainty through Bayesian updates using recorded trajectories without additional rollouts or policy changes.

3 Bayesian Evidence Fusion for Agent Outcomes

3.1 Problem Formulation

We consider a recorded trajectory $\tau = (o_0, a_1, o_1, \ldots, a_N, o_N)$ generated by an LLM agent whose policy is fixed independently of the uncertainty estimator. At step t, the agent produces an action or response a_t and receives feedback o_t. The evidence available to the estimator may include generation-level uncertainty signals $\mathbf{x}_t$ and outcomes from tools or critics. Let $Y \in \{0, 1\}$ indicate whether the task is ultimately completed successfully. Here, N denotes the number of steps available for uncertainty estimation. If the environment's terminal action reveals the evaluation outcome, we assess the trajectory using only the preceding steps.

We use a meta-model to map the recorded evidence $\mathcal{E}(\tau)$ to a score $\widehat{p}_\theta(\tau) = f_\theta(\mathcal{E}(\tau))$ that estimates the probability of final task success, $P(Y = 1 \mid \mathcal{E}(\tau))$. We fit the meta-model using trajectory-level outcome labels from training data and apply it to recorded rollouts without changing the agent's policy. Our evaluation focuses on ranking performance, assessing whether higher failure scores correspond to poorer final outcomes. To prevent outcome leakage, the estimator's inputs exclude the final evaluation label and any observations that directly reveal it.

3.2 Bayesian Estimation of Trajectory Success

We construct the meta-model using Bayesian inference, treating final task success and failure as competing hypotheses. Starting from a prior probability of success, we update this belief according to how likely the observed generation uncertainty and tool feedback are under each hypothesis.

For brevity, write $\mathcal{E} = \mathcal{E}(\tau)$ for this evidence, and let $\pi = P(Y = 1)$ be the prior probability of agent success, with $0 < \pi < 1$. We define $\ell_0 = \log[\pi/(1 - \pi)]$ as the *prior log-odds of success*, the logarithm of the ratio of success to failure probabilities before incorporating trajectory evidence. We denote its estimate from training trajectories by $\widehat{\ell}_0$. In log-odds form, Bayes' rule adds the evidence's log-likelihood ratio to this prior term to obtain the posterior log-odds (Bishop, 2006):

$$\ell^\star(\mathcal{E}) = \log \frac{P(Y = 1 \mid \mathcal{E})}{P(Y = 0 \mid \mathcal{E})} = \ell_0 + \log \frac{p(\mathcal{E} \mid Y = 1)}{p(\mathcal{E} \mid Y = 0)}. \tag{1}$$

The log-likelihood ratio is positive when the observed evidence is more likely under successful outcomes, negative when it is more likely under unsuccessful outcomes, and zero when it is equally likely under both. Tool feedback and generation uncertainty can differ in form and scale: a tool check may return a binary pass/fail outcome, whereas an uncertainty score may be continuous. Their

raw values are not directly comparable. Expressing each observation through its log-likelihood ratio converts these different measurements into contributions on the same log-odds scale.

We represent trajectory evidence using multiple signals, such as a tool-check outcome or a generation uncertainty score, and summaries, such as the mean uncertainty over earlier steps. Each signal or summary forms an *evidence channel*. Let e_j denote the observed value of channel j. For each outcome $y \in \{0, 1\}$, we fit a probability mass or density function $\widehat{p}_{j,y}(e_j)$ using channel values from training trajectories with label $Y = y$. For example, this means estimating a pass probability for a binary tool check, or a mean and variance for a Gaussian model of a continuous uncertainty score, separately for each outcome class. The fitted channel contribution is $\Lambda_j(e_j) = \log[\widehat{p}_{j,1}(e_j)/\widehat{p}_{j,0}(e_j)]$.

If the channels are conditionally independent given Y, their joint log-likelihood ratio equals the sum of their individual log-likelihood ratios. Statistical dependence among evidence channels can cause additive fusion to count shared information more than once. Our estimators combine channel contributions additively. One variant first rescales and shifts the scores accumulated from tool feedback and generation uncertainty over earlier steps. These adjustments, which we call *recalibration*, are learned separately for the two history channels from labeled training trajectories.

For a trajectory τ with resulting score $\ell(\tau)$, we obtain the estimated success probability $\widehat{p}(\tau) = (1 + \exp(-\ell(\tau)))^{-1}$.

3.3 Evidence Sources and Their Aggregation

We instantiate the framework using two complementary sources of evidence: (1) feedback from tool calls, and (2) uncertainty in the model's generations. We first describe how tool feedback contributes to the belief in task success, then incorporate generation uncertainty and distinguish the roles of intermediate and final-step evidence.

Tool-only score. We construct a tool-only score ℓ_C by fitting a Bernoulli model for each check type within each outcome class and adding the resulting log-likelihood ratios to the prior log-odds. Each contribution reflects the association between the check outcome and final success, allowing negative check outcomes in trajectories that subsequently recover.

Fusion with generation uncertainty is done by adding its evidence contribution Λ_U to the tool-only score, giving $\ell_F = \ell_C + \Lambda_U$. Let u_t denote the generation uncertainty signal at step t. We consider three ways to construct Λ_U. The *all-steps* variant sums the log-likelihood ratios of u_t over all steps. The *last-only* variant uses only the log-likelihood ratio of the final value u_N. The *history-plus-last* variant models the mean of the earlier values $u_1, \ldots, u_{N-1}$ and the final value u_N separately, then adds their log-likelihood ratios. For a one-step trajectory, the history term is omitted.

To compute generation evidence, we estimate the distribution of the observed uncertainty scores u_t separately for successful and failed training trajectories. In the binned all-steps model, we estimate $\widehat{p}_y(k)$, the probability that a score falls in bin k for trajectories with final outcome $Y = y$, using smoothed counts of training scores. In the Gaussian all-steps model, we estimate a mean $\widehat{\mu}_y$ and variance $\widehat{\sigma}_y^2$ for each outcome class, giving the density $\mathcal{N}(u_t; \widehat{\mu}_y, \widehat{\sigma}_y^2)$. The last-only variant fits Gaussian models to final-step scores, while the history-plus-last variant fits separate Gaussian models to history means and final-step scores. The likelihoods under success and failure provide the ratios used in Λ_U. Appendix A.1 specifies the likelihood models and aggregation rules.

Structured Bayesian Fusion (SBF) combines three sources of evidence: trajectory length, observations from earlier steps (the history), and observations from the final step. These enter the success score as separate contributions. For a trajectory with N scoring steps, let $K = N - 1$ be the number of history steps and $\widehat{\ell}_0$ denote the estimated prior log-odds of success. We form

$$\ell_{\mathrm{SBF}} = \widehat{\ell}_0 + \Lambda_{\mathrm{len}} + \Lambda_{H,C} + \Lambda_{H,U} + \Lambda_{\mathrm{last}}, \tag{2}$$

where the trajectory-length contribution Λ_{len} reflects whether the rollout length is more typical of successful or failed trajectories. We fit Gaussian distributions to $\log(1 + N)$ for each outcome class and compute their log-likelihood ratio for the observed rollout length.

The history contributions $\Lambda_{H,C}$ and $\Lambda_{H,U}$ summarize critic feedback and generation uncertainty over steps $t < N$, respectively. For $\Lambda_{H,C}$, we fit Bernoulli distributions to per-step indicators of observed check failures within each outcome class. The uncertainty signal in $\Lambda_{H,U}$ is com-

puted by combining uncertainty scores at each step using weights learned through logistic regression on training trajectories. For $\Lambda_{H,U}$, we fit Gaussian distributions to these combined scores within each outcome class. For both history channels $r \in \{C, U\}$, we learn scales $\widehat{\beta}_{H,r}$ and centering values $\widehat{a}_{H,r}$ for affine recalibration. They transform the raw accumulated score $\Lambda^{\text{raw}}_{H,r}$ into $\Lambda_{H,r} = \widehat{\beta}_{H,r}(\Lambda^{\text{raw}}_{H,r} - \widehat{a}_{H,r})$. These four coefficients are fitted using trajectory outcome labels and out-of-fold raw scores. If $K = 0$, the trajectory has no history, so both history contributions are 0.

To construct Λ_{last}, we use logistic regression to combine generation uncertainty measures at the final step N, such as perplexity, token entropy, and verbalized confidence, into a single score. We pair this score with an indicator equal to 1 if any observed tool check returned a failure outcome at that step and 0 otherwise. We model this pair using bivariate Gaussian distributions for successful and failed trajectories, allowing correlation within each outcome class.

Appendix A.2 details how each component is calculated and how its parameters are estimated.

3.4 Statistical Dependence and History Recalibration

Trajectory length can help predict whether a task succeeds or fails. Having more history steps can also make the average uncertainty score more precise. However, when uncertainty scores across steps are correlated, they contain overlapping information, so each additional step provides less new evidence. Let h_t denote the combined generation uncertainty score at history step t. The following proposition uses an idealized Gaussian model conditional on $K = N - 1$ to quantify the resulting history evidence and establish a limitation of affine recalibration shared across lengths.

Proposition 1 (Effective history evidence under equi-correlation)**.** *For every positive history length K in the support of both outcome classes, suppose $\mathbf{h} = (h_1, \ldots, h_K)^\top$ satisfies*

$$\mathbf{h} \mid (Y = y, K) \sim \mathcal{N}\left(\mu_y \mathbf{1}_K, \sigma^2[(1-\rho)I_K + \rho \mathbf{1}_K \mathbf{1}_K^\top]\right), \qquad y \in \{0, 1\}, \tag{3}$$

where $\mathbf{1}_K$ is the vector of K ones and I_K is the $K \times K$ identity matrix. The population parameters satisfy $\sigma^2 > 0$ and $0 \le \rho < 1$, with σ^2 and ρ shared across classes. None of $\mu_0, \mu_1, \sigma^2, \rho$ depends on K. Writing $\bar{h} = K^{-1}\sum_{t=1}^{K} h_t$, let

$$\Lambda_{\text{ind}}(\mathbf{h} \mid K) = \frac{K(\mu_1 - \mu_0)}{\sigma^2}\left(\bar{h} - \frac{\mu_1 + \mu_0}{2}\right)$$

denote the log-likelihood ratio obtained under independence using the same class means and variance. The conditional history log-likelihood ratio satisfies

$$\Lambda_{H,U}(\mathbf{h} \mid K) := \log \frac{p(\mathbf{h} \mid Y = 1, K)}{p(\mathbf{h} \mid Y = 0, K)} = \frac{\Lambda_{\text{ind}}(\mathbf{h} \mid K)}{1 + (K-1)\rho}. \tag{4}$$

If $\mu_1 \neq \mu_0$, $\rho > 0$, and the common support contains at least two different positive lengths, there are no constants $\beta, \gamma \in \mathbb{R}$ shared across lengths for which $\beta\Lambda_{\text{ind}}(\mathbf{h} \mid K) + \gamma = \Lambda_{H,U}(\mathbf{h} \mid K)$ for every history vector at both lengths.

Under the assumptions of Proposition 1, independent accumulation scales the magnitude of the conditional history log-likelihood ratio by $1 + (K-1)\rho$, inflating non-zero evidence when $K > 1$ and $\rho > 0$. The number of steps may help predict success or failure, but correlated uncertainty scores provide less additional information with each new step. This motivates the separate trajectory-length and history components in SBF, as well as the recalibration of its history contributions. The claim about the slope β and offset γ in Proposition 1 shows that, under its assumptions, a common affine recalibration across lengths cannot reproduce the conditional history log-likelihood ratio, even when the population means and variance are known. SBF uses the same rescaling and shift across trajectory lengths, so its history recalibration cannot fully correct for dependence between steps under the proposition's assumptions. Nevertheless, SBF with this limited generation-history recalibration performs well in our experiments (Section 5). Note that the proposition concerns an idealized model of generation history and does not guarantee that SBF's predicted success probabilities are calibrated. Appendix A.4 gives the proof and consequences of Proposition 1.

4 Experimental Setup

Benchmarks and tasks. We consider 15 agentic environments across three task families: (1) *Retrieval* (HotpotQA (Yang et al., 2018), MuSiQue (Trivedi et al., 2022), CRAG (Yang et al., 2024b),

BrowseComp-Plus (Chen et al., 2025)); (2) *Coding* (LiveCodeBench-Hard (LCB) (Jain et al., 2025), BigCodeBench-Hard (BCB) (Zhuo et al., 2024), SWE-bench (Jimenez et al., 2024), CodeContests (CC) (Li et al., 2022), DeepSWE (Huang et al., 2026)); and (3) *Interactive* (τ^3-bench: retail, airline, telecom (Yao et al., 2024; Barres et al., 2025); WebArena-Verified (Hattami et al., 2025), OSWorld-Verified (Xie et al., 2024), ALFWorld (Shridhar et al., 2021)). They differ in actions and tools available to the agent, the meaning of tool feedback, and relationship between intermediate steps and the final answer. Environment descriptions, dataset statistics, and tools are detailed in Appendix B.

Models, agent harnesses, and evidence. We experiment with five models of different sizes: GPT-OSS-20B and GPT-OSS-120B (OpenAI, 2025), Qwen2.5-32B-Instruct (Yang et al., 2024a), Qwen3.6-35B-A3B (Qwen Team, 2026a), and Qwen3.8-27B (Qwen Team, 2026b). Each model is evaluated with several harnesses, including LangGraph ReAct (Yao et al., 2023; LangChain, Inc., 2026); CodeAgent and ToolCallingAgent from smolagents (Roucher et al., 2025); and benchmark-specific agents for DeepSWE, WebArena, τ^3-bench, and OSWorld. We evaluate 80 benchmark-model-harness configurations in total. The configurations and tools are detailed in Appendix B.

Evaluation protocol. We consider in-distribution (ID) and out-of-distribution (OOD) evaluation with five folds and three split seeds. In ID, estimators are fitted and evaluated within the same benchmark–model–harness configuration. In OOD, they are fitted on a source configuration and applied unchanged to a target configuration. We evaluate only completed trajectories, excluding those stopped by the step limit or environment failures. Appendix B.3 details the protocol. Retained trajectory statistics are provided in Appendix B.

Baselines. We compare Bayesian estimators against 11 diverse baselines, including statistical, UQ, unsupervised, and supervised methods. All baselines are detailed in Appendix B.1. An analysis of trajectory-level aggregation schemes for UQ and Bayes Fused methods is presented in Appendix E.

Metrics. We use prediction rejection ratio (PRR) as our primary evaluation measure (Malinin & Gales, 2021; Vashurin et al., 2025). PRR measures how well an uncertainty score ranks incorrect agent solutions above correct ones. It normalizes the rejection curve between random and oracle rankings, with 1 corresponding to the oracle, 0 to random ranking, and negative values indicating performance below random. We report PRR@0.5, considering rejection budgets up to 50%, as this better reflects practical settings where only the most uncertain instances are rejected. Importantly, PRR evaluates ranking quality rather than probability calibration, thus reflecting agent failure detection performance. A detailed description of PRR is provided in Appendix B.2.

5 Results

5.1 In-Distribution Performance

This setting evaluates how well each method detects agent failures when it is fitted on labeled trajectories from the same benchmark, model, and agent as those it scores. Figure 2 reports in-distribution PRR@0.5 for retrieval-based, coding, and interactive tasks, based on the model and harness configurations described in Appendix B. The trajectory-level aggregation used for each task family is specified in the captions of Tables 7 to 9 in Appendix C.1 and detailed in Appendix B.3.

Retrieval-based tasks. Bayes Fused outperforms all corresponding UQ scores, with Bayes Fused (Verb.) achieving the highest average PRR. The gain over raw Verb. is particularly high on MuSiQue (see Table 7), where PRR improves from 0.320 to 0.443. Moreover, Bayes Fused (Verb.) achieves the best PRR on HotpotQA and the second-best on MuSiQue and CRAG. The gain from tool feedback comes from evidence about the search process that the final-step uncertainty does not capture. In retrieval-based tasks, critics record whether each search call executed, returned content, and yielded documents or queries not seen earlier (Table 2). An agent whose searches return nothing new, or that cycles through repeated queries, has likely not collected the evidence required for the answer, yet it may still report high confidence. Bayes Fused weights this evidence which lowers the rank of confident answers produced without supporting retrievals. Consistently, the gain is largest where the critics are informative on their own: on MuSiQue and BrowseComp-Plus, Bayes Tool-only reaches PRR of 0.353 and 0.445. On CRAG, in contrast, the critics do not check whether new documents were retrieved, so they rank close to random (PRR of -0.020), and adding them makes Bayes Fused (Verb.) fall below raw Verb.

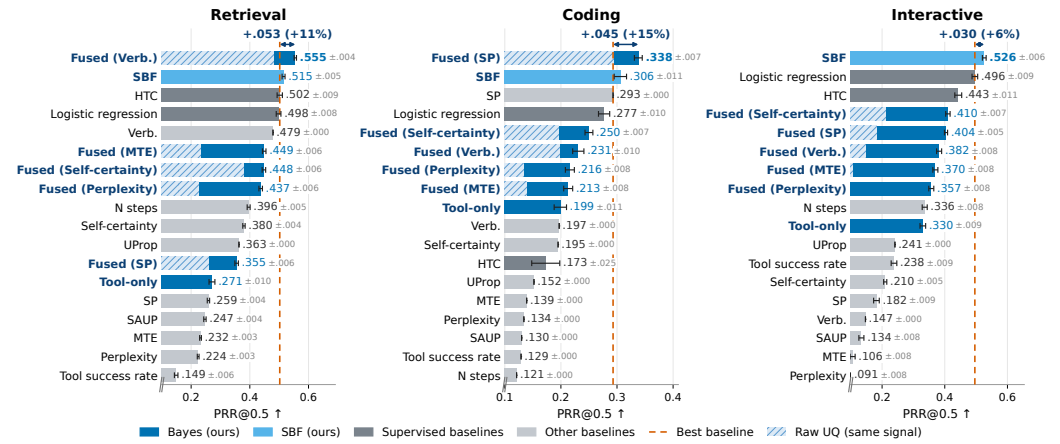

Figure 2: **In-domain results.** Average PRR@0.5↑ for each task: (1) retrieval (HotpotQA, MuSiQue, CRAG, BrowseComp-Plus); (2) coding (LCB, BCB, SWE-bench, CodeContests, DeepSWE); (3) and interactive (τ^3-bench retail, airline, and telecom; WebArena, OSWorld, and ALFWorld). Each bar shows the mean over model and harness configurations. Blue bars denote our methods, and gray bars denote baselines. For Bayes Fused, the striped portion of the bar shows the raw UQ score for the same signal, while the solid portion represents the gain from Bayesian fusion. The dashed orange line marks the best baseline. See Appendix C.1 for per-dataset results.

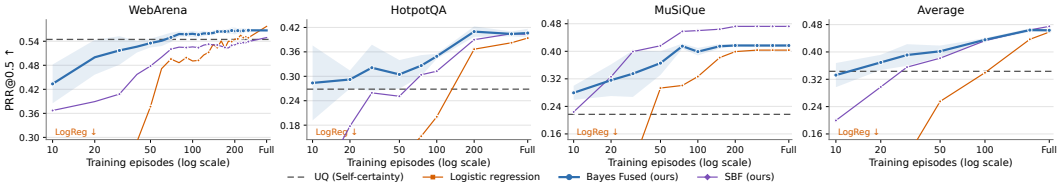

Figure 3: **Training sample efficiency.** PRR@0.5↑ by training size for Bayes Fused, logistic regression, and SBF averaged over setups, except Qwen2.5-32B-Instruct.

Coding tasks. Bayes Fused improves all UQ scores, with Bayes Fused (SP) achieving the highest average PRR (0.338), followed by Structured Bayesian Fusion (0.306). Both substantially outperform logistic regression and raw UQ methods. The gain over raw SP is largest on DeepSWE (0.017 to 0.207: see Table 8). On BigCodeBench and CodeContests, the smallest benchmarks, Bayes Fused (SP) improves over raw SP by only 0.008 PRR. On CodeContests, success rates reach 0.88, leaving only 12% of failures. On these two small benchmarks, limited training data may make parameter estimation less reliable, particularly for Structured Bayesian Fusion, with more parameters. On BigCodeBench, tool feedback is also weakly informative, which may explain why fusion barely improves on raw SP. On CodeContests, tool feedback is more informative, and both proposed methods remain close to raw SP. Overall, our method achieves stronger average performance on coding tasks, with larger training sets and informative tool feedback favoring the fusion of both signals.

Interactive tasks. Table 9 presents the results for interactive tasks. Our Structured Bayesian Fusion method achieves the highest average PRR (0.526), followed by logistic regression (0.496), with both substantially outperforming the Bayes Fused variants. Raw UQ methods perform worse on average than simple baselines such as N steps, and all except UProp also fall below tool success rate, while Bayes Fused improves the average PRR of every corresponding UQ score. The further gains from Structured Bayesian Fusion and logistic regression may reflect their more flexible use of trajectory evidence. Logistic regression jointly learns weights for multiple uncertainty and execution features, allowing it to downweight weak or redundant signals. Structured Bayesian Fusion models trajectory length, learns separate projections of uncertainty signals for the history and final step, and recalibrates the history contributions using training data. These additional features and learned adjustments may better capture the relationship between execution patterns and task success than the simpler evidence combination used by Bayes Fused.

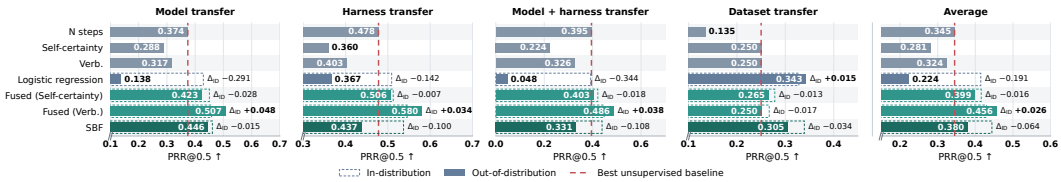

Figure 4: **Within-domain transfer.** Solid bars show OOD PRR@0.5↑ for model, harness, joint model–harness, and dataset shifts, while dashed bars show the corresponding in-distribution scores. Each bar represents average performance across the datasets in the transfer setting: (1) model transfer (WebArena, HotpotQA, and MuSiQue); (2) harness transfer (BrowseComp-Plus, HotpotQA, and MuSiQue); (3) joint model–harness transfer (HotpotQA and MuSiQue); and (4) coding dataset transfer (LCB, BCB, SWE-bench, and CodeContests). Δ denotes OOD performance minus in-distribution performance. Per-transfer results are provided in Tables 10 to 13 in Section D.2.

How many training episodes are needed? Figure 3 shows the dependence of PRR@0.5↑ on the number of training episodes for Bayes Fused (Self-certainty), logistic regression, and Structured Bayesian Fusion. These results highlight an important advantage of the proposed Bayesian framework: strong performance with few training episodes. Specifically, even with only 30–50 training episodes, Bayes Fused achieves high performance, while logistic regression is close to random with fewer than 50 episodes. With around 100 training episodes, Bayes Fused already reaches its near-optimal performance, outperforming other methods, for example, on WebArena. In contrast, logistic regression typically requires 200 training episodes to approach the performance of Bayes Fused.

Overall, these results highlight that the proposed Bayesian framework can be particularly effective when the number of training episodes is limited, making it well suited for real-world applications where collecting additional training data may be costly or impractical.

5.2 Out-of-Distribution Generalization

Transfer across models, agents, and datasets. We consider three transfer settings on a fixed dataset: (i) the same model with different agent harnesses, (ii) the same harness with different models, and (iii) different models and harnesses. We also evaluate cross-dataset transfer with the model and harness held fixed. Figure 4 summarizes the average out-of-distribution results within the same task domain. Detailed results for all four settings are provided in Appendix D.

These results indicate that Bayes Fused methods are robust to the evaluated within-domain shifts. Bayes Fused (Verb.) slightly exceeds its in-distribution PRR in every transfer setting except dataset transfer, where PRR decreases by 0.017. Averaged across transfer settings, it shows *no performance degradation*, while Bayes Fused (Self-certainty) incurs a small average PRR decrease of 0.016. By contrast, logistic regression degrades substantially under model and harness shifts, remaining stable only under dataset transfer. Logistic regression and our Structured Bayesian Fusion method show average PRR decreases of 0.191 and 0.064, respectively, suggesting greater overfitting to model- or harness-specific patterns that do not transfer reliably.

Transfer across environments and domains. For transfer across domains, we consider two settings: (i) a new environment with a fixed model, and (ii) a new domain with the same model but a different agent. Detailed results for transfer between WebArena and OSWorld, and between ALFWorld and DeepSWE, are provided in Figure 6 in Appendix D.1.

These results show that all supervised methods struggle to transfer across environments and domains. When the model remains fixed, transfer to a new environment reduces average PRR of the plotted methods by 0.076-0.227, although the methods retain useful ranking performance. Transfer to a new domain is more challenging, with PRR values near zero, indicating performance close to random ranking. These findings highlight the need for adaptation under domain shifts.

Label-free and few-shot domain adaptation. Label-free adaptation uses unlabeled target trajectories to map generation signals to the source scale for Bayes Fused and to update feature standardization for logistic regression. Few-shot adaptation additionally uses labeled target trajectories to update the signal likelihood means for Bayes Fused or logistic regression coefficients (Appendix A.3).

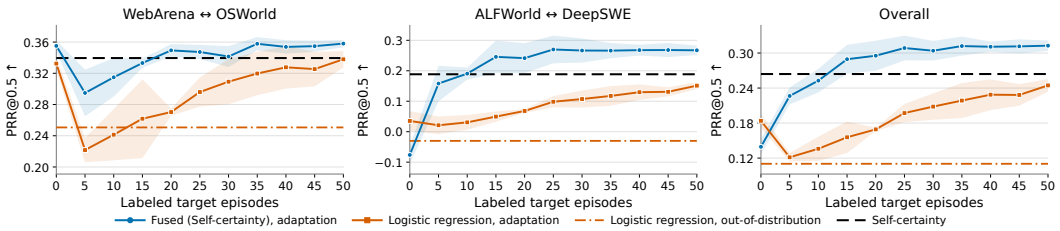

Figure 5: **Adaptation under distribution shift.** PRR@0.5↑ by the number of labeled target episodes for Bayes Fused (Self-certainty) and logistic regression, for transfer between WebArena and OSWorld and between ALFWorld and DeepSWE with the same models. Zero corresponds to label-free adaptation. Dashed lines show raw UQ (black) and logistic regression without adaptation.

Figure 5 shows the results for environment and domain shifts, the most challenging setting, where supervised methods struggle to transfer. Zero labeled episodes correspond to label-free adaptation.

Under environment transfer with the same model, label-free adaptation substantially improves both methods, bringing Bayes Fused close to its best observed PRR. Additional labeled episodes from the target environment provide little benefit for Bayes Fused. Under domain transfer, label-free adaptation is insufficient, and both methods remain close to random ranking. However, Bayes Fused benefits substantially from labeled episodes, outperforming raw UQ with 10-15 training instances. Logistic regression remains below raw UQ at this sample size.

Overall, Bayes Fused adapts effectively with little or no labeled target data, while logistic regression needs substantially more, making Bayes Fused suited to practical transfer where labels are scarce.

5.3 Analysis and Ablations

Aggregation of UQ scores over the trajectory. We compare the last step, the mean, and the maximum of raw step-level UQ scores over the trajectory (Tables 16 to 18 in Appendix E). On average, the last step works best for retrieval and interactive tasks, while the mean works best for coding. In retrieval and interactive tasks, the last step reflects the final solution, whereas coding involves iterative refinement, where every step contributes to the overall uncertainty.

Components of Bayes Fused include a prior, tool feedback and generation uncertainty. We ablate the two evidence sources independently with SP in Tables 19 to 21 in Appendix E. Tool feedback is stronger on retrieval and interactive tasks, generation uncertainty on coding, and combining them improves performance in all domains. For generation uncertainty, history + last is best on interactive tasks and close to the best (SEP) on retrieval, while continuous aggregation is best on coding.

Computational overhead. Table 22 in Appendix E reports the costs of each estimator. Applied post hoc, our methods add negligible cost: Bayes Fused and Structured Bayesian Fusion add under 0.001% to per-step inference time, except Bayes Fused (Verb.), which needs an extra LLM call.

6 Conclusion

We presented a Bayesian fusion framework that combines tool-call evidence with generation uncertainty to assess the correctness of AI agent outcomes. The framework uses likelihood ratios to quantify how strongly tool feedback supports task success or failure, then combines this evidence with generation uncertainty. SBF additionally incorporates trajectory length as evidence, recalibrates accumulated evidence from intermediate steps, and models final-step evidence separately. The basic Bayesian fusion achieves the strongest average outcome rankings on coding and retrieval tasks, while SBF leads on interactive tasks. Our method also requires fewer labeled training instances than logistic regression and maintains its performance under within-domain distribution shifts. Label-free adaptation improves performance when transferring across environments, while few-shot adaptation improves performance across domains using a small amount of labeled target data. Although no UQ method dominates every setting, these results highlight the value of incorporating evidence from tool calls and support Bayesian fusion as a data-efficient approach to UQ for agent outcomes.

AI USE STATEMENT

We have not used generative AI tools for any task with required disclosure under the ICLR 2027 AI Policy for Authors. We used them to edit the manuscript for readability and organization and to assist in writing code for aggregating experimental results, generating tables, and producing figures. We have reviewed all AI-assisted work. The code was checked and tested by the authors, and all text edits were reviewed by the authors. We take responsibility for the final content of this work, including text, claims or artifacts produced with the aid of generative AI.

ETHICS STATEMENT

This work aims to make LLM agents more reliable by detecting likely failures, which can help decide when an agent's outcome requires additional verification or human review. However, inaccurate uncertainty estimates may lead to over-reliance on incorrect outcomes or unnecessary rejection of correct ones, so the proposed estimators should complement rather than replace human oversight in high-stakes settings. All experiments use publicly available benchmarks and models, and no human subjects or personal data are involved.

REPRODUCIBILITY STATEMENT

Section 3 and Appendix A.1 specify the proposed estimators, including their likelihood models and fitting procedures, and Appendix A.3 describes label-free and few-shot adaptation. Appendix B.1 details all baselines, and Appendix B.3 describes the evaluation protocol, including cross-validation splits, random seeds, and out-of-distribution transfer. Appendices B.4, B.5, and B.6 list all environments, models, harnesses, critics, and trajectory statistics. We provide the code and scripts for reproducing all tables and figures as supplementary material.

REFERENCES

Victor Barres, Honghua Dong, Soham Ray, Xujie Si, and Karthik Narasimhan. τ^2-bench: Evaluating conversational agents in a dual-control environment, 2025. URL https://arxiv.org/abs/2506.07982.

Christopher M. Bishop. *Pattern Recognition and Machine Learning*. Springer, 2006. URL https://www.microsoft.com/en-us/research/wp-content/uploads/2006/01/Bishop-Pattern-Recognition-and-Machine-Learning-2006.pdf.

Dylan Bouchard and Mohit Singh Chauhan. Beyond single-turn confidence: Trajectory-adapted uncertainty quantification for LLM agents, 2026. URL https://arxiv.org/abs/2608.11552.

Zijian Chen, Xueguang Ma, Shengyao Zhuang, Ping Nie, Kai Zou, Andrew Liu, Joshua Green, Kshama Patel, Ruoxi Meng, Mingyi Su, Sahel Sharifymoghaddam, Yanxi Li, Haoran Hong, Xinyu Shi, Xuye Liu, Nandan Thakur, Crystina Zhang, Luyu Gao, Wenhu Chen, and Jimmy Lin. Browsecomp-plus: A more fair and transparent evaluation benchmark of deep-research agent. *arXiv preprint arXiv:2508.06600*, 2025.

Bart Custers and Koorosh Aslansefat. Runtime uncertainty monitoring for LLM-based multi-agent systems using Bayesian networks. In Elena Troubitsyna, David de Andrés, Erwin Schoitsch, Joaquín Gracia-Morán, and Friedemann Bitsch (eds.), *Computer Safety, Reliability, and Security. SAFECOMP 2026 Workshops*, volume 16869 of *Lecture Notes in Computer Science*, pp. 736–748. Springer, 2026. doi: 10.1007/978-3-032-35506-5_53. URL https://doi.org/10.1007/978-3-032-35506-5_53.

Louis Donaldson, Connor Walker, Koorosh Aslansefat, and Yiannis Papadopoulos. Bayesian uncertainty propagation for agentic RAG pipelines: A proof-of-concept study on multi-hop question answering, 2026. URL https://arxiv.org/abs/2607.00972.

Jinhao Duan, James Diffenderfer, Sandeep Madireddy, Tianlong Chen, Bhavya Kailkhura, and Kaidi Xu. UProp: Investigating the uncertainty propagation of LLMs in multi-step agentic decision-making, 2025. URL https://arxiv.org/abs/2506.17419.

Ekaterina Fadeeva, Aleksandr Rubashevskii, Artem Shelmanov, Sergey Petrakov, Haonan Li, Hamdy Mubarak, Evgenii Tsymbalov, Gleb Kuzmin, Alexander Panchenko, Timothy Baldwin, Preslav Nakov, and Maxim Panov. Fact-checking the output of large language models via token-level uncertainty quantification. In *Findings of the Association for Computational Linguistics: ACL 2024*, pp. 9367–9385. Association for Computational Linguistics, 2024. doi: 10.18653/v1/2024.findings-acl.558. URL https://aclanthology.org/2024.findings-acl.558/.

Ekaterina Fadeeva, Aleksandr Rubashevskii, Dzianis Piatrashyn, Roman Vashurin, Shehzaad Dhuliawala, Artem Shelmanov, Timothy Baldwin, Preslav Nakov, Mrinmaya Sachan, and Maxim Panov. Faithfulness-aware uncertainty quantification for fact-checking the output of retrieval-augmented generation. In *Findings of the Association for Computational Linguistics: ACL 2026*, pp. 6814–6836. Association for Computational Linguistics, 2026. doi: 10.18653/v1/2026.findings-acl.338. URL https://aclanthology.org/2026.findings-acl.338/.

Marina Fomicheva, Shuo Sun, Lisa Yankovskaya, Frédéric Blain, Francisco Guzmán, Mark Fishel, Nikolaos Aletras, Vishrav Chaudhary, and Lucia Specia. Unsupervised quality estimation for neural machine translation. *Transactions of the Association for Computational Linguistics*, 8: 539–555, 2020. doi: 10.1162/tacl_a_00330.

Jiuzhou Han, Wray Buntine, and Ehsan Shareghi. Towards uncertainty-aware language agent. In *Findings of the Association for Computational Linguistics: ACL 2024*, pp. 6662–6685. Association for Computational Linguistics, 2024. doi: 10.18653/v1/2024.findings-acl.398. URL https://aclanthology.org/2024.findings-acl.398/.

Amine El Hattami, Megh Thakkar, Nicolas Chapados, and Christopher Pal. Webarena verified: Reliable evaluation for web agents. In *Workshop on Scaling Environments for Agents*, 2025. URL https://openreview.net/forum?id=94tlGxmqkN.

Wenqi Huang, Charley Lee, Leonard Tng, and Serena Ge. Deepswe: Measuring frontier coding agents on original, long-horizon engineering tasks, 2026. URL https://arxiv.org/abs/2607.07946.

Naman Jain, King Han, Alex Gu, Wen-Ding Li, Fanjia Yan, Tianjun Zhang, Sida Wang, Armando Solar-Lezama, Koushik Sen, and Ion Stoica. LiveCodeBench: Holistic and contamination free evaluation of large language models for code. In *International Conference on Learning Representations*, 2025. URL https://mlanthology.org/iclr/2025/jain2025iclr-livecodebench/.

Carlos E Jimenez, John Yang, Alexander Wettig, Shunyu Yao, Kexin Pei, Ofir Press, and Karthik R Narasimhan. SWE-bench: Can language models resolve real-world GitHub issues? In *The Twelfth International Conference on Learning Representations*, 2024. URL https://openreview.net/forum?id=VTF8yNQM66.

Zhewei Kang, Xuandong Zhao, and Dawn Song. Scalable best-of-n selection for large language models via self-certainty. In D. Belgrave, C. Zhang, H. Lin, R. Pascanu, P. Koniusz, M. Ghassemi, and N. Chen (eds.), *Advances in Neural Information Processing Systems*, volume 38, Main Conference, pp. 19720–19745. Curran Associates, Inc., 2025. doi: 10.52202/085713-0663. URL https://proceedings.neurips.cc/paper_files/paper/2025/file/1c7eff166a8e345f664f0faa8f4e4d2e-Paper-Conference.pdf.

Lorenz Kuhn, Yarin Gal, and Sebastian Farquhar. Semantic uncertainty: Linguistic invariances for uncertainty estimation in natural language generation. In *International Conference on Learning Representations*, 2023. URL https://openreview.net/forum?id=VD-AYtP0dve.

Woosuk Kwon, Zhuohan Li, Siyuan Zhuang, Ying Sheng, Lianmin Zheng, Cody Hao Yu, Joseph E. Gonzalez, Hao Zhang, and Ion Stoica. Efficient memory management for large language model serving with pagedattention. In *Proceedings of the ACM SIGOPS 29th Symposium on Operating Systems Principles*, 2023.

LangChain, Inc. LangGraph: Low-level orchestration framework for building stateful agents. GitHub repository, 2026. URL https://github.com/langchain-ai/langgraph.

Yujia Li, David Choi, Junyoung Chung, Nate Kushman, Julian Schrittwieser, Rémi Leblond, Tom Eccles, James Keeling, Felix Gimeno, Agustin Dal Lago, Thomas Hubert, Peter Choy, Cyprien de Masson d'Autume, Igor Babuschkin, Xinyun Chen, Po-Sen Huang, Johannes Welbl, Sven Gowal, Alexey Cherepanov, James Molloy, Daniel J. Mankowitz, Esme Sutherland Robson, Pushmeet Kohli, Nando de Freitas, Koray Kavukcuoglu, and Oriol Vinyals. Competition-level code generation with alphacode. *Science*, 378(6624):1092–1097, 2022. doi: 10.1126/science.abq1158. URL https://www.science.org/doi/abs/10.1126/science.abq1158.

Moule Lin, Qizhen Lan, Shuhao Guan, Weipeng Jing, Jiexin Fan, David Gregg, and Goetz Botterweck. Uncertainty quantification for LLM agents: A taxonomy, an evaluation protocol, and an empirical study, 2026. URL https://arxiv.org/abs/2609.07395.

Zhen Lin, Shubhendu Trivedi, and Jimeng Sun. Generating with confidence: Uncertainty quantification for black-box large language models. *Transactions on Machine Learning Research*, 2024. URL https://openreview.net/forum?id=DWkJCSxKU5.

Andrey Malinin and Mark Gales. Uncertainty estimation in autoregressive structured prediction. In *International Conference on Learning Representations*, 2021. URL https://openreview.net/forum?id=jN5y-zb5Q7m.

OpenAI. gpt-oss-120b & gpt-oss-20b model card, 2025. URL https://arxiv.org/abs/2508.10925.

Qwen Team. Qwen3.6-Plus: Towards real world agents, April 2026a. URL https://qwen.ai/blog?id=qwen3.6.

Qwen Team. Qwen3.8-Max: A new bar for coding and cowork, August 2026b. URL https://qwen.ai/blog?id=qwen3.8.

Allen Z. Ren, Anushri Dixit, Alexandra Bodrova, Sumeet Singh, Stephen Tu, Noah Brown, Peng Xu, Leila Takayama, Fei Xia, Jake Varley, Zhenjia Xu, Dorsa Sadigh, Andy Zeng, and Anirudha Majumdar. Robots that ask for help: Uncertainty alignment for large language model planners. In *Proceedings of the 7th Conference on Robot Learning*, volume 229 of *Proceedings of Machine Learning Research*, pp. 661–682, 2023. URL https://proceedings.mlr.press/v229/ren23a.html.

Aymeric Roucher, Albert Villanova del Moral, Thomas Wolf, Leandro von Werra, and Erik Kaunismäki. smolagents: a smol library to build great agentic systems. GitHub repository, 2025. URL https://github.com/huggingface/smolagents.

Simo Särkkä. *Bayesian Filtering and Smoothing*. Cambridge University Press, 2013. doi: 10.1017/CBO9781139344203. URL https://doi.org/10.1017/CBO9781139344203.

Noah Shinn, Federico Cassano, Ashwin Gopinath, Karthik Narasimhan, and Shunyu Yao. Reflexion: Language agents with verbal reinforcement learning. In *Advances in Neural Information Processing Systems*, volume 36, pp. 8634–8652, 2023. URL https://proceedings.neurips.cc/paper_files/paper/2023/hash/1b44b878bb782e6954cd888628510e90-Abstract-Conference.html.

Mohit Shridhar, Xingdi Yuan, Marc-Alexandre Cote, Yonatan Bisk, Adam Trischler, and Matthew Hausknecht. ALFWorld: Aligning text and embodied environments for interactive learning. In *International Conference on Learning Representations*, 2021. URL https://openreview.net/forum?id=0IOX0YcCdTn.

Katherine Tian, Eric Mitchell, Allan Zhou, Archit Sharma, Rafael Rafailov, Huaxiu Yao, Chelsea Finn, and Christopher D. Manning. Just ask for calibration: Strategies for eliciting calibrated confidence scores from language models fine-tuned with human feedback. In *Proceedings of the 2023 Conference on Empirical Methods in Natural Language Processing*, pp. 5433–5442. Association for Computational Linguistics, 2023. doi: 10.18653/v1/2023.emnlp-main.330. URL https://aclanthology.org/2023.emnlp-main.330/.

Harsh Trivedi, Niranjan Balasubramanian, Tushar Khot, and Ashish Sabharwal. MuSiQue: Multi-hop questions via single-hop question composition. *Transactions of the Association for Computational Linguistics*, 2022.

Roman Vashurin, Ekaterina Fadeeva, Artem Vazhentsev, Lyudmila Rvanova, Daniil Vasilev, Akim Tsvigun, Sergey Petrakov, Rui Xing, Abdelrahman Sadallah, Kirill Grishchenkov, Alexander Panchenko, Timothy Baldwin, Preslav Nakov, Maxim Panov, and Artem Shelmanov. Benchmarking uncertainty quantification methods for large language models with LM-Polygraph. *Transactions of the Association for Computational Linguistics*, 13:220–248, 2025. doi: 10.1162/tacl_a_00737. URL https://aclanthology.org/2025.tacl-1.11/.

Tianbao Xie, Danyang Zhang, Jixuan Chen, Xiaochuan Li, Siheng Zhao, Ruisheng Cao, Toh Jing Hua, Zhoujun Cheng, Dongchan Shin, Fangyu Lei, Yitao Liu, Yiheng Xu, Shuyan Zhou, Silvio Savarese, Caiming Xiong, Victor Zhong, and Tao Yu. OSWorld: Benchmarking multimodal agents for open-ended tasks in real computer environments. In *Advances in Neural Information Processing Systems*, volume 37, pp. 52040–52094, 2024. doi: 10.52202/079017-1650. URL https://mlanthology.org/neurips/2024/xie2024neurips-osworld/.

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

Xiao Yang, Kai Sun, Hao Xin, Yushi Sun, Nikita Bhalla, Xiangsen Chen, Sajal Choudhary, Rongze Daniel Gui, Ziran Will Jiang, Ziyu Jiang, Lingkun Kong, Brian Moran, Jiaqi Wang, Yifan Ethan Xu, An Yan, Chenyu Yang, Eting Yuan, Hanwen Zha, Nan Tang, Lei Chen, Nicolas Scheffer, Yue Liu, Nirav Shah, Rakesh Wanga, Anuj Kumar, Wen tau Yih, and Xin Luna Dong. CRAG – comprehensive RAG benchmark. *arXiv preprint arXiv:2406.04744*, 2024b. URL https://arxiv.org/abs/2406.04744.

Zhilin Yang, Peng Qi, Saizheng Zhang, Yoshua Bengio, William W. Cohen, Ruslan Salakhutdinov, and Christopher D. Manning. HotpotQA: A dataset for diverse, explainable multi-hop question answering. In *Conference on Empirical Methods in Natural Language Processing (EMNLP)*, 2018.

Shunyu Yao, Jeffrey Zhao, Dian Yu, Nan Du, Izhak Shafran, Karthik Narasimhan, and Yuan Cao. ReAct: Synergizing reasoning and acting in language models. In *International Conference on Learning Representations*, 2023. URL https://openreview.net/forum?id=WE_vluYUL-X.

Shunyu Yao, Noah Shinn, Pedram Razavi, and Karthik Narasimhan. τ-bench: A benchmark for tool-agent-user interaction in real-world domains, 2024. URL https://arxiv.org/abs/2406.12045.

Jiaxin Zhang, Prafulla Kumar Choubey, Kung-Hsiang Huang, Caiming Xiong, and Chien-Sheng Wu. Agentic uncertainty quantification, 2026a. URL https://arxiv.org/abs/2601.15703.

Jiaxin Zhang, Caiming Xiong, and Chien-Sheng Wu. Agentic confidence calibration. In *Proceedings of the 43rd International Conference on Machine Learning*, 2026b. URL https://openreview.net/forum?id=B1ISNZQHuI.

Qiwei Zhao, Dong Li, Yanchi Liu, Wei Cheng, Yiyou Sun, Mika Oishi, Takao Osaki, Katsushi Matsuda, Huaxiu Yao, Chen Zhao, Haifeng Chen, and Xujiang Zhao. Uncertainty propagation on LLM agent. In *Proceedings of the 63rd Annual Meeting of the Association for Computational Linguistics (Volume 1: Long Papers)*, pp. 6064–6073. Association for Computational Linguistics, 2025. doi: 10.18653/v1/2025.acl-long.302. URL https://aclanthology.org/2025.acl-long.302/.

Shuyan Zhou, Frank F. Xu, Hao Zhu, Xuhui Zhou, Robert Lo, Abishek Sridhar, Xianyi Cheng, Tianyue Ou, Yonatan Bisk, Daniel Fried, Uri Alon, and Graham Neubig. Webarena: A realistic web environment for building autonomous agents, 2023.

Terry Yue Zhuo, Minh Chien Vu, Jenny Chim, Han Hu, Wenhao Yu, Ratnadira Widyasari, Imam Nur Bani Yusuf, Haolan Zhan, Junda He, Indraneil Paul, et al. Bigcodebench: Benchmarking code generation with diverse function calls and complex instructions. *arXiv preprint arXiv:2406.15877*, 2024.

A Theory and Supporting Method Details

A.1 Likelihood Models and Simple Fusion Variants

Let $c_j \in \{0, 1\}$ denote the outcome of a check of type i_j, with $c_j = 1$ indicating a pass. We model its relationship to the outcome through $\theta_{i,y} = P(c = 1 \mid Y = y, i)$, where the observation unit may be an individual check or an episode summary. For each critic type and outcome class, we estimate this probability from the training trajectories using Laplace smoothing,

$$\widehat{\theta}_{i,y} = \frac{n_{i,y}^{\text{pass}} + 1}{n_{i,y} + 2}, \tag{5}$$

where $n_{i,y}$ counts observed checks of type i in trajectories with final label y, and $n_{i,y}^{\text{pass}}$ counts those that pass. Let $\widehat{\pi} \in (0, 1)$ denote the prior success probability estimated from the training trajectories and write $\widehat{\ell}_0 = \log[\widehat{\pi}/(1 - \widehat{\pi})]$. The additive tool score is

$$\ell_C = \widehat{\ell}_0 + \sum_j \left[c_j \log \frac{\widehat{\theta}_{i_j,1}}{\widehat{\theta}_{i_j,0}} + (1 - c_j) \log \frac{1 - \widehat{\theta}_{i_j,1}}{1 - \widehat{\theta}_{i_j,0}} \right]. \tag{6}$$

A pass supports success when it is more frequent among successful trajectories, and its contribution is small when their ratio is close to one. A failed check can also support success if such failures occur more frequently in trajectories that eventually succeed. The sum uses a conditional independence approximation for the chosen observations. We consider implementations that include each episode summary once, implementations that sum individual check contributions across steps, and implementations that average the per-step totals over the scoring steps. Averaging produces a tempered evidence score. The observation unit and accumulation rule therefore form part of the estimator specification. Missing checks are excluded from both the counts and the score, leaving any information carried by their absence unmodeled.

To incorporate generation uncertainty, let u_t denote a scalar signal observed at each scoring step $t = 1, \ldots, N$. We consider a variant based on thresholding, denoted SEP, that maps each value to one of two bins through $b_{\text{bin}} : \mathbb{R} \to \{0, 1\}$, using a threshold selected from the training data. Let $\widehat{p}_y(k)$ denote the probability of bin $k \in \{0, 1\}$ estimated from smoothed frequencies of signal values in training trajectories with outcome y. The fused score is

$$\ell_{\text{SEP}} = \ell_C + \sum_{t=1}^{N} \log \frac{\widehat{p}_1(b_{\text{bin}}(u_t))}{\widehat{p}_0(b_{\text{bin}}(u_t))}. \tag{7}$$

The continuous variant models the signal distribution within each outcome class by a Gaussian density, with mean $\widehat{\mu}_y$ and positive variance $\widehat{\sigma}_y^2$ estimated from the training observations. The contribution of a signal value u is

$$\begin{aligned} G(u) &= \log \frac{\mathcal{N}(u; \widehat{\mu}_1, \widehat{\sigma}_1^2)}{\mathcal{N}(u; \widehat{\mu}_0, \widehat{\sigma}_0^2)} \\ &= \log \frac{\widehat{\sigma}_0}{\widehat{\sigma}_1} + \frac{(u - \widehat{\mu}_0)^2}{2\widehat{\sigma}_0^2} - \frac{(u - \widehat{\mu}_1)^2}{2\widehat{\sigma}_1^2}, \end{aligned} \tag{8}$$

giving

$$\ell_{\text{cont}} = \ell_C + \sum_{t=1}^{N} G(u_t). \tag{9}$$

Both variants add one contribution per scoring step to the tool score ℓ_C, which already includes the prior. The sign of each contribution follows from the fitted likelihoods, allowing the same

construction to accommodate signals whose larger values indicate either confidence or uncertainty. Discretization discards variation within bins, and continuous fusion depends on the adequacy of the Gaussian approximation. All thresholds, bin probabilities, and Gaussian parameters are estimated using training trajectories and reused unchanged during evaluation.

We also consider estimators based on the final signal value and a summary of the preceding values. The last-only variant uses the Gaussian log-likelihood ratio in equation 8, with its parameters estimated from final-step values in the training trajectories. Denoting this fitted function by G_{last}, the score is

$$\ell_{\text{last}} = \ell_C + G_{\text{last}}(u_N). \tag{10}$$

For $N > 1$, the variant based on the joint use of the history and last step summarizes the preceding values by $\bar{u}_{<N} = (N-1)^{-1} \sum_{t=1}^{N-1} u_t$ and fits separate Gaussian models for the history mean and the final value. Let $r \in \{H, L\}$ index these two roles. Within each role, we estimate class means $\widehat{\mu}_{r,y}$ and a common variance $\widehat{s}_r^2$ pooled within the outcome classes, using one available summary per training trajectory. The history model is fitted using only trajectories with $N > 1$. Sharing the variance across classes gives the linear log-likelihood ratio

$$\begin{aligned} G_r(x) &= \frac{(x - \widehat{\mu}_{r,0})^2 - (x - \widehat{\mu}_{r,1})^2}{2\widehat{s}_r^2} \\ &= \frac{\widehat{\mu}_{r,1} - \widehat{\mu}_{r,0}}{\widehat{s}_r^2} \left(x - \frac{\widehat{\mu}_{r,1} + \widehat{\mu}_{r,0}}{2} \right), \; r \in \{H, L\}. \end{aligned} \tag{11}$$

The resulting score is

$$\ell_{\text{H+L}} = \ell_C + G_H(\bar{u}_{<N}) + G_L(u_N). \tag{12}$$

Each summary contributes once with unit weight. The history variance is estimated across trajectory means, and its contribution has no additional multiplier for the number of steps. When $N = 1$, the history term is omitted.

A.2 Structured Bayesian Fusion Components

Structured Bayesian Fusion includes a separate contribution for trajectory length, measured by the number N of scoring steps. We use the transformed length $d = \log(1 + N)$ and approximate its distribution within each outcome class by a Gaussian density. From the training trajectories, we estimate class means $\widehat{\mu}_{d,y}$ and a positive variance $\widehat{s}_d^2$ pooled within the outcome classes, using one length value per trajectory. The resulting contribution is

$$\begin{aligned} \Lambda_{\text{len}} &= \log \frac{\mathcal{N}(d; \widehat{\mu}_{d,1}, \widehat{s}_d^2)}{\mathcal{N}(d; \widehat{\mu}_{d,0}, \widehat{s}_d^2)} \\ &= \frac{(d - \widehat{\mu}_{d,0})^2 - (d - \widehat{\mu}_{d,1})^2}{2\widehat{s}_d^2}. \end{aligned} \tag{13}$$

This term appears once per trajectory and captures the association between its length and outcome. Its dependence on length follows from the difference between the fitted class means, allowing longer trajectories to provide evidence towards either success or failure. The association can change across task domains, agent stopping policies, and step budgets, so the predictive value of trajectory length may vary under transfer.

For critic history, let $z_t = 1$ when at least one observed check at scoring step t returns a failure outcome, and $z_t = 0$ otherwise. The history consists of the $K = N - 1$ steps preceding the final step, and $F = \sum_{t=1}^{K} z_t$ counts steps with negative critic feedback. Conditional on the outcome class and history length, we use a working model of independent Bernoulli indicators with a common probability q_y of negative critic feedback for class y. We estimate this probability from the training histories using Laplace smoothing, $\widehat{q}_y = (n_y^{\text{fail}} + 1)/(n_y^{\text{hist}} + 2)$, where n_y^{hist} counts history steps in training trajectories with outcome y, and n_y^{fail} counts those with $z_t = 1$. The accumulated contribution is

$$\Lambda_{H,C}^{\text{raw}} = F \log \frac{\widehat{q}_1}{\widehat{q}_0} + (K - F) \log \frac{1 - \widehat{q}_1}{1 - \widehat{q}_0}. \tag{14}$$

This summary pools critic types and retains one indicator per step, so multiple failure outcomes at the same step are counted once. Steps with no observed checks and steps with only passing checks both

have $z_t = 0$ and contribute through the $K - F$ term. Thus, zero represents the absence of negative critic feedback, which can include missing feedback. When $K = 0$, the history contribution is zero. The superscript raw identifies the accumulated evidence before the history recalibration described later in this subsection.

For generation history, let $\mathbf{x}_t$ collect the available uncertainty signals at step t, and let $\widetilde{\mathbf{x}}_t$ denote their standardized and clipped values, using preprocessing parameters learned from the training trajectories. We form a scalar projection $h_t = \widehat{\mathbf{w}}_H^\top \widetilde{\mathbf{x}}_t$, where $\widehat{\mathbf{w}}_H$ is fitted using logistic regression with ridge regularization and the trajectory outcomes as labels. For $K > 0$, we use a working model in which the projected history values are independent Gaussian observations conditional on the outcome class and history length. We estimate their class means $\widehat{\mu}_{h,y}$ and a positive variance $\widehat{s}_h^2$ pooled within the outcome classes from projected training history steps. Under this model, the mean $\bar{h}_{<N} = K^{-1} \sum_{t=1}^{K} h_t$ has variance $\widehat{s}_h^2/K$, giving the accumulated contribution

$$\begin{aligned} \Lambda_{H,U}^{\mathrm{raw}} &= \log \frac{\mathcal{N}(\bar{h}_{<N}; \widehat{\mu}_{h,1}, \widehat{s}_h^2/K)}{\mathcal{N}(\bar{h}_{<N}; \widehat{\mu}_{h,0}, \widehat{s}_h^2/K)} \\ &= K \frac{(\bar{h}_{<N} - \widehat{\mu}_{h,0})^2 - (\bar{h}_{<N} - \widehat{\mu}_{h,1})^2}{2\widehat{s}_h^2}, \end{aligned} \tag{15}$$

where $\widehat{s}_h^2$ describes variation among projected step values within each outcome class. The history variance $\widehat{s}_H^2$ in equation 11 instead describes variation among trajectory means of the selected signal. The factor K follows from the working independence assumption and determines how strongly the history mean contributes to the score. Dependence between steps can reduce the information provided by additional observations, motivating the history recalibration described later in this subsection. When $K = 0$, the history contribution is zero and its mean is not evaluated.

For the final step, we form a separate projection $r_N = \widehat{\mathbf{w}}_{\mathrm{last}}^\top \widetilde{\mathbf{x}}_N$, with coefficients $\widehat{\mathbf{w}}_{\mathrm{last}}$ fitted by logistic regression with ridge regularization using final-step training observations and trajectory outcomes. We combine this projection with the indicator z_N of negative critic feedback at the final step to form $\mathbf{v}_N = (r_N, z_N)^\top$. We fit a bivariate Gaussian model with class mean vectors $\widehat{\boldsymbol{\mu}}_{\mathrm{last},y}$ and a shared positive definite covariance matrix $\widehat{\Sigma}_{\mathrm{last}}$ estimated from variation within the outcome classes. Under this model, the final-step contribution is

$$\Lambda_{\mathrm{last}} = \left(\widehat{\boldsymbol{\mu}}_{\mathrm{last},1} - \widehat{\boldsymbol{\mu}}_{\mathrm{last},0}\right)^\top \widehat{\Sigma}_{\mathrm{last}}^{-1} \left[\mathbf{v}_N - \frac{\widehat{\boldsymbol{\mu}}_{\mathrm{last},1} + \widehat{\boldsymbol{\mu}}_{\mathrm{last},0}}{2}\right]. \tag{16}$$

The covariance allows the projected uncertainty signal and the critic indicator to be correlated within outcome classes. Because z_N is binary, the Gaussian model serves as a discriminant approximation for this combination of continuous and binary evidence. The contribution appears once per trajectory, including when $N = 1$ and both history channels are empty.

We recalibrate the two history channels separately to adjust the scale and location of their accumulated evidence. For channel $r \in \{C, U\}$, corresponding to critic and generation history, respectively, we use

$$\Lambda_{H,r} = \begin{cases} \widehat{\beta}_{H,r}\left(\Lambda_{H,r}^{\mathrm{raw}} - \widehat{a}_{H,r}\right), & K > 0, \\ 0, & K = 0, \end{cases} \tag{17}$$

where $\widehat{\beta}_{H,r}$ is the fitted slope and $\widehat{a}_{H,r}$ is a centering parameter on the raw score scale. These parameters are estimated separately for each channel using out-of-fold raw scores and the corresponding trajectory outcomes within the training data. This adjustment can temper excessive accumulation when dependence between steps causes the likelihood model to overstate the information in a history. The same fitted transformation applies across positive history lengths, so it provides a common affine adjustment whose adequacy may vary with length. For an empty history, the entire contribution is zero, including the offset. We use $\Lambda_{H,C}$ and $\Lambda_{H,U}$ as adjusted evidence contributions in the combined score. Recalibrating them separately does not ensure independence between channels or calibration of the resulting probability.

A.3 Label-Free and Few-Shot Adaptation

The adaptation experiments extend the source-trained estimators using observations from a target configuration. We describe the transformations used for Bayes Fused and logistic regression.

Label-free adaptation. Let $\widehat{\mu}_S$ and $\widehat{s}_S$ denote the mean and standard deviation of a scalar generation signal computed from source training trajectories, and let $\widehat{\mu}_T$ and $\widehat{s}_T$ denote the corresponding quantities computed from unlabeled target trajectories. For Bayes Fused, a target signal value u_t is mapped to the source scale through

$$\widetilde{u}_t = \widehat{\mu}_S + \frac{\widehat{s}_S}{\widehat{s}_T}(u_t - \widehat{\mu}_T), \qquad \widehat{s}_T > 0.$$

This transformation matches the target signal's empirical mean and standard deviation to those of the source reference. For logistic regression, input features are standardized using target statistics in place of source statistics. The source likelihood parameters and regression coefficients remain unchanged.

Few-shot adaptation. Using labeled target trajectories, Bayes Fused additionally updates the class means of its signal likelihood while retaining the other source parameters. Let $\widehat{\mu}_y$ denote a source class mean. For each target trajectory, average its rescaled signal values over its scoring steps. Within outcome class y, let n_y be the number of labeled target trajectories, and let $\bar{u}_{T,y}$ and $s^2_{T,y}$ be the sample mean and sample variance of these trajectory averages. For $n_y \geq 2$, the update is

$$\begin{aligned}
\widehat{\mu}'_y &= \widehat{\mu}_y + \alpha_y(\bar{u}_{T,y} - \widehat{\mu}_y), \\
\widehat{\tau}^2_y &= \max\left\{0, (\bar{u}_{T,y} - \widehat{\mu}_y)^2 - \frac{s^2_{T,y}}{n_y}\right\}, \\
\alpha_y &= \frac{\widehat{\tau}^2_y}{\widehat{\tau}^2_y + s^2_{T,y}/n_y},
\end{aligned}$$

whenever the denominator defining α_y is positive. The quantity $\widehat{\tau}^2_y$ is a non-negative estimate of the squared discrepancy between the target and source means after subtracting an estimate of the target mean's sampling variance. It determines how far the update moves towards the target estimate. For logistic regression, adaptation refits the coefficients using labeled target trajectories with an L_2 penalty on their deviation from the source coefficients.

A.4 Proof and Consequences of Proposition 1

We examine how dependence between steps affects the evidence provided by the scalar generation history $\mathbf{h} = (h_1, \ldots, h_K)^\top$, where $K = N - 1$. For each positive history length with $P(K \mid Y = y) > 0$ for both $y \in \{0, 1\}$, assume

$$\begin{aligned}
\mathbf{h} \mid (Y = y, K) &\sim \mathcal{N}(\mu_y \mathbf{1}_K, \Sigma_K), \\
\Sigma_K &= \sigma^2[(1-\rho)I_K + \rho \mathbf{1}_K \mathbf{1}_K^\top],
\end{aligned}$$

where $\mathbf{1}_K$ is the vector of K ones and I_K is the $K \times K$ identity matrix. Each history value has class mean μ_y and variance $\sigma^2 > 0$, and every pair of different history values has correlation $\rho \in [0, 1)$ conditional on the outcome and length. The covariance is shared across outcome classes, and $\mu_0, \mu_1, \sigma^2, \rho$ do not depend on K. The parameters are treated as population quantities, and the calculation therefore ignores uncertainty arising from their estimation. Write $\Delta = \mu_1 - \mu_0$, $m = (\mu_1 + \mu_0)/2$, and $\bar{h} = K^{-1}\sum_{t=1}^K h_t$, so that $\bar{h}$ is the history mean denoted by $\bar{h}_{<N}$ in the estimator. The Gaussian assumption concerns the distribution conditional on the realized history length. When length is determined by an adaptive stopping rule, the assumption must hold after conditioning on the resulting length.

Proof of Proposition 1. Consider any positive history length K in the support of both outcome classes. For every nonzero $\mathbf{v} \in \mathbb{R}^K$,

$$\mathbf{v}^\top \Sigma_K \mathbf{v} = \sigma^2(1-\rho)\|\mathbf{v}\|_2^2 + \sigma^2\rho(\mathbf{1}_K^\top \mathbf{v})^2 > 0,$$

since $\sigma^2 > 0$ and $0 \leq \rho < 1$. Thus, Σ_K is positive definite, and both conditional Gaussian distributions have positive densities on $\mathbb{R}^K$. Their shared covariance makes the normalizing factors

cancel in the density ratio. Expanding the quadratic forms gives

$$\begin{aligned}\Lambda_{H,U}(\mathbf{h} \mid K) &:= \log \frac{p(\mathbf{h} \mid Y=1, K)}{p(\mathbf{h} \mid Y=0, K)} \\ &= \frac{1}{2}(\mathbf{h} - \mu_0 \mathbf{1}_K)^\top \Sigma_K^{-1}(\mathbf{h} - \mu_0 \mathbf{1}_K) \\ &\quad - \frac{1}{2}(\mathbf{h} - \mu_1 \mathbf{1}_K)^\top \Sigma_K^{-1}(\mathbf{h} - \mu_1 \mathbf{1}_K) \\ &= \Delta\, \mathbf{1}_K^\top \Sigma_K^{-1}(\mathbf{h} - m\mathbf{1}_K).\end{aligned}$$

The covariance satisfies

$$\Sigma_K \mathbf{1}_K = \sigma^2[1 + (K-1)\rho]\mathbf{1}_K,$$

and hence

$$\Sigma_K^{-1}\mathbf{1}_K = \frac{\mathbf{1}_K}{\sigma^2[1 + (K-1)\rho]}.$$

Using the symmetry of Σ_K^{-1} in the preceding expression for the log-likelihood ratio yields

$$\begin{aligned}\Lambda_{H,U}(\mathbf{h} \mid K) &= \frac{\Delta}{\sigma^2[1 + (K-1)\rho]}\mathbf{1}_K^\top(\mathbf{h} - m\mathbf{1}_K) \\ &= \frac{K}{1 + (K-1)\rho}\frac{\Delta}{\sigma^2}(\bar{h} - m) \\ &= K_{\text{eff}}\frac{\mu_1 - \mu_0}{\sigma^2}\left(\bar{h} - \frac{\mu_1 + \mu_0}{2}\right),\end{aligned}$$

where $K_{\text{eff}} = K/[1 + (K-1)\rho]$. Since $\Lambda_{\text{ind}}(\mathbf{h} \mid K) = K\Delta(\bar{h} - m)/\sigma^2$, this establishes

$$\Lambda_{H,U}(\mathbf{h} \mid K) = \frac{\Lambda_{\text{ind}}(\mathbf{h} \mid K)}{1 + (K-1)\rho}.$$

For the second claim, suppose $\Delta \neq 0$, $\rho > 0$, and two different positive lengths belong to the support of both outcome classes. If constants β, γ satisfy $\beta\Lambda_{\text{ind}}(\mathbf{h} \mid K) + \gamma = \Lambda_{H,U}(\mathbf{h} \mid K)$ for every history vector at both lengths, then at each of these lengths,

$$\beta\,\frac{K\Delta}{\sigma^2}(\bar{h} - m) + \gamma = \frac{K\Delta}{\sigma^2[1 + (K-1)\rho]}(\bar{h} - m)$$

for every $\bar{h} \in \mathbb{R}$. Indeed, the conditional Gaussian distributions have support $\mathbb{R}^K$, and every real history mean is attained by some vector in that support. Setting $\bar{h} = m$ gives $\gamma = 0$. Equating the coefficients of $\bar{h}$ and using $K\Delta \neq 0$ then gives

$$\beta = \frac{1}{1 + (K-1)\rho}.$$

The right-hand side takes different values at the two lengths because $\rho > 0$. Thus, no common pair (β, γ) satisfies the required equality, proving the second claim. □

The effective history size K_{eff} describes the precision of the history mean under the assumed model, since

$$\text{Var}(\bar{h} \mid Y = y, K) = \frac{\mathbf{1}_K^\top \Sigma_K \mathbf{1}_K}{K^2} = \sigma^2\left(\rho + \frac{1-\rho}{K}\right) = \frac{\sigma^2}{K_{\text{eff}}}.$$

Thus, K_{eff} is the equivalent number of independent observations with variance σ^2 in terms of the variance of their mean. It equals K when $\rho = 0$ and equals one when $K = 1$. For any $\rho \in (0, 1)$, it increases towards $1/\rho$ as K grows, while the variance of the mean approaches $\sigma^2\rho$. Additional steps therefore provide diminishing gains in precision under this dependence model. Length can also carry evidence about the outcome through differences in its distribution between classes. For the evidence pair $(K, \mathbf{h})$, Bayes' rule and the probability chain rule give

$$\begin{aligned}\ell^\star(K, \mathbf{h}) &:= \log \frac{P(Y=1 \mid K, \mathbf{h})}{P(Y=0 \mid K, \mathbf{h})} \\ &= \ell_0 + \log \frac{P(K \mid Y=1)}{P(K \mid Y=0)} + \Lambda_{H,U}(\mathbf{h} \mid K).\end{aligned}$$

This decomposition requires no independence assumption between length and history values. The length ratio describes the association between trajectory length and outcome, while the conditional history ratio measures the evidence in the observed values at that length. Since $N = K + 1$, either length variable carries the same information. The fitted length and history components of Structured Bayesian Fusion serve these two roles, and their agreement with the population terms depends on the adequacy of their models and estimation.

The affine limitation applies to the form of the generation-history recalibration in equation 17. For positive history lengths, that transformation can be written as $\beta\Lambda^{\text{raw}}_{H,U} + \gamma$, with $\beta = \widehat{\beta}_{H,U}$ and $\gamma = -\widehat{\beta}_{H,U}\widehat{a}_{H,U}$. Even when the raw score uses the population means and variance, unequal class means and positive correlation prevent a common pair of coefficients from recovering the conditional history likelihood ratio across different lengths under the assumed model. The required rescaling varies with length, and a common offset cannot compensate for that variation. This limitation concerns exact recovery of the generation-history likelihood ratio. The usefulness of fitted affine recalibration for outcome ranking and probability calibration remains an empirical question.

B Additional Details of Experimental Setup

B.1 Detailed Description of Baselines

We compare the Bayesian methods from Section 3 with trajectory length, tool success rate, five generation-level signals, three trajectory-level UQ methods, and logistic regression. As in Section 3.1, N denotes the number of scoring steps of a trajectory τ, with action a_t and feedback o_t at step t. Let n_t be the number of generated tokens at step t. For the j-th generated token of step t, $j = 1, \ldots, n_t$, let $p_{tj}(v)$ denote the estimated probability of vocabulary token $v \in V$, and p_{tj} the probability of the generated token. For raw generation-level UQ methods, we consider three aggregation techniques: the average and the maximum over the trajectory, and the value at the last step alone. All learnable baselines are trained on the source training folds only, with the binary outcome Y as the label.

N steps. Shorter trajectories indicate simpler tasks for the agent and therefore receive higher confidence:

$$\mathbf{u}_{N_{\text{steps}}} = -N.$$

Tool success rate. Fraction of passed tool checks among the M observed checks $c_j \in \{0, 1\}$ in the trajectory, with missing tools skipped:

$$\mathbf{u}_{\text{TSR}} = \frac{1}{M}\sum_{j=1}^{M} c_j.$$

Sequence probability (SP). Negative sum of the token-level log-probabilities of the generated sequence:

$$\mathbf{u}_{\text{SP}} = -\sum_{j=1}^{n_t} \log p_{tj}.$$

Perplexity. Negative average token-level log-probability, commonly referred to as perplexity (Fomicheva et al., 2020):

$$\mathbf{u}_{\text{PPL}} = \exp\left(-\frac{1}{n_t}\sum_{j=1}^{n_t} \log p_{tj}\right).$$

Mean token entropy (MTE). Average entropy of the token distributions (Fomicheva et al., 2020). We approximate each distribution by its top-k tokens ($k = 20$), renormalized to sum to one:

$$\mathbf{u}_{\text{MTE}} = -\frac{1}{n_t}\sum_{j=1}^{n_t}\sum_{v\in V} p_{tj}(v)\log p_{tj}(v).$$

Self-certainty. Self-certainty (Kang et al., 2025) measures the KL divergence from the uniform distribution $\mathcal{U}$ over the top-k tokens. We approximate it with raw top-k log-probabilities and reverse it to obtain an uncertainty score:

$$\mathbf{u}_{\mathrm{SC}} = \frac{1}{n_t}\sum_{j=1}^{n_t} \mathrm{KL}(\mathcal{U}|p_{tj}) = \frac{1}{n_t}\sum_{j=1}^{n_t}\left(-\frac{1}{k}\sum_{v\in\mathrm{top}-k}\log p_{tj}(v) - \log k\right).$$

Verbalized confidence (Verb.). We additionally explicitly ask the agent to report its confidence at each step and use its response as the uncertainty estimate (Tian et al., 2023):

$$\mathbf{u}_{\mathrm{Verb.}} = 1 - \widehat{p}_t^{\,\mathrm{verb.}}$$

Situation-Aware Uncertainty Propagation (SAUP). SAUP (Zhao et al., 2025) propagates step uncertainty with larger weight on late steps and on steps that differs from the task context. Let $\mathbf{u}_t$ be the MTE of step t and w_t its situational weight:

$$\mathbf{u}_{\mathrm{SAUP}}(\tau) = \sqrt{\frac{1}{N}\sum_{t=1}^{N}(w_t\mathbf{u}_t)^2}.$$

We consider five choices of the weights w_t, following the variants of the original method: uniform, positional (t/N), based on embedding distances between the task, the step and its observation, their sum, or predicted by a Gaussian HMM over those distances. We use the HMM variant without labels, since the original work relies on manually annotated states that are unavailable for our trajectories.

Uncertainty Propagation (UProp). UProp (Duan et al., 2025) splits the uncertainty of each step into the intrinsic uncertainty of the current decision and the extrinsic uncertainty inherited from earlier decisions. At each step t, we resample $S = 10$ alternative actions $a_t^{(1)}, \ldots, a_t^{(S)}$ from the same model and prompt h_t. Let $|a|$ denote the number of tokens in action a, dist a string distance, and κ a Gaussian kernel. The intrinsic uncertainty (IU) is the mean length-normalized negative log-likelihood of the sampled actions, and the pointwise mutual information (PMI) measures how unusual the action a_t is among sampled alternatives:

$$\mathrm{IU}_t = -\frac{1}{S}\sum_{s=1}^{S}\frac{1}{|a_t^{(s)}|}\log p\big(a_t^{(s)} \mid h_t\big), \qquad \mathrm{PMI}_t = -\log\frac{1}{S}\sum_{s=1}^{S}\kappa\big(\mathrm{dist}(a_t^{(s)}, a_t)\big).$$

Finally, the extrinsic uncertainty (EU) accumulates the PMI of earlier steps, $\mathrm{EU}_t = \sum_{t'<t}\mathrm{PMI}_{t'}$. Given the step uncertainty $H_t = \mathrm{IU}_t + \mathrm{EU}_t$, the trajectory uncertainty is

$$\mathbf{u}_{\mathrm{UProp}}(\tau) = \frac{\sum_{t=1}^{N} H_t}{N + \sum_{t=1}^{N}\mathrm{EU}_t/\mathrm{IU}_t}.$$

Holistic Trajectory Calibration (HTC). HTC (Zhang et al., 2026b) is a supervised calibrator that fits a logistic regression on 48 trajectory-level features of the per-token confidence trace, where token confidence is the probability of the top-1 token and the mean probability of the top-k tokens. The features are grouped into dynamics (how confidence changes across steps), position (confidence statistics at the first and last steps), stability (variability of confidence within and across steps), and structure (step count and tokens per step). Following the original work, we report HTC-Reduced, which fits the regression with an L_1 penalty.

Logistic regression. Additionally, we train an L_2-penalized logistic regression that predicts task success from trajectory-level features. The candidate features are aggregates of each generation-level signal over steps (e.g., first, last, mean, maximum, and variability), trajectory-level tool outcomes, and per-step pass rates. A small subset of features is selected greedily on the training folds.

B.2 Detailed Description of PRR

The Prediction Rejection Ratio (PRR) (Malinin & Gales, 2021; Vashurin et al., 2025) measures how effectively an uncertainty score ranks agent trajectories by the quality of their solutions. It is

computed from the rejection curve, which shows the average quality of the remaining trajectories as an increasing fraction of the most uncertain trajectories is rejected. PRR normalizes the area under this curve for the evaluated method with the areas for random and oracle rankings:

$$\mathrm{PRR} = \frac{\mathrm{AUC}_{\mathrm{method}} - \mathrm{AUC}_{\mathrm{random}}}{\mathrm{AUC}_{\mathrm{oracle}} - \mathrm{AUC}_{\mathrm{random}}}.$$

PRR is upper-bounded by one, which corresponds to the oracle ranking. A value of zero corresponds to random ranking, and negative values indicate performance below random. Following prior work (Vashurin et al., 2025), we compute PRR only over the first 50% of the rejection curve (PRR@0.5), since rejecting more than half of the trajectories is impractical.

PRR is analogous to ROC-AUC, but unlike it, applies to both binary and continuous outcome measures. As the final solution quality, we use the native score of each benchmark:

- **Retrieval:** the judge's verdict on the final answer;
- **Coding:** whether the final solution passes all hidden tests; for DeepSWE, the fraction of passed hidden tests;
- **τ^3-bench:** whether the benchmark reward equals one;
- **WebArena-Verified, OSWorld-Verified, and ALFWorld** (Hattami et al., 2025; Xie et al., 2024; Shridhar et al., 2021): the benchmark evaluator's score, including partial credit on OSWorld-Verified.

B.3 Evaluation Protocol

Cross-validation. For each configuration, we split the episodes into five stratified folds. Folds are stratified by the binary outcome. Each estimator is fitted on four folds and applied to the held-out fold. We pool the held-out predictions of all five folds and compute a single PRR@0.5 over the pooled set, so every episode is scored exactly once by a model that has not seen it.

Random seeds and statistical robustness. We repeat the fold splitting with three seeds, $s \in \{0, 1, 2\}$. For each configuration, we compute PRR@0.5 for every seed and report the mean and standard deviation σ_i over seeds. Unsupervised methods, such as raw generation-level signals, do not depend on the fold split and therefore have zero standard deviation. The small standard deviations observed for some of these methods arise from random tie-breaking between equal scores. When a table column averages n configurations, we report the mean of their PRR@0.5 values, and its standard deviation is obtained by propagating the per-configuration standard deviations, treating configurations as independent with $\sigma_{\mathrm{avg}} = \frac{1}{n}\sqrt{\sum_{i=1}^{n} \sigma_i^2}$.

Trajectory-level aggregation. Aggregation techniques vary by the task, as specified in the captions of Tables 7 to 9. For coding, Bayes Fused uses continuous aggregation, while raw UQ scores are averaged over the trajectory. For retrieval-based and interactive tasks, Bayes Fused uses history + last aggregation, while raw UQ uses only the last-step score. These aggregations are supported by the task formulation itself. Coding tasks require iterative refinement, where each step matters equally, while for other tasks, only the last step provides the final solution and should therefore be treated differently. A detailed analysis of different aggregation methods for Bayes Fused and UQ is presented in Appendix E.

Out-of-distribution evaluation. For the out-of-distribution experiments, all parameters of an estimator, including likelihoods, thresholds, normalization statistics, and regression weights, are fitted on the four training folds of the source configuration. The fitted estimator is then applied unchanged to the corresponding held-out fold of the target configuration, and predictions are pooled over the five folds as in the in-distribution setting. We use the same three random seeds and, for reference, also report the in-distribution PRR@0.5 of the target configuration.

Data filtering. We evaluate only episodes that the agent completed itself, excluding those stopped by the step budget, the context window, or an environment failure. Such externally terminated episodes are failures by construction, so estimating their uncertainty is uninformative.

B.4 Retrieval-Based Environments

This subsection describes the HotpotQA, MuSiQue, CRAG, and BrowseComp-Plus environments in detail.

Table 1: **Retrieval task statistics.** For each model and harness configuration, *Episodes* reports the total number of scored episodes and the completed episodes retained for evaluation. *Steps* reports the maximum and median number of scored steps per episode. *Success rate* is the fraction of successful episodes.

Benchmark	Model	Harness	Episodes		Steps		Success rate
			all	completed	max	median	
HotpotQA	Qwen2.5-32B-Instruct	smolagents CodeAgent	700	700	13	2	0.836
		LangGraph ReAct	700	700	7	1	0.737
	GPT-OSS-20B	smolagents CodeAgent	700	646	14	1	0.804
		LangGraph ReAct	700	700	12	2	0.824
	Qwen3.6-35B-A3B	smolagents CodeAgent	700	669	36	2	0.891
		LangGraph ReAct	700	700	20	2	0.876
MuSiQue	Qwen2.5-32B-Instruct	smolagents CodeAgent	300	299	16	3	0.550
		LangGraph ReAct	300	300	12	2	0.450
	GPT-OSS-20B	smolagents CodeAgent	300	235	17	2	0.477
		LangGraph ReAct	300	300	12	3	0.507
	Qwen3.6-35B-A3B	smolagents CodeAgent	300	213	36	6	0.680
		LangGraph ReAct	300	300	24	5	0.600
CRAG	Qwen2.5-32B-Instruct	smolagents CodeAgent	300	296	8004	4	0.323
		LangGraph ReAct	300	297	13	3	0.263
	GPT-OSS-20B	smolagents CodeAgent	300	249	32	2	0.357
		LangGraph ReAct	300	198	13	6	0.313
	Qwen3.6-35B-A3B	smolagents CodeAgent	300	264	602	6	0.543
		LangGraph ReAct	300	237	13	8	0.460
BrowseComp-Plus	Qwen3.6-35B-A3B	LangGraph ReAct	830	696	55	23	0.391
		smolagents ToolCalling	830	734	51	27.5	0.347

Table 2: **Retrieval task critics.** Definitions of the critics used to assess retrieval actions and their outcomes.

Benchmark	Critic	Description
HotpotQA, MuSiQue	`retrieved_any`	the search returned at least one document
	`retrieved_new`	the search returned a document not retrieved earlier
CRAG	`tool_ok`	the tool call executed without an error
	`tool_returned_content`	the tool returned non-empty content
	`query_is_new`	the query differs from all earlier queries
BrowseComp-Plus	`action_format_valid`	all actions in the episode are well-formed
	`ui_execution_success`	all tool calls in the episode executed successfully
	`state_change`	at least one tool call returned new text
	`no_stagnation`	the agent does not end in a repeated cycle of actions

Models and harnesses. We generate trajectories by Qwen2.5-32B-Instruct (Yang et al., 2024a), GPT-OSS-20B (OpenAI, 2025), and Qwen3.6-35B-A3B (Qwen Team, 2026a), each served locally through vLLM (Kwon et al., 2023). On HotpotQA, MuSiQue, and CRAG, each model runs with two harnesses: smolagents CodeAgent (Roucher et al., 2025), which writes Python code that calls the search tool, and LangGraph ReAct (Yao et al., 2023; LangChain, Inc., 2026), which includes free-text thoughts followed by actions. BrowseComp-Plus is run with Qwen3.6-35B-A3B (Qwen Team, 2026a) only, under LangGraph ReAct (Yao et al., 2023; LangChain, Inc., 2026) and smolagents ToolCallingAgent (Roucher et al., 2025), since the other models achieve near-zero success on this benchmark. For the reasoning models, verbalized confidence is elicited with disabled thinking. We evaluate only episodes that the agent completed itself. Table 1 presents all 20 configurations with their numbers of episodes, step statistics, and success rates.

Critics. Critics differ across benchmarks and are listed in Table 2. A critic is a boolean check applied at each step by the harness itself, without additional model calls. Critics check the mechanics of a step, such as whether a tool was executed and returned content, and never inspect the answer itself. Therefore, a step that retrieves irrelevant documents can still pass, so the informativeness of critics varies across benchmarks.

Hardware configuration. We use four NVIDIA A100-SXM4-80GB GPUs for HotpotQA, MuSiQue, and CRAG, and two NVIDIA RTX PRO 6000 Blackwell GPUs for BrowseComp-Plus.

B.5 CODING ENVIRONMENTS

Table 3: **Coding task statistics.** For each model and harness configuration, *Episodes* reports the total and completed episode counts. *Steps* reports the maximum and median number of scored steps per episode. *Success rate* is the fraction of completed episodes passing all hidden tests, except for DeepSWE, where it is the mean fraction of hidden tests passed. Episodes with no candidate solution because every generation ended within the reasoning block are excluded.

Benchmark	Model	Harness	Episodes		Steps		Success rate
			all	completed	max	median	
LiveCodeBench-Hard	GPT-OSS-20B	smolagents CodeAgent	350	309	13	7	0.49
		ReAct	350	176	18	10	0.56
	GPT-OSS-120B	smolagents CodeAgent	350	304	23	13	0.65
		ReAct	350	237	20	13	0.61
	Qwen3.6-35B-A3B	smolagents CodeAgent	350	285	23	16	0.71
		ReAct	350	288	23	17	0.73
	Qwen3.8-27B	smolagents CodeAgent	348	314	23	15	0.77
		ReAct	350	322	23	15	0.74
BigCodeBench-Hard	GPT-OSS-20B	smolagents CodeAgent	117	102	11	7	0.39
		ReAct	118	51	20	10	0.45
	GPT-OSS-120B	smolagents CodeAgent	117	71	20	13	0.58
		ReAct	119	61	16	12	0.52
	Qwen3.6-35B-A3B	smolagents CodeAgent	119	78	20	12	0.47
		ReAct	119	78	23	17	0.47
	Qwen3.8-27B	smolagents CodeAgent	119	73	23	12	0.58
		ReAct	119	79	22	15	0.54
SWE-bench	GPT-OSS-20B	smolagents CodeAgent	234	215	20	7	0.59
		ReAct	232	178	24	8	0.61
	GPT-OSS-120B	smolagents CodeAgent	232	199	23	7	0.67
		ReAct	233	197	23	8	0.67
	Qwen3.6-35B-A3B	smolagents CodeAgent	234	217	24	7	0.66
		ReAct	233	211	24	7	0.67
	Qwen3.8-27B	smolagents CodeAgent	234	213	24	7	0.74
		ReAct	234	222	24	7	0.72
CodeContests	GPT-OSS-20B	smolagents CodeAgent	60	54	12	7	0.50
		ReAct	60	29	14	10	0.48
	GPT-OSS-120B	smolagents CodeAgent	60	53	22	12	0.72
		ReAct	60	36	21	12	0.78
	Qwen3.6-35B-A3B	smolagents CodeAgent	60	52	22	13.5	0.88
		ReAct	60	52	22	17	0.88
	Qwen3.8-27B	smolagents CodeAgent	60	58	23	15	0.84
		ReAct	60	55	23	15	0.85
DeepSWE	GPT-OSS-20B	mini-swe-agent	113	99	155	59	0.48
	GPT-OSS-120B	mini-swe-agent	113	87	158	57	0.58
	Qwen3.6-35B-A3B	mini-swe-agent	113	65	192	125	0.78

Table 4: **Coding task critics.** Definitions of the critics used to assess candidate solutions and coding trajectories.

Benchmark	Critic	Description
LiveCodeBench-Hard, BigCodeBench-Hard, SWE-bench, CodeContests	`critic_L0`	the candidate parses without syntax errors
	`critic_L1`	static linting finds no undefined names or redefinitions
	`critic_L2`	the candidate passes the public tests
	`critic_L3`	a reviewer model accepts the candidate
DeepSWE	`ran_tests`	the agent ran the tests at least once
	`last_test_passed`	the last test run passed
	`test_flipped`	a test run passed after an earlier failure
	`no_format_errors`	the agent produced no malformed responses

This subsection describes the LiveCodeBench-Hard, BigCodeBench-Hard, SWE-bench, CodeContests, and DeepSWE environments in detail.

Models and harnesses. On LiveCodeBench-Hard, BigCodeBench-Hard, SWE-bench, and CodeContests, we generate trajectories with GPT-OSS-20B, GPT-OSS-120B (OpenAI, 2025), Qwen3.6-35B-A3B (Qwen Team, 2026a), and Qwen3.8-27B (Qwen Team, 2026b) in thinking mode, each served locally through vLLM (Kwon et al., 2023). Each model runs with two harnesses: smolagents CodeAgent (Roucher et al., 2025) and ReAct (Yao et al., 2023). On DeepSWE, GPT-OSS-20B, GPT-OSS-120B (OpenAI, 2025), and Qwen3.6-35B-A3B (Qwen Team, 2026a) run with mini-swe-

agent over 113 tasks with a 200-command budget. An episode is successful if the final solution passes all hidden tests; on DeepSWE, where no model resolves a task outright, we use the fraction of passed hidden tests. We evaluate only episodes that the agent completed itself. Table 3 presents all 35 configurations with their numbers of episodes, step statistics, and success rates.

Critics. Critics differ across benchmarks and are listed in Table 4. On LiveCodeBench-Hard, BigCodeBench-Hard, SWE-bench, and CodeContests, critics are tools that the agent may invoke on its current candidate solution; SWE-bench provides no public tests, so `critic_L2` is unavailable there. On DeepSWE, critics summarize the episode's use of tests and version control.

Hardware configuration. We use NVIDIA H100-80GB GPUs for LiveCodeBench-Hard, BigCodeBench-Hard, SWE-bench, and CodeContests, and NVIDIA A100-80GB GPUs for DeepSWE.

B.6 Interactive Environments

Table 5: **Interactive task statistics.** For each model and harness configuration, *Episodes* reports the total and completed episode counts. *Steps* reports the maximum and median number of scored steps per episode. *Success rate* is the fraction of successful episodes.

Benchmark	Model	Harness	Episodes		Steps		Success rate
			all	completed	max	median	
τ^3-retail	Qwen2.5-32B-Instruct	LLM agent	456	445	30	14	0.401
		Planning agent	456	447	30	14	0.408
	GPT-OSS-20B	LLM agent	456	455	30	12	0.417
		Planning agent	454	451	30	13	0.467
	Qwen3.6-35B-A3B	LLM agent	456	451	30	13	0.739
		Planning agent	456	438	30	13	0.669
τ^3-airline	Qwen2.5-32B-Instruct	LLM agent	200	194	30	10	0.305
		Planning agent	197	191	30	13	0.249
	GPT-OSS-20B	LLM agent	200	200	27	8	0.525
		Planning agent	200	199	30	9	0.525
	Qwen3.6-35B-A3B	LLM agent	200	165	30	13	0.640
		Planning agent	200	175	30	13	0.650
τ^3-telecom	Qwen2.5-32B-Instruct	LLM agent	228	154	22	13	0.246
		Planning agent	228	146	25	16	0.206
	GPT-OSS-20B	LLM agent	228	127	26	18	0.329
		Planning agent	228	111	26	19	0.215
	Qwen3.6-35B-A3B	LLM agent	456	225	30	18	0.314
		Planning agent	456	214	30	20	0.241
WebArena-Verified	GPT-OSS-20B	GenericAgent	812	315	96	12	0.441
	Qwen3.6-35B-A3B	GenericAgent	812	662	104	13	0.462
OSWorld-Verified	Qwen3.6-35B-A3B	Qwen OSWorld agent	361	275	100	16	0.590
ALFWorld	GPT-OSS-20B	ReAct	140	97	48	13	0.340
		smolagents CodeAgent	140	139	75	17	0.353
	Qwen3.6-35B-A3B	ReAct	140	86	45	12	0.826
		smolagents CodeAgent	140	122	48	9	0.893

Table 6: **Interactive task critics.** Definitions of the critics used to assess agent actions and their effects on the environment.

Benchmark	Critic	Description
τ^3-bench	`is_tool_call`	the step issues a tool call
	`tool_ok`	the tool call executed without an error
	`tool_returned_content`	the tool returned non-empty content
WebArena-Verified, OSWorld-Verified	`action_format_valid`	all actions in the episode are well-formed
	`ui_execution_success`	all actions executed without an error
	`state_change`	at least one action visibly changed the screen
	`no_stagnation`	the agent does not end in a repeated cycle of ineffective actions
	`llm_judge_pass`	the last LLM verifier call accepted the state
	`agent_reported_success`	the agent reports success (WebArena-Verified only)
ALFWorld	`format_valid`	the response follows the required format
	`action_valid`	the proposed action is admissible
	`no_repeated_fallback`	the harness did not replace the action for exceeding the repeat limit
	`state_changed`	the observation or the set of admissible actions changed

This subsection describes the τ^3-bench, WebArena-Verified, OSWorld-Verified, and ALFWorld environments in detail.

Models and harnesses. On τ^3-bench retail, airline, and telecom, we generate trajectories with Qwen2.5-32B-Instruct (Yang et al., 2024a), GPT-OSS-20B (OpenAI, 2025), and Qwen3.6-35B-A3B (Qwen Team, 2026a), each running with the two τ-bench agents: a plain LLM agent and a planning agent. On WebArena-Verified, GPT-OSS-20B (OpenAI, 2025) and Qwen3.6-35B-A3B (Qwen Team, 2026a) run with WebArenaAgent from the benchmark itself; Qwen3.6-35B-A3B observes the accessibility tree together with a screenshot, and GPT-OSS observes the accessibility tree only. OSWorld-Verified is run with Qwen3.6-35B-A3B (Qwen Team, 2026a) using the Qwen OSWorld agent. On ALFWorld, GPT-OSS-20B (OpenAI, 2025) and Qwen3.6-35B-A3B (Qwen Team, 2026a) run 140 tasks of the `valid_seen` split with a ReAct (Yao et al., 2023) scaffold and smolagents CodeAgent under a 50-step budget. On τ^3-bench, an episode is successful if the benchmark reward equals one; on the other environments, success is determined by the benchmark evaluator. We evaluate only episodes that the agent completed itself. Table 5 presents all 25 configurations with their numbers of episodes, step statistics, and success rates.

Critics. Critics differ across benchmarks and are listed in Table 6. On τ^3-bench and ALFWorld, critics are recorded at each step. On WebArena-Verified and OSWorld-Verified, step-level checks are aggregated into episode-level critics, and `llm_judge_pass` is the verdict of an LLM verifier that the agent may call as a tool.

Hardware configuration. We use four NVIDIA A100-SXM4-80GB GPUs for τ^3-bench, NVIDIA A100-80GB GPUs for ALFWorld, and two NVIDIA RTX PRO 6000 Blackwell GPUs for WebArena-Verified and OSWorld-Verified.

C Detailed Results

C.1 Average Results

Table 7: **In-domain retrieval results.** PRR@0.5↑, averaged over available model and harness configurations on HotpotQA, MuSiQue, CRAG, and BrowseComp-Plus. Raw UQ uses the last-step score, while Bayes Fused uses History + Last aggregation. Avg is the mean across available dataset columns. The best result in each column is **bold**, and the second-best is underlined.

Group	Method	HotpotQA	MuSiQue	CRAG	BrowseComp	Average
Stat.	N steps	.218 ± .010	.370 ± .011	.185 ± .014	.810 ± .000	.396 ± .005
	Tool success rate	.293 ± .010	.340 ± .007	-.031 ± .021	-.005 ± .000	.149 ± .006
UQ	SP	.129 ± .010	.126 ± .005	.267 ± .010	.515 ± .000	.259 ± .004
	Perplexity	-.045 ± .011	-.004 ± .002	.044 ± .005	.900 ± .000	.224 ± .003
	MTE	-.057 ± .010	.002 ± .005	.068 ± .005	.915 ± .000	.232 ± .003
	Self-certainty	.230 ± .011	.173 ± .007	.219 ± .007	.900 ± .000	.380 ± .004
	Verb.	.408 ± .002	.320 ± .000	**.474** ± .000	.715 ± .000	.479 ± .000
	SAUP	.174 ± .002	.190 ± .009	.225 ± .013	.400 ± .000	.247 ± .004
	UProp	.357 ± .000	.208 ± .000	.070 ± .000	.815 ± .000	.363 ± .000
Supervised	HTC	.371 ± .016	**.469** ± .020	.224 ± .025	.945 ± .005	.502 ± .009
	Logistic regression	.307 ± .018	.350 ± .010	.358 ± .027	**.975** ± .005	.498 ± .008
Bayes (ours)	Tool-only	.307 ± .011	.353 ± .009	-.020 ± .025	.445 ± .030	.271 ± .010
	Fused (SP)	.329 ± .013	.376 ± .009	.129 ± .019	.585 ± .000	.355 ± .006
	Fused (Perplexity)	.323 ± .010	.374 ± .012	.177 ± .021	.875 ± .000	.437 ± .006
	Fused (MTE)	.325 ± .010	.376 ± .007	.198 ± .022	.895 ± .000	.449 ± .006
	Fused (Self-certainty)	.319 ± .009	.384 ± .008	.213 ± .021	.875 ± .000	.448 ± .006
	Fused (Verb.)	**.447** ± .005	.443 ± .006	.391 ± .013	.940 ± .005	**.555** ± .004
	SBF	.334 ± .011	.403 ± .009	.391 ± .015	.930 ± .000	.515 ± .005

Table 8: **In-domain coding results.** PRR@0.5↑, averaged over available model and harness configurations on LiveCodeBench-Hard, BigCodeBench-Hard, SWE-bench, CodeContests, and DeepSWE. Raw UQ uses the trajectory mean. Bayes Fused uses Continuous aggregation after a rank-Gaussian transformation fitted to training-fold signal values without using labels or target data. Class-conditional Gaussian distributions are fitted to the transformed values. Avg is the mean across available dataset columns. The best result in each column is **bold**, and the second-best is underlined.

Group	Method	LCB	BCB	SWE	CC	DeepSWE	Average
Stat.	N steps	.082 ± .000	.073 ± .000	.133 ± .000	.262 ± .000	.053 ± .000	.121 ± .000
	Tool success rate	.171 ± .000	.128 ± .000	.088 ± .000	.422 ± .000	-.163 ± .000	.129 ± .000
UQ	SP	.278 ± .000	.254 ± .000	.364 ± .000	.553 ± .000	.017 ± .000	.293 ± .000
	Perplexity	.047 ± .000	.192 ± .000	.218 ± .000	.204 ± .000	.007 ± .000	.134 ± .000
	MTE	.058 ± .000	.187 ± .000	.221 ± .000	.207 ± .000	.023 ± .000	.139 ± .000
	Self-certainty	.173 ± .000	.174 ± .000	.268 ± .000	.387 ± .000	-.030 ± .000	.195 ± .000
	Verb.	.111 ± .000	.157 ± .000	.318 ± .000	.406 ± .000	-.007 ± .000	.197 ± .000
	SAUP	.070 ± .000	.137 ± .000	.188 ± .000	.226 ± .000	.030 ± .000	.130 ± .000
	UProp	.066 ± .000	**.267** ± .000	.169 ± .000	.268 ± .000	-.010 ± .000	.152 ± .000
Supervised	HTC	.184 ± .018	.114 ± .049	.355 ± .015	.094 ± .106	.120 ± .038	.173 ± .025
	Logistic regression	.280 ± .004	.187 ± .033	.354 ± .005	.491 ± .024	.073 ± .032	.277 ± .010
Bayes (ours)	Tool-only	.191 ± .008	.135 ± .034	.169 ± .010	.383 ± .020	.120 ± .038	.199 ± .011
	Fused (SP)	.297 ± .002	.262 ± .026	.362 ± .003	**.561** ± .009	.207 ± .021	**.338** ± .007
	Fused (Perplexity)	.156 ± .011	.205 ± .012	.204 ± .016	.297 ± .018	**.217** ± .026	.216 ± .008
	Fused (MTE)	.160 ± .011	.208 ± .009	.207 ± .015	.323 ± .018	.167 ± .030	.213 ± .008
	Fused (Self-certainty)	.226 ± .004	.204 ± .008	.248 ± .010	.435 ± .015	.137 ± .031	.250 ± .007
	Fused (Verb.)	.257 ± .004	.146 ± .025	.336 ± .006	.328 ± .017	.090 ± .039	.231 ± .010
	SBF	**.308** ± .010	.132 ± .022	**.371** ± .009	.546 ± .029	.173 ± .036	.306 ± .011

Table 9: **In-domain interactive results.** PRR@0.5↑, averaged over available model and harness configurations on τ^3-bench, WebArena-Verified, OSWorld-Verified, and ALFWorld. Raw UQ uses the last-step score, while Bayes Fused uses History + Last aggregation. Avg is the mean across available dataset columns. The best result in each column is **bold**, and the second-best is underlined.

Group	Method	τ^3-retail	τ^3-airline	τ^3-telecom	WebArena	OSWorld	ALFWorld	Average
Stat.	N steps	.131 ± .015	.357 ± .030	.501 ± .032	.290 ± .000	.350 ± .000	.388 ± .000	.336 ± .008
	Tool success rate	.142 ± .015	.145 ± .048	.206 ± .022	.400 ± .000	.000 ± .000	.538 ± .000	.238 ± .009
UQ	SP	.097 ± .023	.077 ± .046	-.058 ± .017	.435 ± .000	.220 ± .000	.323 ± .000	.182 ± .009
	Perplexity	-.026 ± .010	-.126 ± .040	-.060 ± .020	.470 ± .000	.040 ± .000	.250 ± .000	.091 ± .008
	MTE	-.021 ± .009	-.111 ± .041	-.080 ± .018	.480 ± .000	.060 ± .000	.310 ± .000	.106 ± .008
	Self-certainty	.077 ± .015	.042 ± .027	-.026 ± .011	.545 ± .000	.280 ± .000	.340 ± .000	.210 ± .005
	Verb.	.170 ± .000	-.225 ± .000	.184 ± .000	.285 ± .000	.280 ± .000	.188 ± .000	.147 ± .000
	SAUP	.136 ± .006	-.332 ± .010	.419 ± .018	.170 ± .005	.100 ± .010	.313 ± .043	.134 ± .008
	UProp	.231 ± .000	.088 ± .000	.254 ± .000	.420 ± .000	.050 ± .000	.405 ± .000	.241 ± .000
Super-vised	HTC	.278 ± .033	.364 ± .019	.574 ± .029	.555 ± .025	.360 ± .020	.528 ± .029	.443 ± .011
	Logistic regression	.222 ± .017	.305 ± .030	**.646** ± .022	**.575** ± .005	**.450** ± .010	.775 ± .033	.496 ± .009
Bayes (ours)	Tool-only	.154 ± .013	.186 ± .041	.434 ± .012	.345 ± .021	.210 ± .010	.648 ± .017	.330 ± .009
	Fused (SP)	**.286** ± .010	.230 ± .022	.518 ± .011	.505 ± .005	.360 ± .010	.528 ± .015	.404 ± .005
	Fused (Perplexity)	.215 ± .013	.213 ± .025	.475 ± .013	.545 ± .000	.170 ± .030	.523 ± .018	.357 ± .008
	Fused (MTE)	.206 ± .018	.303 ± .032	.480 ± .017	.540 ± .000	.160 ± .020	.530 ± .017	.370 ± .008
	Fused (Self-certainty)	.213 ± .013	.306 ± .024	.530 ± .016	.570 ± .000	.300 ± .020	.540 ± .023	.410 ± .007
	Fused (Verb.)	.256 ± .012	.187 ± .041	.536 ± .011	.505 ± .005	.270 ± .020	.540 ± .003	.382 ± .008
	SBF	.278 ± .019	**.479** ± .013	.626 ± .018	.545 ± .007	.400 ± .010	**.825** ± .010	**.526** ± .006

D GENERALIZATION ACROSS MODELS, AGENTS, ENVIRONMENTS, AND DOMAINS

D.1 AVERAGE CROSS-ENVIRONMENT AND CROSS-DOMAIN RESULTS

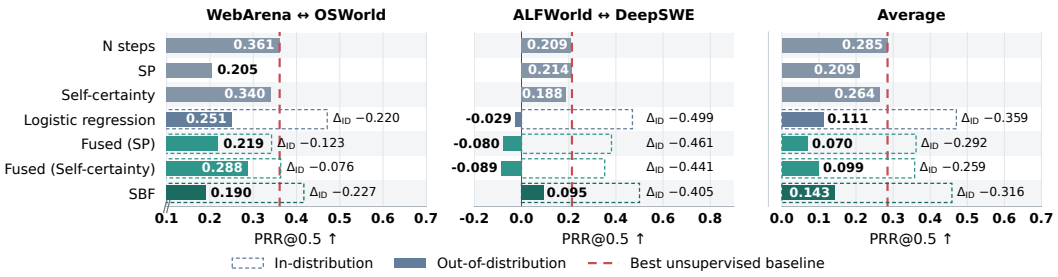

Figure 6: **Cross-environment and cross-domain transfer.** Filled bars show out-of-distribution PRR@0.5↑, while dashed bars show the corresponding in-distribution scores. Bars average performance between WebArena and OSWorld (environment shift) or between ALFWorld and DeepSWE (domain shift). Δ denotes out-of-distribution performance minus in-distribution performance. Detailed per-transfer results are provided in Tables 14 and 15.

D.2 MODEL TRANSFER

Table 10: **Model transfer.** PRR@0.5↑ with the dataset and harness held fixed. Column headers indicate the source → target model. GPT and Qwen denote GPT-OSS-20B and Qwen3.6-35B-A3B, respectively. CodeAgent is the smolagents harness. Δ_{ID} is Avg minus the mean in-domain PRR for the same targets. In-domain scores are reused for N steps, tool success rate, and raw UQ. The best result in each column is **bold**, and the second-best is underlined.

Group	Method	WebArena		HotpotQA				MuSiQue				Avg	Δ_{ID}
		Qwen → GPT	GPT → Qwen	GPT → Qwen		Qwen → GPT		GPT → Qwen		Qwen → GPT			
		WebArenaAgent	WebArenaAgent	CodeAgent	ReAct	CodeAgent	ReAct	CodeAgent	ReAct	CodeAgent	ReAct		
Stat.	N steps	.207 ± .000	.369 ± .000	.380 ± .000	.465 ± .020	.088 ± .000	.341 ± .020	.377 ± .010	**.662** ± .010	.192 ± .050	.656 ± .000	.374 ± .006	–
	Tool success rate	.494 ± .000	.306 ± .000	.369 ± .030	.515 ± .020	.154 ± .010	.503 ± .010	.336 ± .010	.493 ± .000	.225 ± .030	.545 ± .010	.394 ± .005	–
UQ	SP	.680 ± .000	.185 ± .000	.387 ± .000	.072 ± .000	-.060 ± .000	.106 ± .000	.257 ± .000	.118 ± .000	.030 ± .000	.010 ± .000	.179 ± .000	–
	Perplexity	.584 ± .000	.364 ± .000	-.220 ± .000	-.116 ± .000	-.067 ± .000	-.211 ± .000	-.021 ± .000	-.160 ± .000	-.051 ± .000	-.213 ± .000	-.011 ± .000	–
	MTE	.590 ± .000	.373 ± .000	-.265 ± .000	-.135 ± .000	-.068 ± .000	-.213 ± .000	-.099 ± .000	-.083 ± .000	-.018 ± .000	-.227 ± .000	-.014 ± .000	–
	Self-certainty	.692 ± .000	.397 ± .000	.418 ± .000	.331 ± .000	.095 ± .000	.275 ± .000	.188 ± .000	.275 ± .000	-.010 ± .000	.222 ± .000	.288 ± .000	–
	Verb.	.217 ± .000	.346 ± .000	.396 ± .000	.577 ± .000	.205 ± .000	.374 ± .000	.245 ± .000	.394 ± .000	.074 ± .000	.342 ± .000	.317 ± .000	–
	SAUP	.323 ± .007	.156 ± .005	.341 ± .000	.104 ± .000	.100 ± .000	.224 ± .000	.159 ± .000	.106 ± .000	.040 ± .000	.303 ± .000	.186 ± .001	+.019
	UProp	.535 ± .000	.315 ± .000	**.485** ± .000	.586 ± .000	.249 ± .000	.344 ± .000	.340 ± .000	.250 ± .000	.199 ± .000	.368 ± .000	.367 ± .000	–
Super-vised	HTC	.460 ± .071	.176 ± .024	.370 ± .000	-.067 ± .000	.128 ± .000	.327 ± .000	.384 ± .000	.582 ± .000	.270 ± .000	.548 ± .000	.318 ± .007	-.128
	Logistic regression	.613 ± .027	.278 ± .034	.231 ± .000	.135 ± .000	-.034 ± .000	-.058 ± .000	.137 ± .000	.092 ± .000	.030 ± .000	-.040 ± .000	.138 ± .004	-.291
Bayes (ours)	Tool-only	.464 ± .049	.064 ± .022	.401 ± .000	.553 ± .000	.177 ± .000	.524 ± .000	.375 ± .000	.555 ± .000	.246 ± .000	.610 ± .000	.397 ± .005	+.007
	Fused (SP)	.642 ± .006	.269 ± .003	.353 ± .000	.543 ± .000	.217 ± .000	.316 ± .000	.380 ± .000	.576 ± .000	.284 ± .000	.399 ± .000	.398 ± .001	-.034
	Fused (Perplexity)	.682 ± .007	**.428** ± .001	.324 ± .000	.530 ± .000	.102 ± .000	.413 ± .000	.336 ± .000	.570 ± .000	.186 ± .000	.468 ± .000	.404 ± .001	-.024
	Fused (MTE)	.661 ± .007	.411 ± .003	.308 ± .000	.534 ± .000	.120 ± .000	.487 ± .000	.284 ± .000	.501 ± .000	.257 ± .000	.610 ± .000	.417 ± .001	-.022
	Fused (Self-certainty)	**.708** ± .002	.414 ± .002	.340 ± .000	.526 ± .000	.131 ± .000	.424 ± .000	.350 ± .000	.509 ± .000	.250 ± .000	.576 ± .000	.423 ± .000	-.028
	Fused (Verb.)	.589 ± .010	.368 ± .005	**.485** ± .000	**.639** ± .000	**.308** ± .000	**.604** ± .000	.361 ± .000	.563 ± .000	**.474** ± .000	**.674** ± .000	**.507** ± .001	+.048
	SBF	.573 ± .015	.375 ± .013	.408 ± .000	.586 ± .000	.146 ± .000	.429 ± .000	**.399** ± .000	.647 ± .000	.250 ± .000	.652 ± .000	.446 ± .002	-.015

D.3 AGENT TRANSFER

D.4 JOINT MODEL AND AGENT TRANSFER

D.5 ENVIRONMENT TRANSFER

D.6 DOMAIN TRANSFER

Table 11: **Harness transfer.** PRR@0.5↑ with the dataset and model held fixed. Column headers indicate the source → target harness. Code and Tool denote CodeAgent and ToolAgent from smolagents, respectively. Δ_{ID} is Avg minus the mean in-domain PRR for the same targets. In-domain scores are reused for N steps, tool success rate, and raw UQ. The best result in each column is **bold**, and the second-best is underlined.

Group	Method	BrowseComp-Plus		HotpotQA				MuSiQue				Avg	Δ_{ID}
		Qwen3.6-35B-A3B		GPT-OSS-20B		Qwen3.6-35B-A3B		GPT-OSS-20B		Qwen3.6-35B-A3B			
		Tool → ReAct	ReAct → Tool	Code → ReAct	ReAct → Code	Code → ReAct	ReAct → Code	Code → ReAct	ReAct → Code	Code → ReAct	ReAct → Code		
Stat.	N steps	.745 ± .000	.873 ± .000	.341 ± .020	.088 ± .000	.465 ± .020	.380 ± .000	.656 ± .000	.192 ± .050	**.662** ± .010	**.377** ± .010	.478 ± .006	–
Stat.	Tool success rate	.006 ± .000	-.015 ± .000	.503 ± .010	.154 ± .010	.515 ± .020	.369 ± .030	.545 ± .010	.225 ± .030	.493 ± .000	.336 ± .010	.313 ± .005	–
UQ	SP	.553 ± .000	.481 ± .000	.106 ± .000	-.060 ± .000	.072 ± .000	.387 ± .000	.010 ± .000	.030 ± .000	.118 ± .000	.257 ± .000	.195 ± .000	–
UQ	Perplexity	.920 ± .000	.877 ± .000	-.211 ± .000	-.067 ± .000	-.116 ± .000	-.220 ± .000	-.213 ± .000	-.051 ± .000	-.160 ± .000	-.021 ± .000	.074 ± .000	–
UQ	MTE	.943 ± .000	.886 ± .000	-.213 ± .000	-.068 ± .000	-.135 ± .000	-.265 ± .000	-.227 ± .000	-.018 ± .000	-.083 ± .000	-.099 ± .000	.072 ± .000	–
UQ	Self-certainty	.924 ± .000	.883 ± .000	.275 ± .000	.095 ± .000	.331 ± .000	.418 ± .000	.222 ± .000	-.010 ± .000	.275 ± .000	.188 ± .000	.360 ± .000	–
UQ	Verb.	.739 ± .000	.687 ± .000	.374 ± .000	.205 ± .000	.577 ± .000	.396 ± .000	.342 ± .000	.074 ± .000	.394 ± .000	.245 ± .000	.403 ± .000	–
UQ	SAUP	.359 ± .003	.437 ± .004	.326 ± .000	.133 ± .000	.120 ± .000	.267 ± .000	.242 ± .000	.262 ± .000	.165 ± .000	.069 ± .000	.238 ± .001	+.025
UQ	UProp	.813 ± .000	.817 ± .000	.344 ± .000	.249 ± .000	.586 ± .000	**.485** ± .000	.368 ± .000	.199 ± .000	.250 ± .000	.109 ± .000	.422 ± .000	–
Super-vised	HTC	.957 ± .004	.910 ± .005	.251 ± .000	-.015 ± .000	.506 ± .000	.412 ± .000	.539 ± .000	.056 ± .000	.139 ± .000	.051 ± .000	.381 ± .001	-.144
Super-vised	Logistic regression	**.988** ± .003	**.951** ± .005	.216 ± .000	.117 ± .000	.088 ± .000	.477 ± .000	.555 ± .000	.262 ± .000	-.115 ± .000	.134 ± .000	.367 ± .001	-.142
Bayes (ours)	Tool-only	.615 ± .038	-.153 ± .028	.536 ± .000	.178 ± .000	.554 ± .000	.358 ± .000	.640 ± .000	.248 ± .000	.564 ± .000	.132 ± .000	.367 ± .005	-.043
Bayes (ours)	Fused (SP)	.574 ± .005	.572 ± .005	.529 ± .000	.217 ± .000	.545 ± .000	.475 ± .000	.658 ± .000	.276 ± .000	.566 ± .000	.127 ± .000	.454 ± .001	+.007
Bayes (ours)	Fused (Perplexity)	.901 ± .001	.849 ± .002	.526 ± .000	.179 ± .000	.533 ± .000	.451 ± .000	.657 ± .000	.256 ± .000	.552 ± .000	.160 ± .000	.506 ± .000	+.011
Bayes (ours)	Fused (MTE)	.926 ± .001	.863 ± .002	.532 ± .000	.194 ± .000	.529 ± .000	.464 ± .000	.676 ± .000	.288 ± .000	.551 ± .000	.164 ± .000	.519 ± .000	+.008
Bayes (ours)	Fused (Self-certainty)	.890 ± .002	.863 ± .001	.522 ± .000	.236 ± .000	.466 ± .000	**.485** ± .000	.675 ± .000	.285 ± .000	.512 ± .000	.125 ± .000	.506 ± .000	-.007
Bayes (ours)	Fused (Verb.)	.938 ± .002	.932 ± .011	**.588** ± .000	**.321** ± .000	**.639** ± .000	.474 ± .000	**.725** ± .000	**.442** ± .000	.596 ± .000	.144 ± .000	**.580** ± .001	+.034
Bayes (ours)	SBF	.951 ± .001	.894 ± .001	.401 ± .000	.129 ± .000	.603 ± .000	.475 ± .000	.663 ± .000	.280 ± .000	-.117 ± .000	.097 ± .000	.437 ± .000	-.100

Table 12: **Joint model and harness transfer.** PRR@0.5↑ with the dataset held fixed. Column headers specify the source → target model/harness pair. GPT and Qwen denote GPT-OSS-20B and Qwen3.6-35B-A3B, respectively, and Code denotes smolagents CodeAgent. Δ_{ID} is Avg minus the mean in-domain PRR for the same targets. In-domain scores are reused for N steps, tool success rate, and raw UQ. The best result in each column is **bold**, and the second-best is underlined.

Group	Method	HotpotQA				MuSiQue				Avg	Δ_{ID}
		GPT / Code → Qwen / ReAct	Qwen / ReAct → GPT / Code	GPT / ReAct → Qwen / Code	Qwen / Code → GPT / ReAct	GPT / Code → Qwen / ReAct	Qwen / ReAct → GPT / Code	GPT / ReAct → Qwen / Code	Qwen / Code → GPT / ReAct		
Stat.	N steps	.465 ± .020	.088 ± .000	.380 ± .000	.341 ± .020	**.662** ± .010	.192 ± .050	**.377** ± .010	.656 ± .000	.395 ± .007	–
Stat.	Tool success rate	.515 ± .020	.154 ± .010	.369 ± .030	.503 ± .010	.493 ± .000	.225 ± .030	.336 ± .010	.545 ± .010	.392 ± .006	–
UQ	SP	.072 ± .000	-.060 ± .000	.387 ± .000	.106 ± .000	.118 ± .000	.030 ± .000	.257 ± .000	.010 ± .000	.115 ± .000	–
UQ	Perplexity	-.116 ± .000	-.067 ± .000	-.220 ± .000	-.211 ± .000	-.160 ± .000	-.051 ± .000	-.021 ± .000	-.213 ± .000	-.132 ± .000	–
UQ	MTE	-.135 ± .000	-.068 ± .000	-.265 ± .000	-.213 ± .000	-.083 ± .000	-.018 ± .000	-.099 ± .000	-.227 ± .000	-.138 ± .000	–
UQ	Self-certainty	.331 ± .000	.095 ± .000	.418 ± .000	.275 ± .000	.275 ± .000	-.010 ± .000	.188 ± .000	.222 ± .000	.224 ± .000	–
UQ	Verb.	.577 ± .000	.205 ± .000	.396 ± .000	.374 ± .000	.394 ± .000	.074 ± .000	.245 ± .000	.342 ± .000	.326 ± .000	–
UQ	SAUP	.156 ± .000	.153 ± .000	.263 ± .000	.285 ± .000	.176 ± .000	.106 ± .000	.097 ± .000	.234 ± .000	.184 ± .000	+.018
UQ	UProp	.586 ± .000	.249 ± .000	**.485** ± .000	.344 ± .000	.250 ± .000	.199 ± .000	.109 ± .000	.368 ± .000	.324 ± .000	–
Super-vised	HTC	.450 ± .000	.205 ± .000	.420 ± .000	.256 ± .000	.081 ± .000	-.164 ± .000	.102 ± .000	.077 ± .000	.178 ± .000	-.241
Super-vised	Logistic regression	.135 ± .000	-.034 ± .000	.209 ± .000	-.058 ± .000	.092 ± .000	.030 ± .000	.050 ± .000	-.040 ± .000	.048 ± .000	-.344
Bayes (ours)	Tool-only	.577 ± .000	.177 ± .000	.358 ± .000	.515 ± .000	.600 ± .000	.247 ± .000	.126 ± .000	.623 ± .000	.403 ± .000	+.002
Bayes (ours)	Fused (SP)	.542 ± .000	.143 ± .000	.367 ± .000	.506 ± .000	.602 ± .000	.294 ± .000	.143 ± .000	.599 ± .000	.399 ± .000	-.014
Bayes (ours)	Fused (Perplexity)	.530 ± .000	.106 ± .000	.323 ± .000	.405 ± .000	.608 ± .000	.153 ± .000	.180 ± .000	.506 ± .000	.351 ± .000	-.048
Bayes (ours)	Fused (MTE)	.529 ± .000	.126 ± .000	.336 ± .000	.505 ± .000	.620 ± .000	.255 ± .000	.170 ± .000	.645 ± .000	.398 ± .000	-.016
Bayes (ours)	Fused (Self-certainty)	.528 ± .000	.123 ± .000	.382 ± .000	.505 ± .000	.611 ± .000	.254 ± .000	.167 ± .000	.656 ± .000	.403 ± .000	-.018
Bayes (ours)	Fused (Verb.)	**.648** ± .000	**.319** ± .000	.474 ± .000	**.603** ± .000	.599 ± .000	**.404** ± .000	.166 ± .000	**.675** ± .000	**.486** ± .000	+.038
Bayes (ours)	SBF	.537 ± .000	.149 ± .000	.473 ± .000	.511 ± .000	.652 ± .000	.174 ± .000	.015 ± .000	.141 ± .000	.331 ± .000	-.108

Table 13: **Cross-dataset transfer in coding.** PRR@0.5↑ with the model and harness held fixed. Methods are trained on the other coding datasets and evaluated on the held-out dataset named in each column. Δ_{ID} is the average PRR change relative to the corresponding in-domain results on the target datasets. The best result in each column is **bold**, and the second-best is underlined.

Group	Method	LCB	BCB	SWE	CC	Avg (OOD)	Δ_{ID}
Stat.	N steps	.080 ± .000	.070 ± .000	.130 ± .000	.260 ± .000	.135 ± .000	–
	Tool success rate	.170 ± .000	.130 ± .000	.090 ± .000	.420 ± .000	.203 ± .000	–
UQ	SP	**.280** ± .000	.250 ± .000	**.360** ± .000	.550 ± .000	**.360** ± .000	–
	Perplexity	.050 ± .000	.190 ± .000	.220 ± .000	.200 ± .000	.165 ± .000	–
	MTE	.060 ± .000	.190 ± .000	.220 ± .000	.210 ± .000	.170 ± .000	–
	Self-certainty	.170 ± .000	.170 ± .000	.270 ± .000	.390 ± .000	.250 ± .000	–
	Verbalized confidence	.110 ± .000	.160 ± .000	.320 ± .000	.410 ± .000	.250 ± .000	–
	SAUP	.070 ± .000	.140 ± .000	.190 ± .000	.230 ± .000	.158 ± .000	+.002
	UProp	.070 ± .000	**.270** ± .000	.170 ± .000	.270 ± .000	.195 ± .000	–
Super-vised	HTC	.080 ± .020	-.080 ± .050	.170 ± .020	.170 ± .070	.085 ± .023	-.101
	Logistic regression	.270 ± .010	.220 ± .010	.340 ± .010	.540 ± .020	.343 ± .007	+.015
Bayes (ours)	Tool-only	.190 ± .010	.190 ± .010	.160 ± .010	.380 ± .030	.230 ± .009	+.013
	Fused (SP)	**.280** ± .000	.160 ± .010	.270 ± .000	**.580** ± .010	.323 ± .004	-.048
	Fused (Perplexity)	.130 ± .000	.180 ± .010	.240 ± .010	.260 ± .000	.203 ± .004	-.013
	Fused (MTE)	.140 ± .000	.180 ± .000	.250 ± .000	.270 ± .010	.210 ± .003	-.015
	Fused (Self-certainty)	.210 ± .000	.170 ± .000	.280 ± .010	.400 ± .010	.265 ± .004	-.013
	Fused (Verb.)	.120 ± .000	.160 ± .000	.340 ± .010	.380 ± .010	.250 ± .004	-.017
	SBF	.200 ± .010	.190 ± .010	.340 ± .000	.490 ± .010	.305 ± .004	-.034

Table 14: **Cross-environment transfer.** PRR@0.5↑ for WebArena ↔ OSWorld using Qwen3.6-35B-A3B. Δ_{ID} is the average PRR change relative to the corresponding in-domain results. The best result in each column is **bold**, and the second-best is underlined.

Group	Method	WebArena → OSWorld	OSWorld → WebArena	Avg	Δ_{ID}
Stat.	N steps	**.352** ± .000	.369 ± .000	**.361** ± .000	–
	Tool success rate	-.005 ± .000	.306 ± .000	.151 ± .000	–
UQ	SP	.224 ± .000	.185 ± .000	.205 ± .000	–
	Perplexity	.037 ± .000	.364 ± .000	.200 ± .000	–
	MTE	.058 ± .000	.373 ± .000	.215 ± .000	–
	Self-certainty	.282 ± .000	**.397** ± .000	.340 ± .000	–
	Verb.	.283 ± .000	.346 ± .000	.315 ± .000	–
	SAUP	.239 ± .007	.100 ± .029	.169 ± .015	+.046
	UProp	.047 ± .000	.315 ± .000	.181 ± .000	–
Super-vised	HTC	.073 ± .046	.206 ± .005	.140 ± .023	-.258
	Logistic regression	.281 ± .025	.220 ± .011	.251 ± .014	-.220
Bayes (ours)	Tool-only	.073 ± .063	.065 ± .017	.069 ± .033	-.174
	Fused (SP)	.213 ± .003	.225 ± .004	.219 ± .003	-.123
	Fused (Perplexity)	.170 ± .028	.289 ± .040	.230 ± .024	-.062
	Fused (MTE)	.212 ± .005	.337 ± .009	.275 ± .005	-.013
	Fused (Self-certainty)	.319 ± .013	.256 ± .063	.288 ± .032	-.076
	Fused (Verb.)	.314 ± .020	.307 ± .011	.311 ± .011	-.017
	SBF	.020 ± .046	.360 ± .003	.190 ± .023	-.227

Table 15: **Cross-domain transfer.** PRR@0.5↑ for ALFWorld ↔ DeepSWE. Differences from in-domain performance also reflect changes in the available features: transfer uses three shared critics, while in-domain evaluation uses all environment-specific critics. The best result in each column is **bold**, and the second-best is underlined.

Group	Method	ALFWorld → DeepSWE				DeepSWE → ALFWorld				Avg	Δ_{ID}
		GPT-OSS-20B		Qwen3.6-35B-A3B		GPT-OSS-20B		Qwen3.6-35B-A3B			
		ReAct → mini-swe-agent	smolagents → mini-swe-agent	ReAct → mini-swe-agent	smolagents → mini-swe-agent	mini-swe-agent → ReAct	mini-swe-agent → smolagents	mini-swe-agent → ReAct	mini-swe-agent → smolagents		
Stat.	N steps	.170 ± .000	.170 ± .000	-.110 ± .000	-.110 ± .000	.640 ± .000	.570 ± .000	**.490** ± .000	-.150 ± .000	.209 ± .000	–
	Tool success rate	-.080 ± .000	-.080 ± .000	.120 ± .000	.120 ± .000	.620 ± .000	**.930** ± .000	.120 ± .000	.070 ± .000	**.228** ± .000	–
UQ	SP	-.030 ± .000	-.030 ± .000	.240 ± .000	.240 ± .000	-.080 ± .000	.570 ± .000	.450 ± .000	.350 ± .000	.214 ± .000	–
	Perplexity	-.070 ± .000	-.070 ± .000	**.250** ± .000	**.250** ± .000	-.050 ± .000	.510 ± .000	-.010 ± .000	.550 ± .000	.170 ± .000	–
	MTE	-.020 ± .000	-.020 ± .000	.210 ± .000	.210 ± .000	-.150 ± .000	.750 ± .000	.030 ± .000	**.610** ± .000	.203 ± .000	–
	Self-certainty	.030 ± .000	.030 ± .000	.040 ± .000	.040 ± .000	-.030 ± .000	**.930** ± .000	-.130 ± .000	.590 ± .000	.188 ± .000	–
	Verb. confidence	.200 ± .000	.200 ± .000	-.180 ± .000	-.180 ± .000	.190 ± .000	.760 ± .000	-.160 ± .000	-.040 ± .000	.099 ± .000	–
	SAUP	-.080 ± .070	-.070 ± .040	.120 ± .120	.180 ± .020	-.100 ± .080	.540 ± .140	-.190 ± .080	.000 ± .040	.050 ± .029	-.144
	UProp	.180 ± .000	.180 ± .000	-.170 ± .000	-.170 ± .000	**.710** ± .000	.430 ± .000	.140 ± .000	.340 ± .000	.205 ± .000	–
Super-vised	HTC	.040 ± .070	-.040 ± .050	.020 ± .070	-.090 ± .070	.170 ± .070	.470 ± .170	.050 ± .050	.030 ± .170	.081 ± .036	-.230
	Logistic regression	.020 ± .040	-.030 ± .030	-.070 ± .130	-.040 ± .030	-.070 ± .080	-.210 ± .080	.130 ± .030	.040 ± .040	-.029 ± .024	-.499
Bayes (ours)	Tool-only	.030 ± .100	-.190 ± .020	.110 ± .040	-.110 ± .050	-.170 ± .060	-.700 ± .070	-.390 ± .020	-.210 ± .030	-.204 ± .019	-.638
	Fused (SP)	.030 ± .040	-.020 ± .010	-.120 ± .040	-.030 ± .030	-.090 ± .060	-.310 ± .140	-.090 ± .120	-.010 ± .060	-.080 ± .027	-.461
	Fused (Perplexity)	.020 ± .080	-.070 ± .010	-.090 ± .010	-.040 ± .060	-.110 ± .040	-.340 ± .190	-.040 ± .140	-.010 ± .060	-.085 ± .033	-.469
	Fused (MTE)	**.270** ± .020	-.070 ± .000	-.010 ± .030	.150 ± .090	-.120 ± .040	-.420 ± .210	-.050 ± .100	.230 ± .090	-.003 ± .034	-.368
	Fused (Self-certainty)	-.060 ± .080	.040 ± .030	.000 ± .040	-.020 ± .000	-.060 ± .100	-.560 ± .100	.010 ± .040	-.060 ± .080	-.089 ± .024	-.441
	Fused (Verb.)	.190 ± .020	**.210** ± .000	-.100 ± .040	-.050 ± .020	.090 ± .050	-.050 ± .110	.120 ± .050	.110 ± .000	.065 ± .017	-.263
	SBF	.050 ± .020	-.010 ± .050	-.020 ± .000	-.030 ± .030	.340 ± .080	.380 ± .040	-.180 ± .030	.230 ± .020	.095 ± .014	-.405

E ABLATIONS

Aggregation of UQ scores over the trajectory. Tables 16 to 18 compare three ways to aggregate step-level raw UQ scores over the trajectory: the last step, the mean, and the maximum over steps. No aggregation is uniformly best, but the preferred choice is consistent within a task family. On interactive and retrieval tasks, the last step is usually best, since the final step typically contains the answer or terminating action, e.g., on WebArena it is best for every UQ method. On coding, the mean is best on average, as errors can arise at any step that writes or edits code. The maximum is typically the weakest choice outside coding (e.g., SP drops to 0.007 on retrieval). We therefore use the last step for retrieval and interactive tasks and the mean for coding.

Table 16: **Aggregation of raw UQ scores: retrieval tasks.** In-domain PRR@0.5↑ of raw uncertainty scores aggregated over the trajectory by the last step, the mean, or the maximum over steps, averaged over the model and harness configurations of each benchmark except Qwen2.5-32B-Instruct. Avg is the mean over the benchmarks. The best aggregation for each score and column is **bold**.

Score	Agg.	HotpotQA	MuSiQue	CRAG	BrowseComp-Plus	Avg
SP	last	.126	.104	**.174**	.515	.230
	mean	**.160**	**.128**	.161	**.540**	**.247**
	max	-.041	-.026	.029	.065	.007
Perplexity	last	**-.154**	**-.111**	**-.166**	**.900**	**.117**
	mean	-.157	-.199	-.223	.730	.038
	max	-.162	-.281	-.249	.710	.005
MTE	last	-.170	**-.106**	**-.150**	**.915**	**.122**
	mean	**-.162**	-.174	-.173	.725	.054
	max	-.167	-.257	-.242	.715	.012
Self-certainty	last	.280	.169	**.134**	**.900**	**.371**
	mean	**.284**	**.218**	.121	.695	.330
	max	.072	-.019	.021	.245	.080
Verb.	last	**.388**	.264	**.469**	.715	.459
	mean	.306	**.289**	.410	.905	**.478**
	max	.298	.227	.400	**.930**	.464

Table 17: **Aggregation of raw UQ scores: coding tasks.** In-domain PRR@0.5↑ of raw uncertainty scores aggregated over the trajectory by the last step, the mean, or the maximum over steps, averaged over the model and harness configurations of each benchmark. Avg is the mean over the benchmarks. The best aggregation for each score and column is **bold**.

Score	Agg.	LCB	BCB	SWE	CC	DeepSWE	Avg
SP	last	.250	.230	**.370**	.460	-.063	.249
	mean	**.280**	.250	.360	.550	**.017**	**.291**
	max	.250	**.280**	.310	**.560**	-.030	.274
Perplexity	last	.030	.160	.200	.180	-.063	.101
	mean	**.050**	.190	**.220**	**.200**	**.007**	**.133**
	max	.020	**.200**	.210	.090	-.070	.090
MTE	last	.040	.140	.210	**.210**	**.030**	.126
	mean	**.060**	.190	.220	**.210**	.023	**.141**
	max	.040	**.200**	**.230**	.120	-.027	.113
Self-certainty	last	.140	.160	.260	**.390**	**.023**	**.195**
	mean	**.170**	**.170**	**.270**	**.390**	-.030	.194
	max	**.170**	.140	.230	.360	.010	.182
Verb.	last	**.120**	**.180**	**.320**	.340	**.030**	.198
	mean	.110	.160	**.320**	**.410**	-.007	**.199**
	max	.100	.160	**.320**	.340	-.043	.175

Table 18: **Aggregation of raw UQ scores: interactive tasks.** In-domain PRR@0.5↑ of raw uncertainty scores aggregated over the trajectory by the last step, the mean, or the maximum over steps, averaged over the model and harness configurations of each benchmark except Qwen2.5-32B-Instruct. Avg is the mean over the benchmarks. The best aggregation for each score and column is **bold**.

Score	Agg.	τ^3-retail	τ^3-airline	τ^3-telecom	WebArena	OSWorld	ALFWorld	Avg
SP	last	.126	**.153**	.012	**.435**	.220	**.323**	**.211**
	mean	**.129**	-.089	.009	.375	**.400**	.320	.191
	max	.019	.048	**.043**	.030	.080	-.085	.022
Perplexity	last	**-.090**	-.220	**.005**	**.470**	.040	.250	**.076**
	mean	-.179	**-.012**	-.357	.400	**.180**	.215	.041
	max	-.181	-.098	-.288	.310	**.180**	**.353**	.046
MTE	last	**-.083**	-.218	**-.017**	**.480**	.060	.310	**.089**
	mean	-.142	**-.021**	-.321	.385	.170	.200	.045
	max	-.186	-.121	-.297	.305	**.180**	**.323**	.034
Self-certainty	last	.114	**.087**	.066	**.545**	.280	**.340**	.239
	mean	**.186**	-.025	**.387**	.390	**.320**	.275	**.256**
	max	.041	-.066	.088	.320	.150	.002	.089
Verb.	last	.146	**.027**	.100	**.285**	**.280**	.187	**.171**
	mean	**.192**	-.026	**.211**	.220	.200	**.215**	.169
	max	.185	-.090	.153	.200	-.010	.158	.099

Components of Bayes Fused. Tables 19 to 21 present the contribution of each evidence source of Equation (9), using SP as the generation signal. The relative importance of the two sources depends on the domain. On retrieval, tool feedback is much stronger than generation uncertainty (PRR of 0.306 vs. 0.089), while on coding the ordering reverses (0.198 vs. 0.309). On interactive tasks, tool feedback is slightly stronger (0.301 vs. 0.254). Importantly, in all domains, the best combination of the two sources outperforms each of them alone.

We also compare four aggregations of the generation evidence. The results largely support the choices made in the main tables: history + last performs best on interactive tasks (0.384 vs. 0.352 for SEP), and continuous aggregation performs best on coding (0.337 vs. 0.321 for history + last). On retrieval, history + last (0.382) is close to the best variant, SEP (0.393), and substantially outperforms continuous aggregation (0.179). Continuous aggregation performs even worse than tool feedback alone, since summing one log-likelihood ratio per step lets long trajectories accumulate evidence from uninformative steps.

Note that continuous aggregation and history + last differ not only in separating the final step. Continuous aggregation adds one contribution per step, so its magnitude grows with trajectory length, and fits class-specific variances, which makes each contribution quadratic in the signal. In contrast, history + last adds exactly two contributions regardless of trajectory length and pools the variance within each temporal role, which makes each contribution linear. Thus, the comparison reflects all these design choices rather than the temporal split alone.

Training and inference cost. Table 22 provides the training and inference cost of each estimator. Training is measured on a single CPU core over nine agent runs on ALFWorld, DeepSWE, and WebArena-Verified, and inference on WebArena-Verified trajectories with GPT-OSS (OpenAI, 2025), where one agent generation takes 614 ms. Bayes Fused and Structured Bayesian Fusion take less than 0.1 s to train, offline, and scoring a step adds less than 0.001% to its LLM inference time, so their cost is negligible compared to the agent run. The cost of an estimator is therefore dominated by the signals it requires. Token-level UQ scores and tool feedback are produced during the agent run at no extra cost. Verbalized confidence requires one additional short LLM call per step, adding 317 ms, or 52% overhead relative to a generation. UProp (Duan et al., 2025) requires ten additional generations per step, adding 1.8 s, or 300% overhead relative to a generation, making it the most expensive method.

Table 19: **Components of Bayes Fused: retrieval tasks.** In-domain PRR@0.5↑ with SP, averaged over the model and harness configurations of each benchmark. The two evidence sources are separated: ℓ_C carries the prior and the tool checks, Λ_U carries generation uncertainty, and Bayes Fused is their sum. The last four rows differ only in how the generation evidence is aggregated over the trajectory: the final step only (Λ_L), the history mean plus the final step ($\Lambda_H + \Lambda_L$), a binarized per-step scores (SEP), or all steps (Λ_U). The best score in each column is **bold**. Avg is the mean over the benchmarks. HotpotQA, MuSiQue, and CRAG are averaged over the GPT-OSS-20B and Qwen3.6-35B-A3B configurations (without Qwen2.5-32B-Instruct).

Score		HotpotQA	MuSiQue	CRAG	BrowseComp-Plus	Avg
Generation evidence only	$\widehat{\ell}_0 + \Lambda_U$	.015	-.050	**.148**	.245	.089
Tool evidence only	ℓ_C	.388	.413	-.021	.445	.306
Bayes Fused, last only	$\ell_C + \Lambda_L$	.392	.357	.089	.495	.333
Bayes Fused, history + last	$\ell_C + \Lambda_H + \Lambda_L$	.410	**.417**	.117	.585	.382
Bayes Fused, SEP	$\ell_C + \Lambda_U^{\text{SEP}}$	**.427**	.399	.134	**.610**	**.393**
Bayes Fused, continuous	$\ell_C + \Lambda_U$	.217	.146	.110	.245	.179

Table 20: **Components of Bayes Fused: coding tasks.** In-domain PRR@0.5↑ with SP, averaged over the model and harness configurations of each benchmark. The two evidence sources are separated: ℓ_C carries the prior and the tool checks, Λ_U carries generation uncertainty, and Bayes Fused is their sum. The last four rows differ only in how the generation evidence is aggregated over the trajectory: the final step only (Λ_L), the history mean plus the final step ($\Lambda_H + \Lambda_L$), a binarized per-step scores (SEP), or all steps (Λ_U). The best score in each column is **bold**. Avg is the mean over the benchmarks.

Score		LCB	BCB	SWE	CC	DeepSWE	Avg
Generation evidence only	$\widehat{\ell}_0 + \Lambda_U$	.280	**.260**	.360	**.570**	.077	.309
Tool evidence only	ℓ_C	.190	.130	.170	.380	.120	.198
Bayes Fused, last only	$\ell_C + \Lambda_L$	.260	.200	.340	.410	.117	.265
Bayes Fused, history + last	$\ell_C + \Lambda_H + \Lambda_L$	**.300**	**.260**	**.370**	.530	.147	.321
Bayes Fused, SEP	$\ell_C + \Lambda_U^{\text{SEP}}$	.290	.250	.330	.500	**.227**	.319
Bayes Fused, continuous	$\ell_C + \Lambda_U$	**.300**	**.260**	.360	.560	.207	**.337**

Table 21: **Components of Bayes Fused: interactive tasks.** In-domain PRR@0.5↑ with SP, averaged over the model and harness configurations of each benchmark. The two evidence sources are separated: ℓ_C carries the prior and the tool checks, Λ_U carries generation uncertainty, and Bayes Fused is their sum. The last four rows differ only in how the generation evidence is aggregated over the trajectory: the final step only (Λ_L), the history mean plus the final step ($\Lambda_H + \Lambda_L$), a binarized per-step scores (SEP), or all steps (Λ_U). The best score in each column is **bold**. Avg is the mean over the benchmarks. The τ^3-bench columns are averaged over the GPT-OSS-20B and Qwen3.6-35B-A3B configurations (without Qwen2.5-32B-Instruct).

Score		τ^3-retail	τ^3-airline	τ^3-telecom	WebArena	OSWorld	ALFWorld	Avg
Generation evidence only	$\widehat{\ell}_0 + \Lambda_U$	.278	.090	.184	.285	**.390**	.295	.254
Tool evidence only	ℓ_C	.105	-.037	.535	.345	.210	**.648**	.301
Bayes Fused, last only	$\ell_C + \Lambda_L$	.193	-.069	.534	.500	.200	.585	.324
Bayes Fused, history + last	$\ell_C + \Lambda_H + \Lambda_L$	.308	.036	.565	**.505**	.360	.528	**.384**
Bayes Fused, SEP	$\ell_C + \Lambda_U^{\text{SEP}}$	.229	.074	**.576**	.410	.350	.475	.352
Bayes Fused, continuous	$\ell_C + \Lambda_U$	**.322**	**.107**	.448	.390	.380	.378	.337

Table 22: **Cost of each estimator.** *Training* is the one-time cost of fitting the estimator offline on a single CPU core, median over nine agent runs on ALFWorld, DeepSWE, and WebArena-Verified. *Inference* shows the extra cost of scoring one agent step: the number of additional LLM calls, the added time including these calls, and this time as a percentage of one agent generation. Inference time is the median across WebArena-Verified trajectories with GPT-OSS, with a median of 614 ms per agent generation.

Group	Method	Training	Inference		
		Time (ms)	LLM calls	Time (ms)	Overhead (%)
Stat.	N steps	–	0	<0.001	<0.001
	Tool success rate	–	0	<0.001	<0.001
UQ	SP, Perplexity, MTE, Self-certainty	–	0	<0.001	<0.001
	Verb.	–	1	317	52
	SAUP	2,595	0	0.009	0.001
	UProp	–	10	1843	300
Super-vised	HTC	760	0	0.033	0.005
	Logistic regression	399	0	0.025	0.004
Bayes (ours)	Tool-only	0.05	0	<0.001	<0.001
	Fused (SP, Perplexity, MTE, Self-cert.)	1.8	0	<0.001	<0.001
	Fused (Verb.)	0.9	1	317	52
	SBF	23	0	0.005	<0.001

F Additional Results

Effect of reasoning effort. Table 23 re-evaluates the strongest methods of Table 8 at different reasoning efforts the models expose (low, medium, and high for GPT-OSS-120B (OpenAI, 2025); none, low, and medium for Qwen3.8-27B (Qwen Team, 2026b)), averaged over LiveCodeBench-Hard and SWE-bench and over both agent harnesses. Structured Bayesian Fusion performs best at every effort level except for GPT-OSS-120B with medium thinking, and its margin over raw SP is largest in the no-thinking setup.

Table 23: **Reasoning effort.** In-domain PRR@0.5↑, averaged over LiveCodeBench-Hard and SWE-Bench and two harnesses: smolagents CodeAgent and ReAct. SP denotes raw sum-logprob averaged over the trajectory. Fused (SP), Structured Bayesian Fusion (SBF), and logistic regression are the methods reported in Table 8. GPT-OSS-120B is evaluated at low, medium, and high effort, and Qwen3.8-27B at none, low, and medium. Δ is the PRR of the best fitted method minus that of SP.

Model	Effort	SP	Fused (SP)	SBF	Logistic regression	Δ
GPT-OSS-120B	low	.340 $\pm$ .000	.355 $\pm$ .004	.405 $\pm$ .013	.382 $\pm$ .006	+.065
	medium	.294 $\pm$ .000	.327 $\pm$ .003	.316 $\pm$ .015	.303 $\pm$ .010	+.033
	high	.172 $\pm$ .000	.142 $\pm$ .006	.229 $\pm$ .013	.193 $\pm$ .014	+.057
Qwen3.8-27B	none	.280 $\pm$ .000	.319 $\pm$ .006	.379 $\pm$ .008	.337 $\pm$ .013	+.099
	low	.274 $\pm$ .000	.254 $\pm$ .009	.297 $\pm$ .013	.245 $\pm$ .010	+.023
	medium	.261 $\pm$ .000	.255 $\pm$ .004	.306 $\pm$ .010	.264 $\pm$ .012	+.045

G Limitations

While our methods substantially improve failure detection for AI agents, no uncertainty quantification method is perfect. Some failures may remain undetected, and some successful trajectories may still receive low confidence. Therefore, these methods should be viewed as assistance for verification rather than as an endpoint oracle. In addition, although we evaluate across a wide range of environments, models, and harnesses, our evaluation is limited to single-agent systems and does not cover multi-agent systems.