Factor-Graph Based Calibration For LLM Agents In Complex Reasoning

Anonymous ACL submission

Abstract

LLM Agents achieve strong performance on complex reasoning tasks but remain poorly calibrated, often assigning unjustified confidence to incorrect intermediate steps. Most existing confidence estimators operate primarily at the answer level, and therefore fail to model how uncertainty evolves and propagates across intermediate reasoning steps. Factor graphs provide a natural fit for this setting: they decompose global trajectory uncertainty into local dependency factors and use message passing to propagate and fuse evidence across steps, yielding principled step-wise belief updates. Motivated by this alignment, we propose the first framework that converts an agent's multi-step trajectory into a factor graph for step-wise confidence modeling, explicitly capturing step-to-step dependencies so that uncertainty can be propagated and calibrated throughout the entire reasoning process. Building on this factor-graph view, we further leverage reinforcement learning to align the model's verbal confidence with calibrated step-level uncertainty estimates. Experiments on six QA benchmarks demonstrate improved calibration and stronger overall performance. [1]

1 Introduction

LLMs have demonstrated strong performance across a diverse spectrum of tasks such as scientific reasoning, coding, web browsing, etc (Achiam et al., 2023; Phan et al., 2025; Jimenez et al., 2023; Wu et al., 2025; Wei et al., 2025). Nevertheless, even the strongest LLMs remain prone to hallucinations and overconfident errors (Kalai et al., 2025; Chhikara, 2025; Kapoor et al., 2024). Accurate self-assessment of confidence thus becomes essential, as well-calibrated confidence scores allow users and downstream systems to decide when to rely on model outputs (Tao et al., 2024; Zhang et al., 2025). Prior work has examined confidence estimation methods such as verbalized scores, token probabilities, and self-reflection (Kotelanski et al., 2023; Zhang et al., 2024; Wang et al., 2024; Li et al., 2024), but these approaches mostly target single-step tasks. In contrast, confidence estimation in long-horizon agentic settings involving external input and adaptive reasoning remains underexplored (Yao et al., 2024; Barres et al., 2025; Patil et al.; Ou et al., 2025). These scenarios introduce more challenges for LLM agents, as prior work has revealed their tendency to forget previously acquired information and their difficulty in recovering from earlier errors. This ultimately results in an unreliable estimation of confidence in the final output (Chen et al., 2025; Zhao et al., 2024; Liu et al., 2024). Recent work has begun to explore agent-level uncertainty estimation by modeling entire reasoning trajectories. These methods typically rely on heuristic aggregation strategies and lack a principled probabilistic framework (Zhao et al., 2024). They cannot explicitly model the structural dependencies among states, actions, and observations across steps, nor do they provide a mechanism to correct miscalibrated beliefs in a step-wise and interpretable manner.

Factor graph (Kschischang et al., 2002; Loeliger, 2004) are well-suited for long-horizon reasoning because they decompose the global uncertainty of a multi-step trajectory into a set of local dependency factors and use message passing to propagate and fuse evidence across steps, thereby naturally enabling step-wise confidence calibration. Motivated by this insight, we propose a structured, calibrated, and interpretable framework for multi-step uncertainty modeling: a factor-graph calibration system for agentic trajectories. We first represent each multi-step trajectory via a State–Action–Observation decomposition, and then convert it into a probabilistic factor graph whose factors explicitly capture step-to-step dependencies. In this

[1] Our Code is available at https://anonymous.4open.science/r/Graph_Calibration-D7C0

graph, token-level model probabilities provide priors over step confidences, while supervision from an external judge is injected as evidence. By performing bidirectional belief propagation (Murphy et al., 2013), the agent can revise miscalibrated beliefs using evidence, yielding globally consistent posterior confidence estimates across the entire trajectory. Building on these calibrated posteriors, we further introduce a step-wise calibration-guided reinforcement learning objective to directly calibrate the agent's verbal confidence. Specifically, we treat the posterior beliefs produced by factor-graph inference as step-level calibration targets and optimize the policy so that its declared confidence aligns with these evidence-corrected probabilities. Our method provides step-level, evidence-grounded supervision, encouraging the agent to adjust confidence appropriately when evidence is missing, noisy, or misleading. Experiments on six QA benchmarks demonstrate improved calibration and stronger overall performance, highlighting the effectiveness of factor-graph-based uncertainty modeling for long-horizon LLM agents.

2 Preliminary

2.1 SAO Decomposition Framework

We study interactive tasks with partial observations, following prior work (Song et al., 2024; Qiao et al., 2024) and formulate them as a POMDP $(\mathcal{U}, \mathcal{S}, \mathcal{A}, \mathcal{O}, \mathcal{T}, \mathcal{R})$ interaction. Here, $\mathcal{U}$ denotes natural-language instructions, and $\mathcal{S}$, $\mathcal{A}$, and $\mathcal{O}$ are the state, action, and observation spaces, respectively. The environment defines the transition function $\mathcal{T}$ and the reward function $\mathcal{R} : \mathcal{S} \times \mathcal{A} \to [0, 1]$. For language agents, all components are represented in natural language. We adopt ReAct (Yao et al., 2022), which generates rationales before actions. At time step t, the trajectory prefix is

$$\tau_t = (a_1, o_1, \ldots, a_t, o_t) \sim \pi_\theta(\tau_t \mid u) \quad (1)$$

The above formulation abstracts the agent–environment interaction but omits the agent's internal reasoning dynamics. We therefore introduce a **SAO decomposition** (depicted in Fig 1 Phase 1), representing each step t with three latent variables: a belief state S_t, an action A_t (e.g., retrieval or answer), and the resulting observation O_t that updates S_t. The resulting trajectory prefix is

$$\tau_t = (s_1, a_1, o_1, \ldots, s_t, a_t, o_t) \sim \pi_\theta(\tau_t \mid u) \quad (2)$$

2.2 Factor Graph

A factor graph is a bipartite probabilistic graphical model that represents the factorization of a joint probability distribution (Loeliger, 2004; Kschischang et al., 2002). It consists of two types of nodes, variable nodes and factor nodes, connected through edges that represent probabilistic dependencies. Each factor node encodes a local potential function over its neighboring variables, allowing inference through message passing. Forward belief propagation performs iterative message passing between factor nodes and variable nodes on a factor graph: factor-to-variable messages are computed by marginalizing all other variables, and the prior belief $b^-(X)$ of a variable is obtained by taking the product of incoming factor messages followed by normalization. Backward belief propagation then propagates an additional set of backward messages β from the reverse direction and fuses them with the forward information α via normalization to produce the posterior belief $b^+(X)$, thereby integrating evidence from both sides of the graph. More Detail about factor graph in Appendix A.2.

3 Factor Graph From Agent Trajectory

To model and calibrate uncertainty propagation along multi-step reasoning trajectories, we introduce a factor-graph formulation. We convert each agent trajectory into a probabilistic graph that overlays the implicit reasoning of the model, combining span-level confidence signals, step-wise dependency factors, and outcome-level evidence to produce globally consistent calibrated confidence along the trajectory, depicted in Fig 1 Phase 2.

3.1 Overall Graph Structure

Given a reasoning trajectory of length T, we represent each step using three variable nodes: $((S_t, A_t, O_t), t = 0 \ldots T-1)$ denoting respectively the internal state, action reliability, and observation correctness at step t. These variable nodes are connected through five types of factor nodes, each encoding a specific probabilistic dependency.

(1) Prior Factor: $\phi_{\text{prior}} S_t$ and $\phi_{\text{prior}} A_t$ define the model's intrinsic confidence in the state and action before receiving any external evidence, see Sec 3.2 for detail.

(2) Transfer Factor: $\phi_{S_t A_t}$ and $\phi_{O_{t-1} S_t}$ capture the dependencies between consecutive reasoning variables, linking each state with its corresponding action and connecting the previous observation

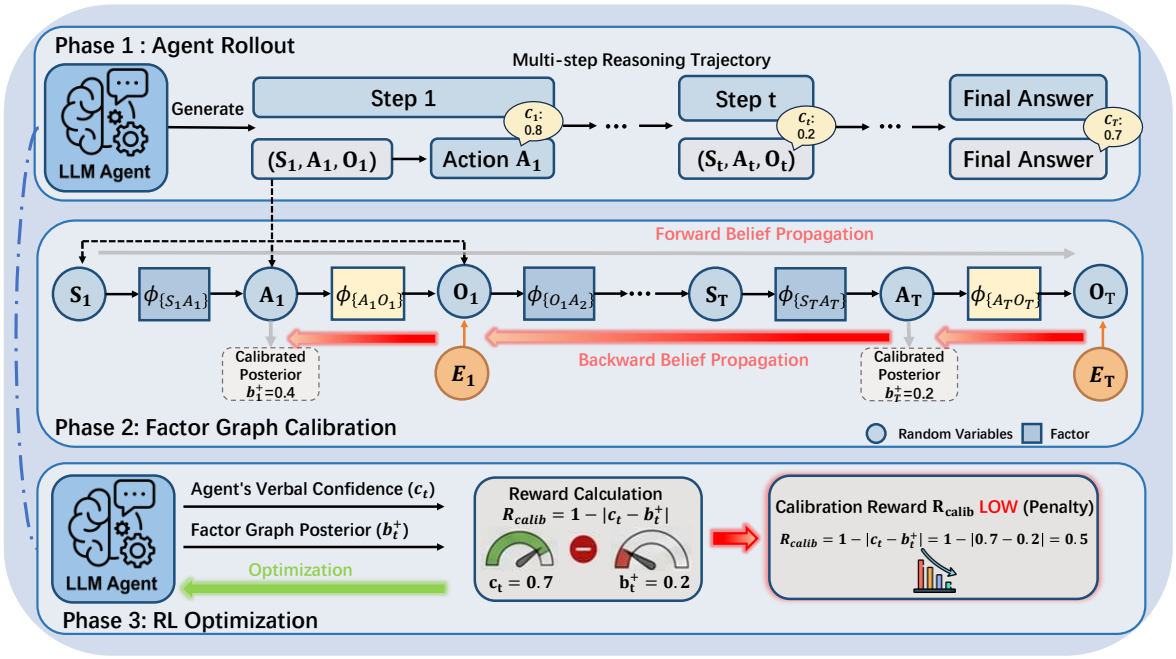

Figure 1: Overview of calibration-guided reinforcement learning framework. Phase 1 rollouts multi-step trajectories. Phase 2 performs factor-graph calibration via forward belief propagation and backward belief propagation to obtain calibrated posteriors. Phase 3 computes step-wise calibration rewards and updates the policy.

with the current state. These factors are parameterized by a confidence prediction transformer (CPT), which models the transition likelihoods between adjacent variables, see Sec 4 for detail.

(3) Observation Factor: $\phi_{A_tO_t}$ enforce consistency between actions and their resulting observations. They are modeled as near-identity matrices:

$$\phi_{A_tO_t}(A_t, O_t) = \begin{bmatrix} 1-\varepsilon & \varepsilon \\ \varepsilon & 1-\varepsilon \end{bmatrix}, \quad \varepsilon \ll 1$$

implying that a correct action is highly likely to produce a consistent observation.

3.2 Priors from Internal Confidence

Inspired by (Fu et al., 2025), we derive span-level confidence solely from the model's token-level sampling probabilities obtained from the vLLM (Kwon et al., 2023) decoding path. Our training template is depicted in A.6. For each tagged segment (<think>, <search>, <answer>), we collect the sampling probability assigned to every emitted token within that segment, and define the segment confidence as the arithmetic average of these token probabilities. We then map the resulting confidences for the reasoning and action segments into priors over the corresponding variable nodes:

$$\phi_{\text{prior}}(S_t) = [1-c_S,\ c_S], \phi_{\text{prior}}(A_t) = [1-c_A,\ c_A]$$

where c_S and c_A are the span level confidence extracted from the <think> and <search> or <answer>. We parse the value enclosed in the <confidence> </confidence> tags as the self-reported verbal confidence for the current action.

3.3 Exponentially Decayed Evidence Injection

When the model's final answer is verified against the ground truth, the correctness signal is introduced as binary evidence attached to observation nodes O_t. The evidence strength decays exponentially from the last step backward:

$$s_t = \max(s_{\min},\ s_{\max}\,\gamma^{T-1-t})$$

$$\mathbf{e}_t = \begin{cases} [1-s_t,\ s_t], & \text{if correct} \\ [s_t,\ 1-s_t], & \text{if incorrect} \end{cases}$$

This introduces an RL-style discount factor γ to exponentially decay the final correctness evidence backward across earlier steps, avoiding over-penalizing early steps.

3.4 Forward and Posterior Inference

For each reasoning trajectory, two inference passes are performed to model the propagation and correction of uncertainty.

(1) In the **forward pass**, message passing is conducted from left to right using confidence-derived priors, producing the marginal distributions $b^{-}(X)$ that describe the model's evolving belief states along the reasoning path.

(2) In the **backward pass**, external evidence derived from the correctness of the final answer is injected into the observation nodes and propagated backward through the graph, yielding the calibrated posteriors $b^{+}(X)$. These posterior beliefs serve as the calibrated ground-truth probabilities for each variable node. All inference is implemented using Loopy Belief Propagation (LBP) (Murphy et al., 2013; Ihler et al., 2005) with normalization to ensure consistent marginal probabilities under approximate inference.

4 Construction CPT for Variable Node

To obtain probabilistic estimates for variable nodes within the factor graph framework, we train a **Confidence Prediction Transformer** that learns to approximate the conditional probability tables (CPT) governing node transitions, depicted in 2.

4.1 Trajectory Collection

We first collect multi step reasoning trajectories from manus agent rollouts (Liang et al., 2025), where each trajectory $\tau = \{(s_t, a_t, o_t)\}_{t=0}^{T}$ consists of intermediate reasoning states s_t, actions a_t, and corresponding observations o_t. Each triple captures one local transition in the agent's belief–propagation process, see Appendix A.3 for more information.

4.2 SA And OS Decomposition

We first collect multi step trajectories $\tau = \{(s_t, a_t, o_t)\}_{t=0}^{T}$ with a strong closed–source model (GPT-5) , then assign a binary label using ground truth: if the final answer matches the ground truth, the whole trajectory is labeled $y = 1$; otherwise $y = 0$. We assume label consistency along the path, i.e., every step in a correct (incorrect) trajectory is treated as correct (incorrect). We then decompose τ into two supervised edge types:
(1) State–Action (SA) pairs (x^{SA}, y) with $x^{\mathrm{SA}} = (\text{history context}, a_t)$, capturing the quality of proposing a_t from the current reasoning state.
(2) Observation–State (OS) pairs (x^{OS}, y) with $x^{\mathrm{OS}} = (\text{history context}, s_{t+1})$, measuring the consistency of the next state with current evidence.

For each pair, we additionally obtain a soft reliability score $p \in [0, 1]$ by prompting an external LLM judge, forming triplets (x^{SA}, y, p) and (x^{OS}, y, p). We then filter the data by retaining only those pairs whose judged confidence aligns with the binary label, i.e., $|p - y| < 0.3$. The remaining SA pairs ($\mathcal{D}_{\mathrm{SA}}$) and OS pairs ($\mathcal{D}_{\mathrm{OS}}$) are used to train and evaluate the CPT model for conditional probability estimation; dataset statistics are provided in Appendix A.3.1.

4.3 Mixed Data Training

We combine $\mathcal{D}_{\mathrm{SA}}$ and $\mathcal{D}_{\mathrm{OS}}$ into a unified dataset $\mathcal{D} = \mathcal{D}_{\mathrm{SA}} \cup \mathcal{D}_{\mathrm{OS}}$ and train a lightweight transformer classifier $P_\theta(y{=}1 \mid x)$ to approximate the conditional probabilities of variable–node transitions. Soft supervision is applied through the objective

$$\mathcal{L}(\theta) = -\mathbb{E}_{(x,p)\sim\mathcal{D}}\Big[p \log P_\theta(y{=}1 \mid x) + (1-p)\log(1 - P_\theta(y{=}1 \mid x))\Big] \tag{3}$$

which aligns the model's predicted node–transition likelihoods with the judged confidences.

4.4 Integration Into Factor Graph

A post–hoc temperature–bias calibration (T, b) is optimized on a held–out validation set via logistic regression: $\hat{p} = \sigma((d + b)/T)$ where $d = \ell_1 - \ell_0$ is the logit margin. The calibrated confidence $\hat{p}$ serves as the estimated conditional probability for the corresponding variable node in the factor graph. During inference, each factor node consumes $\hat{p}$ values from connected CPT predictors, and the graph performs belief propagation to update marginal distributions across reasoning steps.

4.5 CPT Evaluation

To assess both the prediction quality and probability calibration of the CPT model, we compare its predicted probability $\hat{p} = P_\theta(y{=}1 \mid x)$ against the reference target p provided by an external LLM judge. We report a suite of probability-level metrics that capture complementary aspects of calibration and distributional mismatch, including ℓ_1 error, MSE, soft cross-entropy (Soft-CE), KL divergence, and Jensen–Shannon (JS) divergence. We evaluate CPT calibration across four backbones: Qwen2.5-7B-Instruct, Qwen3-4B-Instruct, Qwen3-4B-Base, and Qwen3-1.7B-Base (Team et al., 2024; Yang et al., 2025). For each backbone, we compute all metrics both *before* and *after* applying our calibration procedure. The full evaluation protocol and metrics are provided in Appendix A.3.3.

5 Step-wise Calibration-Guided RL

To integrate calibration into RL optimization, we design a calibration reward based on the posterior probabilities estimated via factor-graph inference, depicted in Fig 1 Phase 3.

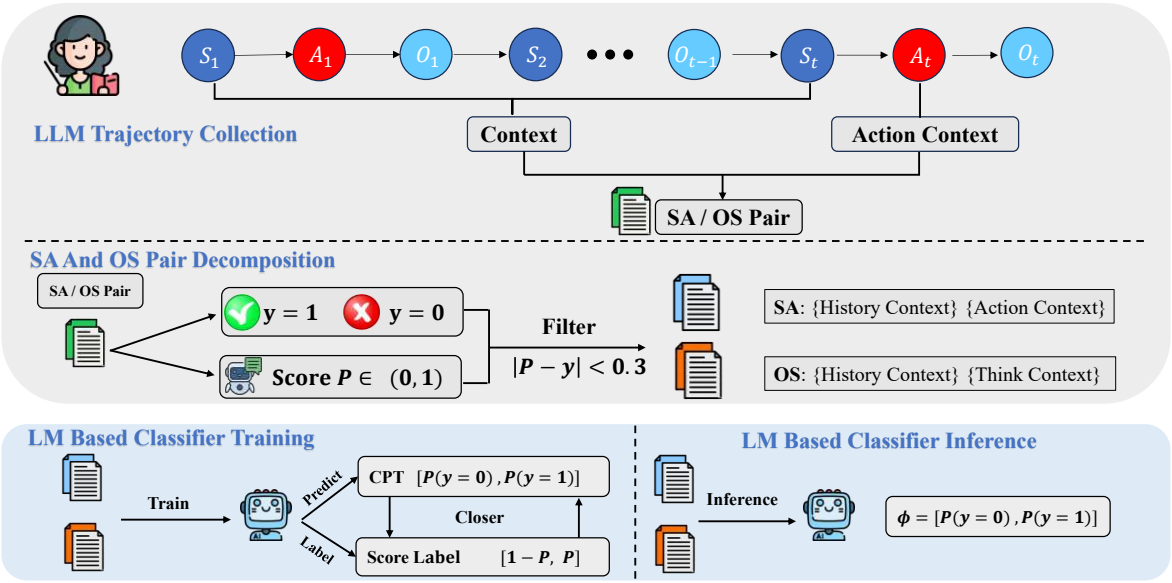

Figure 2: Construction of SA/OS pairs and training of the Confidence Prediction Transformer(CPT) classifier.

5.1 Step-wise Reward

For each reasoning step t, we compute a step-wise score s_t that combines format correctness, confidence quality, calibration agreement.

(1) Format Gating A step is rewarded only if its textual structure satisfies all required formatting constraints. We define a step as *format-valid* when the generated segment contains:

- **One pair of reasoning tags:** it must include one `<think></think>` block;
- **One pair of confidence tags:** it must include at least one `<confidence></confidence>` block;
- **One action tag:** `<search></search>` and `<answer></answer>`, *exactly one* must appear.

Formally, a step is valid if and only if:

$$\text{valid}_t = \mathbb{I}\left(\begin{array}{l}\texttt{<think>} \wedge \texttt{<confidence>} \\ \wedge \left(\texttt{<search>} \oplus \texttt{<answer>}\right)\end{array}\right)$$

where $\oplus$ denotes exclusive OR.
We then define the format reward as:

$$R_t^{\text{fmt}} = \begin{cases} 1, & \text{if the step is format-valid} \\ 0, & \text{otherwise} \end{cases} \tag{4}$$

(2) Calibration Agreement Each step predicts a verbal confidence $c_t \in [0, 1]$ extracted from the model's generated score. This confidence reflects the model's self-reported confidence in the step's correctness. The factor-graph inference produces a posterior marginal p_t for the step action. We measure the agreement between verbal confidence and posterior belief via:

$$R_t^{\text{calib}} = 1 - |c_t - p_t| \tag{5}$$

This encourages verbal confidence to be numerically consistent with probabilistic inference.

(3) Combined Step-wise Score The final step reward is a normalized weighted combination:

$$s_t = R_t^{\text{fmt}} + R_t^{\text{calib}} \tag{6}$$

5.2 Trajectory-level Reward Aggregation

Given a trajectory with T reasoning steps, the trajectory reward aggregates the step-wise scores, together with a terminal correctness bonus:

$$R_{\text{steps}} = \frac{1}{T}\sum_{t=1}^{T} s_t, \tag{7}$$

$$R_{\text{traj}} = R_{\text{steps}} + R_{\text{final}} \tag{8}$$

where $R_{\text{final}} = 1$ if the final predicted answer is correct and 0 otherwise; Algorithm 1 for detail.

6 Experimental Settings

6.1 Datasets

We evaluate on six benchmark datasets, categorized as follows: (1) General Question Answering: NQ (Kwiatkowski et al., 2019) , TriviaQA (Joshi et al., 2017), and PopQA (Mallen et al., 2022). (2) Multi-Hop Question Answering: HotpotQA (Yang et al., 2018), 2WikiMultiHopQA (Ho et al., 2020), and Bamboogle (Press et al., 2022).

Algorithm 1: Trajectory-Level Reward

Input: Rollout trajectory $\tau = \{(S_t, A_t, O_t)\}_{t=1}^{T}$
Output: Total trajectory reward R_{traj}

1. Forward Belief Propagation
Propagate beliefs forward without evidence.
Compute forward marginals:
$b_t^-(S_t), b_t^-(A_t), b_t^-(O_t)$

2. Backward Belief Propagation
Propagate beliefs backward with evidence.
Compute posterior marginals:
$b_t^+(S_t), b_t^+(A_t), b_t^+(O_t)$

3. Step-Wise Reward Computation
$R_{\text{traj}} \leftarrow 0$
for $t = 1$ **to** T **do**
  Format validation:
  $\text{valid}_t \leftarrow \mathbb{I}\left(\begin{array}{l}\texttt{<think>} \wedge \texttt{<confidence>} \\ \wedge\ (\texttt{<search>} \oplus \texttt{<answer>})\end{array}\right)$
  if *valid*$_t = 0$ **then**
    $R_t^{\text{fmt}} \leftarrow 0, \quad s_t \leftarrow 0;$
    continue
  end
  $R_t^{\text{fmt}} \leftarrow 1$
  Calibration agreement:
  Get verbal confidence c_t and posterior belief $p_t = b_t^+(A_t)$.
  $R_t^{\text{calib}} \leftarrow 1 - |c_t - p_t|$
  Combined step reward:
  $s_t \leftarrow R_t^{\text{fmt}} + R_t^{\text{calib}}$
  Accumulate:
  $R_{\text{traj}} \leftarrow R_{\text{traj}} + s_t$
end
4. Terminal Bonus
$R_{\text{traj}} \leftarrow \frac{1}{T} R_{\text{traj}} + R_{\text{final}}$
return R_{traj}

6.2 Trajectory-Level Confidence Aggregation

After obtaining the step-wise confidence scores c_t at each reasoning step, we aggregate them into trajectory-level confidence metrics. For a trajectory consisting of T reasoning steps with per-step confidences $c_t{}_{t=1}^{T}$, following (Zhao et al., 2025a), we compute five aggregated statistics: (1) arithmetic mean, (2) root mean square, (3) geometric mean, (4) maximum, and (5) KL-weighted mean. The detailed definitions of these metrics and aggression method are provided in Section A.4.

6.3 Evaluation Metrics

We evaluate calibration from two levels of granularity: trajectory-level and step-level—and further analyze Evidence Gain.

(1) Trajectory-Level Calibration We first measure whether the *aggregated* confidence of an entire reasoning trajectory aligns with its final correctness. We report the standard **Expected Calibration Error (ECE)** (Guo et al., 2017; Chen et al., 2022):

$$\text{ECE} = \sum_{b=1}^{B} \frac{n_b}{N} |c_b - a_b| \tag{9}$$

and **AUCROC** (Hendrycks and Gimpel, 2016), which quantifies the discriminative ability of trajectory-level confidence using:

$$\text{AUCROC} = \int_0^1 \text{TPR}\big(\text{FPR}^{-1}(x)\big)\, dx \tag{10}$$

These metrics capture global correctness calibration but do not reveal whether the model is calibrated within a multi-step reasoning process.

(2) Step-wise Calibration (Step-ECE) To assess calibration at a finer granularity, we introduce **Step-wise Judge-ECE**, which measures how well the model's step-level confidence aligns with step correctness assessed by an external LLM judge. For each example i with T_i reasoning steps, let $c_{i,t}$ denote the model-reported confidence at step t, and let $p_{i,t}$ denote the judge-assigned correctness score.

$$\text{Step-ECE} = \frac{1}{M} \sum_{i=1}^{M} \left(\frac{1}{T_i} \sum_{t=1}^{T_i} |c_{i,t} - p_{i,t}| \right) \tag{11}$$

Unlike trajectory-level calibration metrics that only evaluate the final answer, Step-wise Judge-ECE directly captures whether the agent remains calibrated throughout intermediate reasoning.

(3) Evidence Gain on ECE and AUC To directly measure how external information affects calibration quality, we introduce **ECE Gain** and **AUC Gain** by comparing the same model evaluated *with evidence* versus *without evidence*. Let ECE^{with} and ECE^{no} denote the ECE under the two settings, and AUC^{with} and AUC^{no} denote the AUROC scores for confidence correctness discrimination.

$$\text{ECE-Gain} = \text{ECE}^{\text{no}} - \text{ECE}^{\text{with}} \tag{12}$$

$$\text{AUC-Gain} = \text{AUC}^{\text{with}} - \text{AUC}^{\text{no}} \tag{13}$$

A positive ECE-Gain indicates that external information reduces miscalibration, while a positive AUC-Gain shows that external information strengthens the separability between correct and incorrect steps.

(4) Task Accuracy Finally, we report end-task accuracy to confirm that improvements in calibration do not compromise reasoning performance.

Dataset	Model	GMEAN						RMS					
		ECE-F	ECE-B	ECE-V	AUC-F	AUC-B	AUC-V	ECE-F	ECE-B	ECE-V	AUC-F	AUC-B	AUC-V
HotpotQA	3B-Base	0.6776	0.5496	0.5010	0.4940	0.8172	0.4876	0.7038	0.6415	0.5278	0.5041	0.7792	0.5409
	Base-RL	0.5099	0.3519	0.3646	0.6878	0.9387	0.6330	0.5404	0.4796	0.3923	0.6809	0.9181	0.6466
	3B-Instruct	0.5115	0.3585	0.3754	0.4984	0.9480	0.5390	0.5293	0.4459	0.4284	0.5002	0.9350	0.5198
	Ins-RL	**0.2864**	0.2266	**0.2546**	**0.6551**	0.9452	**0.6678**	**0.3183**	0.2961	**0.2713**	**0.6598**	0.9163	**0.6767**
PopQA	3B-Base	0.6257	0.5137	0.4240	0.5133	0.8601	0.6188	0.6584	0.6016	0.4503	0.4988	0.8182	0.6569
	Base-RL	0.4035	0.2973	0.2244	0.6801	0.9516	0.7019	0.4369	0.3936	0.2521	0.6776	0.9207	0.7138
	3B-Instruct	0.4965	0.3412	0.3008	0.5928	0.9633	0.6763	0.5102	0.4342	0.3875	0.5929	0.9481	0.6000
	Ins-RL	**0.2703**	0.2032	**0.2040**	**0.6816**	0.9451	**0.7427**	**0.2844**	0.2705	**0.2077**	**0.6792**	0.9254	**0.7286**
2WikiMultiHopQA	3B-Base	0.7047	0.5895	0.4927	0.5125	0.8414	0.5416	0.7327	0.6773	0.5294	0.5044	0.8073	0.5605
	Base-RL	0.4574	0.3281	0.3381	0.6058	0.8901	0.6463	0.4956	0.4439	0.3558	0.5999	0.8422	0.6504
	3B-Instruct	0.4627	0.3466	0.3315	0.4960	0.9223	0.5315	0.4855	0.4186	0.3670	0.4965	0.8962	0.5324
	Ins-RL	**0.2902**	0.2298	**0.2389**	**0.6464**	0.9044	**0.6471**	**0.3120**	0.2913	**0.2533**	**0.6533**	0.8683	**0.6399**
Bamboogle	3B-Base	0.7439	0.6048	0.5887	0.7115	0.9398	0.4643	0.7819	0.7078	0.6123	0.7115	0.9132	0.4524
	Base-RL	0.5578	0.4078	0.3709	0.6591	0.9185	0.6426	0.5909	0.5396	0.4144	0.6646	0.8965	0.6591
	3B-Instruct	0.5568	0.4210	0.4121	0.5237	0.9375	0.6576	0.5665	0.5152	0.4822	0.5039	0.9341	0.6562
	Ins-RL	**0.4184**	0.2710	**0.3821**	**0.6872**	0.9579	**0.7018**	**0.4483**	0.3904	**0.4035**	**0.6928**	0.9509	**0.7012**
NQ	3B-Base	0.6131	0.4813	0.4569	0.5053	0.8498	0.5998	0.6345	0.5691	0.4796	0.5059	0.8127	0.6248
	Base-RL	0.3830	0.2744	0.1943	0.7688	0.9549	0.6501	0.4041	0.3773	0.2100	0.7597	0.9452	0.6884
	3B-Instruct	0.4552	0.3457	0.3537	0.5654	0.9462	0.6129	0.4842	0.4229	0.3892	0.5641	0.9394	0.5967
	Ins-RL	**0.2586**	0.1981	**0.2306**	**0.6901**	0.9483	**0.7117**	**0.2791**	0.2677	**0.2342**	**0.6905**	0.9257	**0.7177**
TriviaQA	3B-Base	0.5676	0.4697	0.4105	0.4581	0.8393	0.6530	0.5943	0.5437	0.4375	0.4561	0.7963	0.6775
	Base-RL	0.3355	0.2446	0.1738	0.6567	0.9418	0.6992	0.3607	0.3287	0.2019	0.6542	0.9097	0.7006
	3B-Instruct	0.3266	0.2461	0.2459	0.5311	0.9225	0.5893	0.3393	0.2977	0.2625	0.5193	0.9091	0.5714
	Ins-RL	**0.1265**	0.1058	**0.1036**	**0.7474**	0.9624	**0.7365**	**0.1334**	0.1360	**0.1019**	**0.7431**	0.9486	**0.7393**

Table 1: Unified calibration results across all datasets and models, reported under both GMEAN and RMS aggregation schemes. ECE-F, ECE-B, and ECE-V measure calibration error using, respectively: (F) forward beliefs obtained without evidence injection, (B) backward beliefs obtained through factor-graph evidence propagation, and (V) verbal confidence produced by the model. Lower ECE and higher AUC indicate better calibration.

7 Experimental Results

7.1 Evidence Propagation Validity

To assess whether the SAO factor-graph structure provides meaningful probabilistic corrections, we compare forward beliefs (F)—computed without any external evidence—with backward beliefs (B), which incorporate final-answer correctness through evidence injection (depicted in Sec 3.3) and backward message passing. As shown in Table 1, Across all datasets and model variants, ECE-B is consistently and substantially lower than ECE-F, while AUC-B is uniformly higher than AUC-F. These results indicate that backward belief propagation effectively corrects model uncertainty—bringing posterior estimates closer to true correctness—and simultaneously improves discriminability between correct and incorrect trajectories.

7.2 Effectiveness of RL Calibration

We evaluate whether the proposed calibration-guided RL improves confidence calibration. As shown in Table 1, Across all datasets, both Base-RL and Ins-RL models exhibit consistently lower ECE and higher AUC compared to their respective non-RL baselines (3B-Base and 3B-Instruct), demonstrating that RL substantially enhances both structural and verbal calibration. The RL drives forward beliefs closer to ground-truth correctness distributions, yielding more calibrated internal reasoning states (ECE-F ↓, AUC-F ↑), while also improving the reliability of verbalized confidence, which becomes more aligned with posterior beliefs (ECE-V ↓, AUC-V ↑). Here, ECE-V and AUC-V are computed using the *verbalized confidence*.

7.3 Accuracy Improvements After RL

Beyond calibration, our RL framework consistently yields end-task accuracy gains across all datasets (Table 2). We compare standard (3B-Base, 3B-Instruct) and retrieval-augmented (Search-R1-Base and Search-R1-Ins (Jin et al., 2025)) baselines and our RL-calibrated counterparts (Base-RL and Ins-RL). The accuracy progression is clear: RL improves both Base and Instruct backbones, with gains persisting in both the no-evi (no information retrieval) and with-evi (with information retrieval) settings. Specifically, Base-RL provides provements over 3B-Base, while Ins-RL further strengthens the stronger 3B-Instruct baseline.

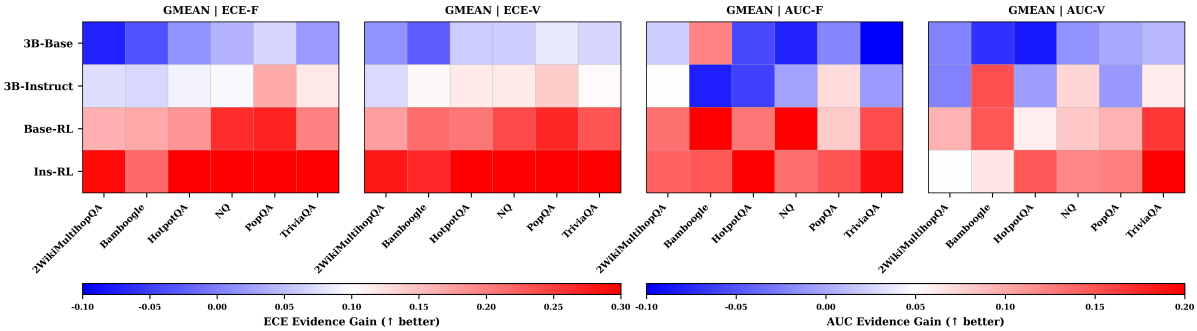

Figure 3: Evidence gain under the GMEAN aggregation setting. Heatmaps report **ECE-Gain** (left two; ECE-F and ECE-V) and **AUC-Gain** (right two; AUC-F and AUC-V). Positive ECE-Gain indicates reduced miscalibration with retrieval evidence. Positive AUC-Gain indicates improved correct incorrect separability with retrieval evidence.

Dataset	3B-Base		3B-Instruct		R1-Base		R1-Ins		Base-RL		Ins-RL	
	No-Evi	With-Evi	No-Evi	With-Evi	No-Evi	With-Evi	No-Evi	With-Evi	No-Evi	With-Evi	No-Evi	With-Evi
HotpotQA	0.0873	0.1152	0.1786	0.2656	0.1387	0.2559	0.1426	0.2207	0.1151	0.1797	0.1409	**0.4082**
PopQA	0.0913	0.1738	0.1250	0.2891	0.1055	0.2958	0.1230	0.2207	0.1270	0.3066	0.1091	**0.4648**
2wikimultihopQA	0.1706	0.1113	0.2976	0.3496	0.2188	0.2539	0.2129	0.2227	0.1905	0.2168	0.2817	**0.4238**
Bamboogle	0.1094	0.0469	0.1953	0.2518	0.1677	0.1424	0.1382	0.1635	0.1406	0.1659	0.1766	**0.2945**
NQ	0.1528	0.2090	0.2579	0.3555	0.2441	0.3926	0.2207	0.2637	0.2202	0.3730	0.2143	**0.4727**
TriviaQA	0.2183	0.2520	0.4147	0.5273	0.3594	0.5176	0.3691	0.3984	0.3194	0.3867	0.3948	**0.6504**

Table 2: Unified accuracy across six QA datasets for different model variants, evaluated under two settings: *No-Evi* (without retrieved information) and *With-Evi* (with retrieved information). Best results are highlighted in **bold**.

Dataset	3B-Base	3B-Instruct	R1-Base	R1-Ins	Base-RL	Ins-RL
HotpotQA	0.3591	0.3228	0.3676	0.3196	0.2610	**0.2094**
PopQA	0.3671	0.3195	0.3864	0.3440	0.2750	**0.2310**
2WikiMultihopQA	0.3738	0.3373	0.3570	0.3287	0.2635	**0.1942**
Bamboogle	0.3895	0.3319	0.3894	0.3562	0.2677	**0.2189**
NQ	0.3442	0.3267	0.3868	0.3295	0.2536	**0.1915**
TriviaQA	0.3564	0.3466	0.3717	0.3059	0.2653	**0.2220**

Table 3: Step-level calibration results measured by **Step-Judge-ECE** (lower is better) across six QA datasets. Best results are highlighted in **bold**.

7.4 Step-Judge-ECE

We further evaluate step-level calibration using Step-Judge-ECE in Equation 11, which measures the discrepancy between the model's step confidence and the step-wise correctness probability predicted by LLM judge. As shown in Table 3, RL-based calibration consistently and substantially reduces Step-level ECE across all datasets. These results indicate that our RL-based calibration not only improves confidence alignment at the final-answer level, but also progressively mitigates overconfidence throughout the reasoning process, making the model's step-wise confidence signals more faithful and reliable.

7.5 Evidence Gain

Figure 3 reports evidence gains under the GMean aggregation setting. Across datasets, ECE-Gain (ECE-F/ECE-V) is predominantly positive, indicating that incorporating evidence generally reduces miscalibration; these gains are strongest and most consistent for the RL-calibrated policies, whereas non-RL baselines show smaller or occasionally mixed improvements. In parallel, AUC-Gain (AUC-F/AUC-V) is also largely positive, suggesting improved correct–incorrect separability when evidence is available. Taken together, the larger and more stable gains under RL imply that RL induces an *evidence-responsive* confidence mechanism: the model more reliably updates both its beliefs and expressed confidence in the presence of informative observations, and the simultaneous improvements in calibration and discrimination.

8 Conclusion

We introduced the first *factor-graph-based* calibration framework for multi-step LLM reasoning and coupled it with *calibration-guided* RL to improve reliability and end-task performance. By modeling step-level dynamics via SAO decomposition, our method estimates and corrects confidence along the trajectory rather than only at the final answer, and train agents to be confident only when intermediate evidence supports it. Across six QA benchmarks, we consistently improve step-wise and trajectory-level calibration and achieve gains in accuracy, indicating that better uncertainty modeling stabilizes multi-hop reasoning and reduces error propagation.

9 Limitation

Our study has two main limitations. First, the proposed framework is primarily evaluated in the QA-centric agent setting, where reasoning can be naturally decomposed into step-wise trajectories and measured against answer supervision. While the factor-graph formulation is general, its current instantiation and empirical validation may not directly transfer to other agentic scenarios such as long-horizon planning, open-ended dialogue, or complex tool-use with heterogeneous observations. Second, although we show that calibration-guided RL improves confidence–accuracy alignment, we do not fully explore how calibrated confidence can be leveraged at inference time for more efficient and robust *test-time scaling*. In particular, future work could use calibrated uncertainty to drive adaptive compute allocation, such as selectively expanding search depth, revising intermediate steps, reranking candidate trajectories, or early-stopping when confidence is sufficiently supported by evidence.

References

Josh Achiam, Steven Adler, Sandhini Agarwal, Lama Ahmad, Ilge Akkaya, Florencia Leoni Aleman, Diogo Almeida, Janko Altenschmidt, Sam Altman, Shyamal Anadkat, and 1 others. 2023. Gpt-4 technical report. *arXiv preprint arXiv:2303.08774*.

Victor Barres, Honghua Dong, Soham Ray, Xujie Si, and Karthik Narasimhan. 2025. Bench: Evaluating conversational agents in a dual-control environment. *arXiv preprint arXiv:2506.07982*.

Evan Becker and Stefano Soatto. Measuring llm confidence through stable explanations.

Haotian Chen, Zijun Song, Boye Niu, Ke Zhang, Litu Ou, Yaxi Lu, Zhong Zhang, Xin Cong, Yankai Lin, Zhiyuan Liu, and 1 others. 2025. Toleap: Rethinking development of tool learning with large language models. *arXiv preprint arXiv:2505.11833*.

Yangyi Chen, Lifan Yuan, Ganqu Cui, Zhiyuan Liu, and Heng Ji. 2022. A close look into the calibration of pre-trained language models. *arXiv preprint arXiv:2211.00151*.

Prateek Chhikara. 2025. Mind the confidence gap: Overconfidence, calibration, and distractor effects in large language models. *arXiv preprint arXiv:2502.11028*.

Daya Guo DeepSeek-AI, Dejian Yang, Haowei Zhang, Junxiao Song, Ruoyu Zhang, Runxin Xu, and 1 others. 2025. Deepseek-r1: Incentivizing reasoning capability in llms via reinforcement learning." arxiv. *Preprint posted online on*, 22:13–14.

Jinhao Duan, Hao Cheng, Shiqi Wang, Alex Zavalny, Chenan Wang, Renjing Xu, Bhavya Kailkhura, and Kaidi Xu. 2024. Shifting attention to relevance: Towards the predictive uncertainty quantification of free-form large language models. In *Proceedings of the 62nd Annual Meeting of the Association for Computational Linguistics (Volume 1: Long Papers)*, pages 5050–5063.

Jinhao Duan, James Diffenderfer, Sandeep Madireddy, Tianlong Chen, Bhavya Kailkhura, and Kaidi Xu. 2025. Uprop: Investigating the uncertainty propagation of llms in multi-step agentic decision-making. *arXiv preprint arXiv:2506.17419*.

Yichao Fu, Xuewei Wang, Yuandong Tian, and Jiawei Zhao. 2025. Deep think with confidence. *arXiv preprint arXiv:2508.15260*.

Tobias Groot and Matias Valdenegro-Toro. 2024. Overconfidence is key: Verbalized uncertainty evaluation in large language and vision-language models. *arXiv preprint arXiv:2405.02917*.

Chuan Guo, Geoff Pleiss, Yu Sun, and Kilian Q Weinberger. 2017. On calibration of modern neural networks. In *International conference on machine learning*, pages 1321–1330. PMLR.

Daya Guo, Dejian Yang, Haowei Zhang, Junxiao Song, Ruoyu Zhang, Runxin Xu, Qihao Zhu, Shirong Ma, Peiyi Wang, Xiao Bi, and 1 others. 2025. Deepseek-r1: Incentivizing reasoning capability in llms via reinforcement learning. *arXiv preprint arXiv:2501.12948*.

Jiuzhou Han, Wray Buntine, and Ehsan Shareghi. 2024. Towards uncertainty-aware language agent. *arXiv preprint arXiv:2401.14016*.

Dan Hendrycks and Kevin Gimpel. 2016. A baseline for detecting misclassified and out-of-distribution examples in neural networks. *arXiv preprint arXiv:1610.02136*.

Xanh Ho, Anh-Khoa Duong Nguyen, Saku Sugawara, and Akiko Aizawa. 2020. Constructing a multi-hop qa dataset for comprehensive evaluation of reasoning steps. *arXiv preprint arXiv:2011.01060*.

Alexander T Ihler, John W Fisher III, Alan S Willsky, and David Maxwell Chickering. 2005. Loopy belief propagation: convergence and effects of message errors. *Journal of Machine Learning Research*, 6(5).

Carlos E Jimenez, John Yang, Alexander Wettig, Shunyu Yao, Kexin Pei, Ofir Press, and Karthik Narasimhan. 2023. Swe-bench: Can language models resolve real-world github issues? *arXiv preprint arXiv:2310.06770*.

Bowen Jin, Hansi Zeng, Zhenrui Yue, Jinsung Yoon, Sercan Arik, Dong Wang, Hamed Zamani, and Jiawei Han. 2025. Search-r1: Training llms to reason and leverage search engines with reinforcement learning. *arXiv preprint arXiv:2503.09516*.

Mandar Joshi, Eunsol Choi, Daniel S Weld, and Luke Zettlemoyer. 2017. Triviaqa: A large scale distantly supervised challenge dataset for reading comprehension. *arXiv preprint arXiv:1705.03551*.

Adam Tauman Kalai, Ofir Nachum, Santosh S Vempala, and Edwin Zhang. 2025. Why language models hallucinate. *arXiv preprint arXiv:2509.04664*.

Sanyam Kapoor, Nate Gruver, Manley Roberts, Katie Collins, Arka Pal, Umang Bhatt, Adrian Weller, Samuel Dooley, Micah Goldblum, and Andrew G Wilson. 2024. Large language models must be taught to know what they don't know. *Advances in Neural Information Processing Systems*, 37:85932–85972.

Vladimir Karpukhin, Barlas Oguz, Sewon Min, Patrick SH Lewis, Ledell Wu, Sergey Edunov, Danqi Chen, and Wen-tau Yih. 2020. Dense passage retrieval for open-domain question answering. In *EMNLP (1)*, pages 6769–6781.

Jannik Kossen, Jiatong Han, Muhammed Razzak, Lisa Schut, Shreshth Malik, and Yarin Gal. 2024. Semantic entropy probes: Robust and cheap hallucination detection in llms. *arXiv preprint arXiv:2406.15927*.

Maia Kotelanski, Robert Gallo, Ashwin Nayak, and Thomas Savage. 2023. Methods to estimate large language model confidence. *arXiv preprint arXiv:2312.03733*.

Frank R Kschischang, Brendan J Frey, and H-A Loeliger. 2002. Factor graphs and the sum-product algorithm. *IEEE Transactions on information theory*, 47(2):498–519.

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

Tom Kwiatkowski, Jennimaria Palomaki, Olivia Redfield, Michael Collins, Ankur Parikh, Chris Alberti, Danielle Epstein, Illia Polosukhin, Jacob Devlin, Kenton Lee, and 1 others. 2019. Natural questions: a benchmark for question answering research. *Transactions of the Association for Computational Linguistics*, 7:453–466.

Woosuk Kwon, Zhuohan Li, Siyuan Zhuang, Ying Sheng, Lianmin Zheng, Cody Hao Yu, Joseph Gonzalez, Hao Zhang, and Ion Stoica. 2023. Efficient memory management for large language model serving with pagedattention. In *Proceedings of the 29th symposium on operating systems principles*, pages 611–626.

Yukun Li, Sijia Wang, Lifu Huang, and Li-Ping Liu. 2024. Graph-based confidence calibration for large language models. *arXiv preprint arXiv:2411.02454*.

Xinbin Liang, Jinyu Xiang, Zhaoyang Yu, Jiayi Zhang, Sirui Hong, Sheng Fan, and Xiao Tang. 2025. Openmanus: An open-source framework for building general ai agents.

Xi Victoria Lin, Xilun Chen, Mingda Chen, Weijia Shi, Maria Lomeli, Richard James, Pedro Rodriguez, Jacob Kahn, Gergely Szilvasy, Mike Lewis, and 1 others. 2023. Ra-dit: Retrieval-augmented dual instruction tuning. In *The Twelfth International Conference on Learning Representations*.

Hao Liu, Zi-Yi Dou, Yixin Wang, Nanyun Peng, and Yisong Yue. 2024. Uncertainty calibration for tool-using language agents. In *Findings of the Association for Computational Linguistics: EMNLP 2024*, pages 16781–16805.

H-A Loeliger. 2004. An introduction to factor graphs. *IEEE Signal Processing Magazine*, 21(1):28–41.

Andrey Malinin and Mark Gales. 2020. Uncertainty estimation in autoregressive structured prediction. *arXiv preprint arXiv:2002.07650*.

Alex Mallen, Akari Asai, Victor Zhong, Rajarshi Das, Daniel Khashabi, and Hannaneh Hajishirzi. 2022. When not to trust language models: Investigating effectiveness of parametric and non-parametric memories. *arXiv preprint arXiv:2212.10511*.

Kevin Murphy, Yair Weiss, and Michael I Jordan. 2013. Loopy belief propagation for approximate inference: An empirical study. *arXiv preprint arXiv:1301.6725*.

Shiyu Ni, Keping Bi, Lulu Yu, and Jiafeng Guo. 2024. Are large language models more honest in their probabilistic or verbalized confidence? In *China Conference on Information Retrieval*, pages 124–135. Springer.

Alexander Nikitin, Jannik Kossen, Yarin Gal, and Pekka Marttinen. 2024. Kernel language entropy: Fine-grained uncertainty quantification for llms from semantic similarities. *Advances in Neural Information Processing Systems*, 37:8901–8929.

Litu Ou, Kuan Li, Huifeng Yin, Liwen Zhang, Zhongwang Zhang, Xixi Wu, Rui Ye, Zile Qiao, Pengjun Xie, Jingren Zhou, and 1 others. 2025. Browseconf: Confidence-guided test-time scaling for web agents. *arXiv preprint arXiv:2510.23458*.

Long Ouyang, Jeffrey Wu, Xu Jiang, Diogo Almeida, Carroll Wainwright, Pamela Mishkin, Chong Zhang, Sandhini Agarwal, Katarina Slama, Alex Ray, and 1 others. 2022. Training language models to follow instructions with human feedback. *Advances in neural information processing systems*, 35:27730–27744.

Shishir G Patil, Huanzhi Mao, Fanjia Yan, Charlie Cheng-Jie Ji, Vishnu Suresh, Ion Stoica, and Joseph E Gonzalez. The berkeley function calling leaderboard (bfcl): From tool use to agentic evaluation of large language models. In *Forty-second International Conference on Machine Learning*.

Long Phan, Alice Gatti, Ziwen Han, Nathaniel Li, Josephina Hu, Hugh Zhang, Chen Bo Calvin Zhang, Mohamed Shaaban, John Ling, Sean Shi, and 1 others. 2025. Humanity's last exam. *arXiv preprint arXiv:2501.14249*.

Ofir Press, Muru Zhang, Sewon Min, Ludwig Schmidt, Noah A Smith, and Mike Lewis. 2022. Measuring and narrowing the compositionality gap in language models. *arXiv preprint arXiv:2210.03350*.

Shuofei Qiao, Runnan Fang, Ningyu Zhang, Yuqi Zhu, Xiang Chen, Shumin Deng, Yong Jiang, Pengjun Xie, Fei Huang, and Huajun Chen. 2024. Agent planning with world knowledge model. *CoRR*, abs/2405.14205.

Rafael Rafailov, Archit Sharma, Eric Mitchell, Christopher D Manning, Stefano Ermon, and Chelsea Finn. 2023. Direct preference optimization: Your language model is secretly a reward model. *Advances in neural information processing systems*, 36:53728–53741.

Zhihong Shao, Peiyi Wang, Qihao Zhu, Runxin Xu, Junxiao Song, Xiao Bi, Haowei Zhang, Mingchuan Zhang, YK Li, Yang Wu, and 1 others. 2024. Deepseekmath: Pushing the limits of mathematical reasoning in open language models. *arXiv preprint arXiv:2402.03300*.

Yifan Song, Da Yin, Xiang Yue, Jie Huang, Sujian Li, and Bill Yuchen Lin. 2024. Trial and error: Exploration-based trajectory optimization for llm agents. *arXiv preprint arXiv:2403.02502*.

Shuchang Tao, Liuyi Yao, Hanxing Ding, Yuexiang Xie, Qi Cao, Fei Sun, Jinyang Gao, Huawei Shen, and Bolin Ding. 2024. When to trust llms: Aligning confidence with response quality. *arXiv preprint arXiv:2404.17287*.

Qwen Team and 1 others. 2024. Qwen2 technical report. *arXiv preprint arXiv:2407.10671*, 2(3).

Cheng Wang, Gyuri Szarvas, Georges Balazs, Pavel Danchenko, and Patrick Ernst. 2024. Calibrating verbalized probabilities for large language models. *arXiv preprint arXiv:2410.06707*.

Liang Wang, Nan Yang, Xiaolong Huang, Binxing Jiao, Linjun Yang, Daxin Jiang, Rangan Majumder, and Furu Wei. 2022. Text embeddings by weakly-supervised contrastive pre-training. *arXiv preprint arXiv:2212.03533*.

Zhiyuan Wang, Jinhao Duan, Chenxi Yuan, Qingyu Chen, Tianlong Chen, Yue Zhang, Ren Wang, Xiaoshuang Shi, and Kaidi Xu. 2025. Word-sequence entropy: Towards uncertainty estimation in free-form medical question answering applications and beyond. *Engineering Applications of Artificial Intelligence*, 139:109553.

Jason Wei, Zhiqing Sun, Spencer Papay, Scott McKinney, Jeffrey Han, Isa Fulford, Hyung Won Chung, Alex Tachard Passos, William Fedus, and Amelia Glaese. 2025. Browsecomp: A simple yet challenging benchmark for browsing agents. *arXiv preprint arXiv:2504.12516*.

Jialong Wu, Wenbiao Yin, Yong Jiang, Zhenglin Wang, Zekun Xi, Runnan Fang, Linhai Zhang, Yulan He, Deyu Zhou, Pengjun Xie, and 1 others. 2025. Webwalker: Benchmarking llms in web traversal. *arXiv preprint arXiv:2501.07572*.

Miao Xiong, Zhiyuan Hu, Xinyang Lu, Yifei Li, Jie Fu, Junxian He, and Bryan Hooi. 2023. Can llms express their uncertainty? an empirical evaluation of confidence elicitation in llms. *arXiv preprint arXiv:2306.13063*.

Duygu Nur Yaldiz, Yavuz Faruk Bakman, Baturalp Buyukates, Chenyang Tao, Anil Ramakrishna, Dimitrios Dimitriadis, Jieyu Zhao, and Salman Avestimehr. 2025. Do not design, learn: A trainable scoring function for uncertainty estimation in generative llms. In *Findings of the Association for Computational Linguistics: NAACL 2025*, pages 691–713.

An Yang, Anfeng Li, Baosong Yang, Beichen Zhang, Binyuan Hui, Bo Zheng, Bowen Yu, Chang Gao, Chengen Huang, Chenxu Lv, and 1 others. 2025. Qwen3 technical report. *arXiv preprint arXiv:2505.09388*.

Zhilin Yang, Peng Qi, Saizheng Zhang, Yoshua Bengio, William W Cohen, Ruslan Salakhutdinov, and Christopher D Manning. 2018. Hotpotqa: A dataset for diverse, explainable multi-hop question answering. *arXiv preprint arXiv:1809.09600*.

Shunyu Yao, Noah Shinn, Pedram Razavi, and Karthik Narasimhan. 2024. bench: A benchmark for tool-agent-user interaction in real-world domains. *arXiv preprint arXiv:2406.12045*.

Shunyu Yao, Jeffrey Zhao, Dian Yu, Nan Du, Izhak Shafran, Karthik R Narasimhan, and Yuan Cao. 2022. React: Synergizing reasoning and acting in language models. In *The eleventh international conference on learning representations*.

Mozhi Zhang, Mianqiu Huang, Rundong Shi, Linsen Guo, Chong Peng, Peng Yan, Yaqian Zhou, and Xipeng Qiu. 2024. Calibrating the confidence of large language models by eliciting fidelity. *arXiv preprint arXiv:2404.02655*.

Zhaohan Zhang, Ziquan Liu, and Ioannis Patras. 2025. Grace: A generative approach to better confidence elicitation in large language models. *arXiv preprint arXiv:2509.09438*.

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. 2025a. Uncertainty propagation on LLM agent. In *Proceedings of the 63rd Annual Meeting of the Association for Computational Linguistics (Volume 1: Long Papers)*, pages 6064–6073, Vienna, Austria. Association for Computational Linguistics.

Qiwei Zhao, Dong Li, Yanchi Liu, Wei Cheng, Yiyou Sun, Mika Oishi, Takao Osaki, Katsushi Matsuda, Huaxiu Yao, Chen Zhao, and 1 others. 2025b. Uncertainty propagation on llm agent. In *Proceedings of the 63rd Annual Meeting of the Association for Computational Linguistics (Volume 1: Long Papers)*, pages 6064–6073.

Qiwei Zhao, Xujiang Zhao, Yanchi Liu, Wei Cheng, Yiyou Sun, Mika Oishi, Takao Osaki, Katsushi Matsuda, Huaxiu Yao, and Haifeng Chen. 2024. Saup: Situation awareness uncertainty propagation on llm agent. *arXiv preprint arXiv:2412.01033*.

A Appendix

A.1 Related Work

A.1.1 Agent Uncertainty Estimation

Traditional uncertainty estimation methods such as entropy, semantic entropy, and sampling variance operate at the token or answer level and therefore fail to capture how uncertainty accumulates across intermediate steps in multi-hop reasoning. Recent work has begun to move toward *agent-level* uncertainty that accounts for the full reasoning trajectory. UALA (Han et al., 2024) integrates uncertainty into the ReAct reasoning loop by using uncertainty scores to decide when to perform tool retrieval or escalate to human supervision. While effective in reducing unnecessary tool calls, its uncertainty estimates remain largely answer-level and do not explicitly track uncertainty flow across steps. SAUP (Zhao et al., 2025b) aggregates step-level uncertainty using situation-aware weights that reflect contextual importance and reasoning progress throughout the agent trajectory. This yields stronger AUROC than final-step uncertainty but relies on heuristic weighting without a formal probabilistic model of information propagation. UProp (Duan et al., 2025) provides an information-theoretic decomposition that separates intrinsic uncertainty from extrinsic uncertainty arising from earlier decisions. By approximating conditional mutual information using trajectory-level PMI, it captures how uncertainty propagates through multi-step decisions.

A.1.2 RL with a Search Engine

We formulate the RL objective utilizing a search engine $\mathcal{R}$ following Search-R1 (Jin et al., 2025):

$$\max_{\pi_\theta} \mathbb{E}_{x\sim\mathcal{D},\, y\sim\pi_\theta(\cdot|x;\mathcal{R})}\left[r_\phi(x,y)\right] - \beta\, D_{\mathrm{KL}}(\pi_\theta(y \mid x;\mathcal{R}) \,\|\, \pi_{\mathrm{ref}}(y \mid x;\mathcal{R})) \tag{14}$$

Here π_θ denotes the policy LLM, π_{ref} is the reference LLM, r_ϕ is the reward function, and D_{KL} is the KL divergence. Input samples x are drawn from the dataset $\mathcal{D}$, and the generated outputs y interleave LLM reasoning with search engine results, both sampled from $\pi_{\mathrm{ref}}(y \mid x)$ and retrieved through $\mathcal{R}$. Unlike prior RL approaches relying solely on the policy LLM (Rafailov et al., 2023; Ouyang et al., 2022), Search-R1 (Jin et al., 2025) explicitly incorporates retrieval-interleaved reasoning via $\pi_\theta(\cdot \mid x;\mathcal{R})$, which can be interpreted as $\pi_\theta(\cdot \mid x) \otimes \mathcal{R}$, where $\otimes$ denotes interleaved retrieval–and–reasoning. Our approach builds upon Group Relative Policy Optimization (GRPO) (Shao et al., 2024; Guo et al., 2025), leveraging their complementary strengths for retrieval-augmented reasoning.

Loss Masking for Retrieved Tokens. In GRPO, losses are computed over the full rollout sequence. However, retrieval-augmented rollouts contain both LLM-generated tokens and tokens copied from retrieved content. Optimizing over retrieved tokens may create undesired learning dynamics, as these tokens do not reflect model reasoning. Therefore, we following Search-R1 (Jin et al., 2025) apply *token masking*, ensuring that only LLM-generated tokens contribute to the policy gradient while retrieved tokens are excluded, stabilizing training without restricting search-augmented generation.

GRPO with Search Engine Group Relative Policy Optimization (Shao et al. 2024) improves stability by removing the need for value-function learning. GRPO samples a group $\{y_1, \ldots, y_G\}$ from the reference model π_{ref} for each input x, and optimizes the policy by maximizing:

$$\begin{aligned}\mathcal{J}_{\mathrm{GRPO}}(\theta) = \mathbb{E}_{x\sim\mathcal{D},\, y_{1:G}\sim\pi_{\mathrm{old}}(\cdot|x;\mathcal{R})}\Bigg[\frac{1}{G}\sum_{j=1}^{G}\frac{1}{\sum_t I(y_{j,t})} \\ \sum_{t:I(y_{j,t})=1}\min\Bigg(\frac{\pi_\theta(y_{j,t} \mid x, y_{j,<t};\mathcal{R})}{\pi_{\mathrm{old}}(y_{j,t} \mid x, y_{j,<t};\mathcal{R})}\hat{A}_{j,t}, \\ \mathrm{clip}\Bigg(\frac{\pi_\theta(y_{j,t} \mid x, y_{j,<t};\mathcal{R})}{\pi_{\mathrm{old}}(y_{j,t} \mid x, y_{j,<t};\mathcal{R})}, 1-\epsilon,\ 1+\epsilon\Bigg)\hat{A}_{j,t}\Bigg) \\ -\beta\, D_{\mathrm{KL}}[\pi_\theta \,\|\, \pi_{\mathrm{ref}}]\Bigg]\end{aligned} \tag{15}$$

Here ϵ and β are hyperparameters. Group-based advantages $\hat{A}_{j,t}$ are computed using the relative reward of responses within each group. Unlike PPO, GRPO directly penalizes the KL divergence using a loss term, while still applying retrieved-token masking during the KL computation.

A.1.3 LLM Calibration

Logit-based methods Logit-based methods are the most widely used and effective approaches in uncertainty estimation. As a foundational method, Predictive Entropy (PE) (Malinin and Gales, 2020), defines total uncertainty as the entropy of the output logits distribution. After that, researchers proposed a series of methods based on the inherent characteristics of natural language generation to improve upon PE methods. (Kuhn et al., 2023) introduced semantic entropy (SE) that estimates uncertainty by marginalizing over semantically equivalent samples in NLG tasks. In the similar framework, (Nikitin et al., 2024) employed positive semi-definite kernels and von Neumann entropy to capture semantic similarities. Furthermore, (Wang et al., 2025) proposed Word-Sequence Entropy (WSE) to adjust uncertainty proportions at both the word and sequence levels based on semantic relevance, ensuring that uncertainty is aligned with the semantic importance of words within a response. In addition to measuring the similarity between generated responses. (Duan et al., 2024) proposed Shifting Attention to Relevance (SAR), which focus on relevant components and assigns significance weights to tokens based on their contributions to the overall response. Unlike these carefully designed methods, (Yaldiz et al., 2025) introduced a Learnable Response Scoring Function (LARS), which utilizes supervised data to capture complex token-probability dependencies. While effective, the above methods are computationally expensive. To alleviate these computational cost, (Kossen et al., 2024) proposed Semantic Entropy Probes (SEPs) to approximate semantic entropy by leveraging hidden states from a single generation.

Verbal confidence methods Verbal confidence methods Due to LLMs' strong language abilities and adherence to instructions, Verbal confidence methods are proposed. For instance, one may attach the question with a prompt like "Please respond and provide your confidence score ranging from 0 to 100.". (Xiong et al., 2023) constructed a prompting, sampling, and aggregation framework to systematically evaluate various strategies and their integration, enabling LLMs to express their confidence in response. (Groot and Valdenegro-Toro, 2024) proposed FaR prompting strategy, which improves the confidence calibration of LLMs by separating the fact retrieval and reflective reasoning steps. However, verbal confidence methods face significant challenges with over-confidence. (Ni et al., 2024) found that LLMs cannot convey their uncertainties faithfully in natural language. (Becker and Soatto) found that combining language confidence and proxy model probability estimation can improve the estimation of uncertainty. (Joshi et al., 2017) noted LLMs' language perception accuracy often lags behind probability perception, especially in specific domains Furthermore, (Tao et al., 2024) found that LLMs often exhibit a high degree of overconfidence when expressing their own confidence by comparing language-based methods, consistency-based methods, and their hybrid benchmark testing methods. Their research indicates that some prompt strategies can improve the calibration of verbal confidence.

A.2 Preliminary For Factor Graph

A.2.1 Forward Belief Propagation

Forward propagation centers on message passing between factor nodes to variable nodes and variable nodes to factor nodes, ultimately calculating the belief of variable nodes.

The belief of a variable node X is computed as the normalized product of incoming messages from its neighboring factor nodes:

$$b^{-}(X) = \frac{\prod_{f \in nb(X)} m_{f \to X}(X)}{\sum_{x' \in \mathcal{X}} \prod_{f \in nb(X)} m_{f \to X}(x')} \quad (16)$$

$nb(X)$ denotes the set of factor nodes connected to X, and each $m_{f \to X}(X)$ represents a message in the form of a probability vector. The numerator aggregates probabilistic evidence from all connected factors, while the denominator normalizes the belief to ensure it sums to one.

The message from a factor node f to a variable node X is obtained by marginalizing over all variables except X:

$$m_{f \to X}(x) = \sum_{v \setminus x} f(v) \prod_{Y \in nb(f) \setminus X} m_{Y \to f}(y) \quad (17)$$

$f(v)$ denotes the factor function (or conditional probability table) over all variables connected to f, and $m_{Y \to f}(y)$ represents the message sent from variable node Y to factor node f. The summation $\sum_{v \setminus x}$ marginalizes over all variables except X, allowing the message to reflect the combined influence of neighboring variables on X.

A.2.2 Backward Belief Propagation

Formulas for Backward Belief Propagation Backward propagation calculates the posterior belief by fusing forward messages (α) and backward messages (β), with the core being backward message passing and normalization.

The posterior belief of a variable node X is the normalized product of forward messages and backward messages, and the formula is as follows:

$$b^{+}(X) = \frac{\alpha_X(x) \cdot \beta_X(x)}{\sum_{x' \in \mathcal{X}} \alpha_X(x') \cdot \beta_X(x')} \tag{18}$$

In this formulation, $\alpha_X(x)$ represents the forward message, obtained as the product of probabilistic information propagated to X from its left-side factor nodes during the forward pass, while $\beta_X(x)$ denotes the backward message, carrying information from the right-side factors in the backward pass. The denominator in the equation serves as a normalization term, ensuring that the posterior beliefs over all possible values of X sum to one.

Formula for Backward Messages from Factor Nodes to Variable Nodes The backward message sent from a factor node f to a variable node X follows the same logic as that in forward propagation and requires marginalization over all variable values except X. The formula is as follows:

$$m^{(bwd)}_{f \to X}(x) = \sum_{v \setminus x} f(v) \prod_{Y \in nb(f) \setminus X} m^{(bwd)}_{Y \to f}(y) \tag{19}$$

The superscript bwd indicates that the message is propagated in the backward direction, while the meanings of all other symbols remain consistent with those in the forward message formulation.

A.3 CPT Related

A.3.1 CPT Data Detail

We construct the CPT training dataset from the *HotpotQA* (Yang et al., 2018) training corpus by randomly sampling a subset of 2000 examples. Using the `DeepSeek-R1` (DeepSeek-AI et al., 2025) model as the backend agent within the `Manus` framework (Liang et al., 2025) , we generate multi-step reasoning trajectories for each instance. Among the generated trajectories, 1,492 lead to correct final answers, while 508 result in incorrect outcomes. From these trajectories, we extract **Observation–State (OS)** and **State–Action (SA)** pairs for Conditional Probability Table (CPT) training. A subset of these pairs is annotated by a strong LLM using the evaluation prompts described in A.3.2. To ensure reliability, we filter out inconsistent annotations based on agreement between gold labels and LLM-assigned scores. The full data pair-level statistics at each stage are summarized in Table 4.

Stage	OS Pairs	SA Pairs
Extracted from Trajectories	40,234	41,564
LLM-Scored Subset	6,000	6,000
Filtered ($\lvert$Score–Label$\rvert < 0.3$)	3,273	3,264
Held-out for Evaluation	2,727	2,736

Table 4: Summary of the CPT data construction process based on the *HotpotQA* dataset.

A.3.2 CPT Data Construction Prompt

To construct training data for the Conditional Probability Tables (CPT) used in our factor-graph calibration model, we design two types of evaluation prompts that elicit scalar quality judgments from a strong LLM. (1) *Action-level* evaluation assesses whether a proposed action is appropriate given the current reasoning state, and (2) *State-level* evaluation measures whether the generated intermediate reasoning step (state) logically advances toward the final goal.

Action-level Evaluation Prompt. The following prompt evaluates the quality and suitability of an action proposed by the model, given the current reasoning state and the task goal.

Action_Quality_Judge_Prompt

<USER>
You are a strict and consistent judge. Score the suitability of the PROPOSED ACTION given the current STATE to achieve the GOAL.
Rubric (weights):
Relevance to goal (0.35)
Progress/utility toward next step (0.25)
Feasibility and tool/constraint fit (0.20)

Scoring guide (integer 0–10):
0–2: off-topic/unsafe/infeasible;
3–5: weak/low utility;
6–7: reasonable;
8–9: strong;
10: near-optimal.
If key context is missing or constraints are violated, score low.

[GOAL]
`{QUESTION}`
[CONTEXT]
`{CONTEXT}`

Return **ONLY one line:**
`<judge> YOUR SCORE </judge>`
(Where SCORE is an integer 0–10; no other text.)

State-level Evaluation Prompt. This prompt judges the quality of a candidate reasoning state (<think>) based on its logical consistency, evidential grounding, and contribution to task progress. It helps build the conditional factor $\phi_{O_{t-1}S_t}$ in the CPT by quantifying how well each intermediate state aligns with its preceding observation.

State_Quality_Judge_Prompt

<USER>
You are a strict and consistent judge. Score the quality of the CANDIDATE NEXT THINK/STATE given the OBSERVATION (and context) for advancing the GOAL.
Rubric (weights):
Evidence fit to observation (0.40)
Logical consistency with prior state/action (0.20)
Direction/correctness toward the goal (0.20)

Scoring guide (integer 0–10):
0–2: contradicts or ignores OBS;
3–5: generic/low progress;
6–7: reasonable;
8–9: strong evidence-based step;
10: excellent.
Penalize contradictions, hallucinations, or lack of progress.

[GOAL]
`{QUESTION}`
[PRIOR STATE]
`{CONTEXT}`

Return **ONLY one line:**
`<judge> YOUR SCORE </judge>`
(Where YOUR SCORE is an integer 0–10; no other text.)

A.3.3 CPT Evaluation Protocol

To evaluate the calibration quality of the proposed CPT model, we compute five probability-level metrics that quantify the discrepancy between the predicted probability $\hat{p} = P_\theta(y{=}1 \mid x)$ and the reference confidence p (LLM-judge score). We report: L1 difference (L1), Mean Squared Error (MSE), Soft Cross-Entropy (Soft-CE), Kullback–Leibler Divergence (KL-Div), and Jensen–Shannon Divergence (JS-Div), defined as follows:

L1 difference (L1):

$$\mathrm{L1} = |\hat{p} - p|$$

Mean Squared Error (MSE):

$$\mathrm{MSE} = (\hat{p} - p)^2$$

Soft Cross-Entropy (Soft-CE):

$$\mathrm{SoftCE} = -\big[p \log \hat{p} + (1-p)\log(1-\hat{p})\big]$$

Kullback–Leibler Divergence (KL-Div):

$$\mathrm{KL}(p\|\hat{p}) = p \log \frac{p}{\hat{p}} + (1-p)\log \frac{1-p}{1-\hat{p}}$$

Jensen–Shannon Divergence (JS-Div):

$$\mathrm{JS}(p\|\hat{p}) = \tfrac{1}{2}\mathrm{KL}(p\|m) + \tfrac{1}{2}\mathrm{KL}(\hat{p}\|m)$$
$$m = \tfrac{1}{2}(p + \hat{p})$$

These metrics collectively assess how well the CPT model approximates the underlying probabilistic relationship between reasoning components and correctness signals, with lower values indicating better calibration.

Model	Metric	Before Calibration		After Calibration	
		Mean ↓	Std ↓	Mean ↓	Std ↓
Qwen2.5-7B-Inst	L1 diff	0.1692	0.1934	0.1926	0.1448
	MSE diff	0.0614	0.0864	0.0581	0.0799
	Soft-CE	0.9125	0.9541	0.5017	0.3012
	KL div	0.5785	0.9465	0.1682	0.2039
	JS div	0.1145	0.1695	0.0477	0.0544
Qwen3-4B-Inst	L1 diff	0.1026	0.1053	0.1063	0.1131
	MSE diff	0.0216	0.0580	0.0234	0.0596
	Soft-CE	0.7928	0.7637	0.4117	0.3060
	KL div	0.4587	0.7748	0.0782	0.1439
	JS div	0.1025	0.1557	0.0221	0.0371
Qwen3-4B-Base	L1 diff	0.1668	0.1366	0.1399	0.1269
	MSE diff	0.0465	0.0578	0.0357	0.0530
	Soft-CE	0.7277	0.6223	0.4436	0.3014
	KL div	0.3493	0.6309	0.1102	0.1603
	JS div	0.0953	0.1334	0.0308	0.0454
Qwen3-1.7B-Base	L1 diff	0.1142	0.1164	0.1164	0.1371
	MSE diff	0.0266	0.0555	0.0273	0.0580
	Soft-CE	0.7963	0.8811	0.4211	0.2919
	KL div	0.4182	0.6970	0.0987	0.1291
	JS div	0.1010	0.1568	0.0247	0.0375

Table 5: Cross-model evaluation of CPT calibration performance across five probability-level metrics. Lower values indicate better calibration quality.

As shown in Table 5, the CPT calibration notably reduces divergence-based errors (KL, JS) and cross-entropy loss, demonstrating improved probability alignment between model predictions and reference confidences across both instruction-tuned and base models.

A.4 Step and Trajectory-level Confidence aggregation

In this subsection, we define the confidence estimation pipeline used throughout our framework. Starting from step-wise calibrated marginals produced by factor-graph inference, we construct per-step confidence scores and propose several trajectory-level aggregation strategies.

A.4.1 Step-wise confidence aggregation

At each reasoning step t, our factor graph defines three binary latent variables (S_t, A_t, O_t), indicating whether the internal *state*, the executed *action*, and the resulting *observation*. Running belief propagation yields two sets of marginal distributions for each variable: (i) *forward* marginals computed without injecting terminal evidence, denoted as $b_t^-(S_t), b_t^-(A_t), b_t^-(O_t)$, and (ii) *backward/posterior* marginals obtained after injecting terminal correctness evidence and propagating it backward through the graph, denoted as $b_t^+(S_t), b_t^+(A_t), b_t^+(O_t)$. We convert these marginals into a scalar step-wise confidence by averaging the probability assigned to the correct value of each variable:

$$c_t = \frac{b_t(S_t) + b_t(A_t) + b_t(O_t)}{3}.$$

When the probabilities are taken from the forward marginals $b_t^-(\cdot)$, we obtain the *Forward* step confidence c_t^-; when they are taken from the posterior marginals $b_t^+(\cdot)$, we obtain the *Backward/Post* step confidence c_t^+.

A.4.2 Trajectory-level confidence aggregation

Given the confidence sequence $\{c_1, c_2, \ldots, c_T\}$ for a T-step reasoning chain, we compute several trajectory-level confidence metrics. To additionally account for the mismatch between calibrated posteriors and evidence-free forward marginals, we compute a KL-weighted confidence score using the formulation introduced in below:

$$\text{Mean:} \quad c_{\text{mean}} = \frac{1}{T}\sum_{i=1}^{T} c_i$$

$$\text{RMS:} \quad c_{\text{rms}} = \sqrt{\frac{1}{T}\sum_{i=1}^{T} c_i^2}$$

$$\text{GMean:} \quad c_{\text{geo}} = \exp\left(\frac{1}{T}\sum_{i=1}^{T}\log(c_i)\right)$$

$$\text{Max:} \quad c_{\max} = \max_i c_i$$

$$\mathrm{KL}_t = \tfrac{1}{3}\Big[\mathrm{KL}\big(b_t^+(S_t)\,\|\,b_t^0(S_t)\big) + \mathrm{KL}\big(b_t^+(A_t)\,\|\,b_t^0(A_t)\big) + \mathrm{KL}\big(b_t^+(O_t)\,\|\,b_t^0(O_t)\big)\Big]$$

$$\text{KL-Weighted} = \frac{\sum_t \mathrm{KL}_t\, c_t}{\sum_t \mathrm{KL}_t}$$

A.4.3 Verbal confidence aggregation

In addition to these factor-graph–based confidences, the model also outputs a scalar *verbal confidence* in natural language at each step, enclosed in `<confidence> </confidence>` tags. We parse the numerical value from each tag to obtain a sequence $\{v_1, v_2, \ldots, v_T\}$, where $v_i \in [0, 1]$ denotes the self-reported confidence for the i-th action. We aggregate verbal confidence using the same operators as above:

$$\text{Mean:} \quad v_{\text{mean}} = \frac{1}{T}\sum_{i=1}^{T} v_i$$

$$\text{RMS:} \quad v_{\text{rms}} = \sqrt{\frac{1}{T}\sum_{i=1}^{T} v_i^2}$$

$$\text{GMean:} \quad v_{\text{geo}} = \exp\left(\frac{1}{T}\sum_{i=1}^{T}\log(v_i)\right)$$

$$\text{Max:} \quad v_{\max} = \max_i v_i$$

A.5 More Results

A.5.1 Reward Ablation

To disentangle the effects of structural alignment and confidence calibration, we conduct a reward ablation study with two post-training variants that differ only in their reward design.

Ins-Format is trained using a *format-only reward*, which enforces strict adherence to a predefined response structure. In this stage, the model is optimized solely with the format reward defined in Eq. 4, without incorporating any task correctness or confidence calibration signals, the model training dynamics is depicted in Fig 4 left column.

Format-RL is initialized from the converged **Ins-Format** checkpoint and then further optimized by augmenting the objective with an additional *calibration reward*. During this stage, we jointly optimize the format reward and the calibration reward as defined in Eq. 8; the corresponding training dynamics are illustrated in Fig. 4 (right column).

Table 9 and Table 10 demonstrate a clear and consistent distinction between format alignment and confidence calibration. While **Ins-Format**—trained solely with a format-only reward—already yields valid and stable structured outputs, it remains poorly calibrated across all datasets, as evidenced by relatively high ECE scores under all aggregation schemes. In contrast, introducing an explicit calibration reward in

Format-RL leads to substantial and consistent reductions in calibration error (ECE-F/V), accompanied by simultaneous improvements in discriminative ability (AUC-F/V).

These results confirm that format alignment alone is insufficient for reliable confidence estimation. Meaningful calibration emerges only after explicitly optimizing for confidence–accuracy consistency, validating the necessity of calibration-guided reinforcement learning beyond structural post-training.

A.5.2 Additional Calibration Results

Calibration Results For Gmean and Rms Aggregation is shown in Table 1, more results for mean, max, kl-weighted in 6. The Calibration result without retrieval external information is shown in Table 7 and Table 8.

A.5.3 ECE Gain and AUC Gain

ECE Gain and AUC Gain results for Gmean Aggregation method is shown in Fig 3. More ECE Gain and AUC Gain results is shown in Fig 6.

A.5.4 Training Dynamics

The base model and instruct model training dynamics is shown in Fig 5.

A.6 Training Prompt Template

This calibration prompt defines a structured reasoning protocol for training.

Calibration Training Prompt Template

Answer the given question.
You must conduct reasoning inside `<think>` and `</think>` every time you get new information.
If you find that you lack knowledge after reasoning, you may call a search engine using `<search>` query `</search>`, and the system will return results inside `<information>` and `</information>`.
After each search, you must provide your belief score inside `<confidence>` and `</confidence>` (a value between 0 and 1).
Your confidence must reflect how likely your current reasoning or answer is correct, increase only when evidence becomes stronger, and decrease whenever evidence is weak or contradictory.
You may repeat the pattern `<think> <search> <confidence>` up to four times, but on the fourth repetition you **MUST** provide the final answer.
In the fourth round, instead of performing another search, you must output your final answer inside `<answer>` and `</answer>` followed by the final `<confidence>` value.
For example: `<answer>` Beijing `</answer>` `<confidence>` 0.92 `</confidence>`

Question: `{question}`

A.7 Evaluation Prompt Template

Step-level correctness judge The prompt scores the *correctness and usefulness* of each reasoning step against the golden answer.

Step_Judge_Prompt

<USER>
You are a strict and consistent judge for multi-step reasoning agents. Your task is to score the QUALITY of ONE reasoning step in a trajectory for answering a question.
You will be given the question, the golden correct answer (not the agent's answer), the previous trajectory context, and the current step fields: STATE, THINK, ACTION, and OBSERVATION.
Judge this step by: (i) relevance to the question/goal, (ii) coherence with prior context, (iii) correct and useful use of the observation, and (iv) progress toward the golden answer.
Scoring rule (real number in [0,1]): 0.0–0.2 useless/off-topic/wrong; 0.2–0.4 weak; 0.4–0.6 mediocre; 0.6–0.8 good; 0.8–1.0 excellent.
Output format (one line only): `<score>S</score>` where S is a float in [0,1]. No other text.
[QUESTION] `{question}`
[GOLDEN ANSWER] `{golden_answer}`
[PREVIOUS CONTEXT] `{up_context}`
[CURRENT STEP - STATE] `{current_state}`
[CURRENT STEP - THINK] `{current_think}`
[CURRENT STEP - ACTION] `{current_action}`
[CURRENT STEP - OBSERVATION] `{current_observation}`

Now output ONLY: `<score>...</score>`

A.8 Retrieval Settings

For retrieval, we use the 2018 Wikipedia dump (Karpukhin et al., 2020) as the knowledge source and E5 (Wang et al., 2022) as the retriever. To ensure fair comparison, we follow (Lin et al., 2023) and set the number of retrieved passages to 3 across all retrieval-based methods.

A.9 Dataset Overview

As shown in Table 11, we report the train/dev/test splits of our benchmarks and the evaluation subset used for each dataset.

Dataset	#Tr	#Dv	#Te	Eval
NQ	79,168	8,757	3,610	2,000
TriviaQA	78,785	8,837	11,313	2,000
PopQA	–	–	14,267	2,000
HotpotQA	90,447	7,405	–	2,000
2WikiMultiHopQA	15,000	12,576	–	2,000
Bamboogle	–	–	125	125

Table 11: Dataset splits and evaluation subset used in our experiments (2,000 sampled from test/dev when available; otherwise full split).

		KL-Weighted				Max						Mean					
Dataset	**Model**	ECE-F	ECE-B	AUC-F	AUC-B	ECE-F	ECE-B	ECE-V	AUC-F	AUC-B	AUC-V	ECE-F	ECE-B	ECE-V	AUC-F	AUC-B	AUC-V
hotpotqa	3B-Base	0.7144	0.6263	0.5014	0.7890	0.8370	0.8126	0.6216	0.5068	0.5981	0.6505	0.6906	0.6023	0.5115	0.4993	0.7982	0.5173
	Base-RL	0.5500	0.4535	0.6793	0.9239	0.6816	0.6633	0.5503	0.6296	0.6025	0.6870	0.5228	0.4286	0.3749	0.6856	0.9326	0.6383
	3B-Instruct	0.5419	0.4130	0.5016	0.9400	0.6547	0.5850	0.5862	0.5129	0.7138	0.5499	0.5186	0.4109	0.4047	0.4988	0.9434	0.5271
	Ins-RL	0.3361	0.2829	0.6587	0.9281	0.4601	0.4428	0.3879	0.6101	0.6172	0.7108	0.3061	0.2674	0.2617	0.6577	0.9336	0.6733
popqa	3B-Base	0.6695	0.5860	0.4983	0.8306	0.7851	0.7631	0.5658	0.4874	0.6152	0.7159	0.6426	0.5646	0.4317	0.5061	0.8389	0.6373
	Base-RL	0.4480	0.3773	0.6792	0.9328	0.5653	0.5502	0.4168	0.6325	0.6165	0.7483	0.4252	0.3582	0.2303	0.6788	0.9395	0.7076
	3B-Instruct	0.5225	0.4029	0.5937	0.9560	0.6277	0.5847	0.5635	0.5945	0.7778	0.5291	0.4995	0.3941	0.3609	0.5923	0.9568	0.6340
	Ins-RL	0.2992	0.2534	0.6804	0.9352	0.4090	0.3976	0.3331	0.6651	0.6664	0.7375	0.2742	0.2454	0.2155	0.6794	0.9377	0.7334
2wikimultihopqa	3B-Base	0.7401	0.6662	0.5077	0.8167	0.8550	0.8410	0.6306	0.5099	0.6006	0.5991	0.7195	0.6425	0.5150	0.5073	0.8275	0.5529
	Base-RL	0.5075	0.4197	0.5967	0.8596	0.6508	0.6380	0.5016	0.5584	0.5363	0.6767	0.4741	0.4032	0.3534	0.6020	0.8715	0.6465
	3B-Instruct	0.4878	0.4017	0.4972	0.9050	0.5916	0.5485	0.4979	0.5021	0.6459	0.5704	0.4718	0.3892	0.3423	0.4953	0.9116	0.5293
	Ins-RL	0.3166	0.2849	0.6525	0.8832	0.4452	0.4333	0.3730	0.6059	0.5723	0.6317	0.3053	0.2678	0.2506	0.6495	0.8896	0.6429
bamboogle	3B-Base	0.7867	0.6888	0.7213	0.9160	0.9069	0.8925	0.7125	0.7381	0.7717	0.6492	0.7683	0.6657	0.5997	0.7073	0.9314	0.4643
	Base-RL	0.6014	0.5176	0.6659	0.9080	0.7130	0.7078	0.5735	0.6449	0.6039	0.6238	0.5781	0.4891	0.3934	0.6604	0.9139	0.6414
	3B-Instruct	0.5855	0.4900	0.5018	0.9321	0.6806	0.6576	0.6230	0.4252	0.6912	0.6151	0.5547	0.4753	0.4463	0.5129	0.9348	0.6586
	Ins-RL	0.4627	0.3593	0.6955	0.9545	0.5769	0.5648	0.5149	0.6967	0.6778	0.7356	0.4247	0.3451	0.3900	0.6863	0.9576	0.7061
nq	3B-Base	0.6425	0.5567	0.5058	0.8226	0.7510	0.7280	0.5632	0.5399	0.6409	0.7140	0.6241	0.5334	0.4665	0.5043	0.8312	0.6101
	Base-RL	0.4194	0.3602	0.7622	0.9503	0.5115	0.5052	0.3716	0.6835	0.6496	0.7277	0.3914	0.3389	0.1951	0.7641	0.9520	0.6691
	3B-Instruct	0.4971	0.4028	0.5659	0.9423	0.5842	0.5402	0.5139	0.5646	0.7684	0.5822	0.4704	0.3876	0.3798	0.5647	0.9433	0.6058
	Ins-RL	0.2948	0.2547	0.6874	0.9348	0.4062	0.3925	0.3312	0.6450	0.6247	0.7240	0.2675	0.2459	0.2284	0.6907	0.9401	0.7167
triviaqa	3B-Base	0.6031	0.5262	0.4537	0.8077	0.7102	0.6932	0.5190	0.5107	0.6286	0.7315	0.5866	0.5108	0.4306	0.4566	0.8186	0.6654
	Base-RL	0.3729	0.3081	0.6534	0.9237	0.4842	0.4666	0.3482	0.6278	0.6187	0.7279	0.3445	0.2924	0.1968	0.6555	0.9303	0.7003
	3B-Instruct	0.3465	0.2780	0.5199	0.9130	0.4426	0.3887	0.3614	0.4835	0.7214	0.5689	0.3259	0.2773	0.2567	0.5256	0.9166	0.5822
	Ins-RL	0.1401	0.1251	0.7424	0.9544	0.2249	0.2050	0.1666	0.6340	0.6626	0.7512	0.1292	0.1271	0.1121	0.7459	0.9576	0.7389

Table 6: Unified appendix calibration table (KL-weighted, Max, Mean) across models.

Dataset	Model	GMean						RMS					
		ECE-F	ECE-B	ECE-V	AUC-F	AUC-B	AUC-V	ECE-F	ECE-B	ECE-V	AUC-F	AUC-B	AUC-V
hotpotqa	3B-Base	0.6921	0.5324	0.5607	0.5511	0.8914	0.5731	0.7213	0.6486	0.5904	0.5595	0.8621	0.5885
	Base-RL	0.6919	0.5294	0.5723	0.5580	0.9040	0.5729	0.7191	0.6480	0.5967	0.5620	0.8797	0.5765
	3B-Instruct	0.6023	0.4235	0.4914	0.5601	0.9266	0.5457	0.6224	0.5261	0.5329	0.5600	0.9082	0.5381
	Ins-RL	0.6764	0.5350	0.6613	0.4527	0.8839	0.5203	0.7037	0.6365	0.6778	0.4444	0.8357	0.5219
popqa	3B-Base	0.6929	0.5278	0.5053	0.5341	0.8915	0.6196	0.7235	0.6483	0.5495	0.5295	0.8435	0.6346
	Base-RL	0.6773	0.5094	0.4940	0.5991	0.9312	0.6070	0.7043	0.6319	0.5423	0.6007	0.9116	0.6161
	3B-Instruct	0.6635	0.4760	0.4406	0.5228	0.9405	0.6867	0.6810	0.5823	0.5249	0.5175	0.9297	0.6445
	Ins-RL	0.7051	0.5621	0.6225	0.5311	0.9232	0.6166	0.7404	0.6775	0.6673	0.5177	0.8844	0.6180
2wikimultihopqa	3B-Base	0.6320	0.4912	0.5075	0.4910	0.8764	0.5632	0.6564	0.5992	0.5461	0.4876	0.8131	0.5758
	Base-RL	0.6190	0.4907	0.5132	0.4737	0.8809	0.5517	0.6480	0.5877	0.5435	0.4820	0.8391	0.5620
	3B-Instruct	0.5372	0.4208	0.4033	0.4471	0.8853	0.5543	0.5610	0.4961	0.4396	0.4440	0.8529	0.5561
	Ins-RL	0.5833	0.4867	0.5205	0.5041	0.8804	0.5977	0.5998	0.5650	0.5381	0.5021	0.8366	0.6002
bamboogle	3B-Base	0.7089	0.5686	0.5613	0.5894	0.8887	0.5351	0.7279	0.6695	0.6013	0.5830	0.8475	0.5235
	Base-RL	0.7264	0.5705	0.5865	0.4206	0.9180	0.4940	0.7337	0.6690	0.5894	0.4242	0.8977	0.5335
	3B-Instruct	0.6266	0.4399	0.5178	0.6052	0.9292	0.5056	0.6393	0.5497	0.5691	0.5916	0.9172	0.4660
	Ins-RL	0.6346	0.5022	0.6503	0.5380	0.9109	0.6370	0.6706	0.5987	0.6786	0.5322	0.8879	0.6194
nq	3B-Base	0.6523	0.5112	0.5186	0.5846	0.8773	0.6126	0.6792	0.6045	0.5448	0.5844	0.8384	0.6231
	Base-RL	0.6471	0.5327	0.4377	0.5092	0.8907	0.5675	0.6606	0.6205	0.4864	0.5109	0.8666	0.5698
	3B-Instruct	0.5500	0.4006	0.4701	0.5693	0.9349	0.5391	0.5773	0.5036	0.5025	0.5638	0.9161	0.5315
	Ins-RL	0.6436	0.5189	0.5915	0.5556	0.8973	0.5910	0.6657	0.6102	0.6180	0.5506	0.8739	0.5876
triviaqa	3B-Base	0.5893	0.4725	0.4782	0.5690	0.8945	0.6451	0.6159	0.5589	0.4978	0.5665	0.8666	0.6585
	Base-RL	0.5343	0.4329	0.4068	0.5022	0.8992	0.5309	0.5470	0.5109	0.4356	0.5002	0.8706	0.5553
	3B-Instruct	0.4441	0.3171	0.3495	0.5402	0.9296	0.5280	0.4520	0.3905	0.3736	0.5294	0.9159	0.5297
	Ins-RL	0.4609	0.3623	0.4703	0.5547	0.9275	0.4455	0.4787	0.4390	0.4651	0.5478	0.9092	0.4538

Table 7: Unified calibration table (GMean + RMS) across models without external information.

		KL-Weighted				Max						Mean					
Dataset	**Model**	ECE-F	ECE-B	AUC-F	AUC-B	ECE-F	ECE-B	ECE-V	AUC-F	AUC-B	AUC-V	ECE-F	ECE-B	ECE-V	AUC-F	AUC-B	AUC-V
hotpotqa	3B-Base	0.7310	0.6208	0.5573	0.8708	0.8644	0.8460	0.7127	0.5194	0.6384	0.6237	0.7072	0.6000	0.5768	0.5544	0.8784	0.5823
	Base-RL	0.7287	0.6229	0.5640	0.8871	0.8476	0.8300	0.7437	0.5352	0.6157	0.5565	0.7046	0.6000	0.5909	0.5614	0.8933	0.5750
	3B-Instruct	0.6300	0.4956	0.5608	0.9149	0.7378	0.6828	0.6702	0.5190	0.6961	0.5364	0.6115	0.4847	0.5118	0.5598	0.9204	0.5384
	Ins-RL	0.7128	0.6160	0.4440	0.8504	0.8239	0.8050	0.7650	0.4660	0.5641	0.5321	0.6862	0.5962	0.6678	0.4485	0.8635	0.5216
popqa	3B-Base	0.7333	0.6210	0.5371	0.8571	0.8600	0.8353	0.6902	0.5464	0.6250	0.6303	0.7088	0.6037	0.5246	0.5310	0.8681	0.6286
	Base-RL	0.7155	0.6047	0.6044	0.9183	0.8351	0.8147	0.7059	0.5762	0.6747	0.6051	0.6881	0.5819	0.5192	0.5999	0.9238	0.6077
	3B-Instruct	0.6890	0.5520	0.5208	0.9335	0.7985	0.7504	0.6844	0.5179	0.6858	0.5777	0.6706	0.5395	0.4828	0.5193	0.9372	0.6677
	Ins-RL	0.7510	0.6561	0.5195	0.8972	0.8642	0.8545	0.7833	0.5238	0.6125	0.6092	0.7223	0.6331	0.6504	0.5240	0.9067	0.6179
2wikimultihopqa	3B-Base	0.6665	0.5747	0.4918	0.8342	0.7948	0.7779	0.6772	0.4923	0.5790	0.5961	0.6427	0.5544	0.5327	0.4878	0.8457	0.5694
	Base-RL	0.6570	0.5680	0.4798	0.8538	0.7790	0.7606	0.6692	0.4645	0.5515	0.5951	0.6359	0.5505	0.5317	0.4795	0.8627	0.5558
	3B-Instruct	0.5736	0.4786	0.4480	0.8656	0.6609	0.6193	0.5492	0.4842	0.6125	0.5752	0.5457	0.4638	0.4274	0.4452	0.8723	0.5555
	Ins-RL	0.6136	0.5539	0.5004	0.8554	0.6981	0.6905	0.6269	0.4901	0.5613	0.6165	0.5965	0.5327	0.5336	0.5031	0.8612	0.5994
bamboogle	3B-Base	0.7371	0.6511	0.5901	0.8629	0.8487	0.8337	0.7255	0.5985	0.6641	0.5627	0.7163	0.6289	0.5825	0.5882	0.8700	0.5277
	Base-RL	0.7431	0.6551	0.4211	0.9086	0.8307	0.8194	0.7173	0.5332	0.6397	0.6044	0.7273	0.6291	0.5858	0.4206	0.9164	0.5078
	3B-Instruct	0.6531	0.5178	0.5872	0.9212	0.7448	0.7059	0.7029	0.5368	0.6932	0.4272	0.6331	0.5058	0.5465	0.5980	0.9244	0.4850
	Ins-RL	0.6812	0.5770	0.5446	0.8989	0.7780	0.7608	0.7591	0.5446	0.6443	0.5293	0.6418	0.5574	0.6713	0.5340	0.9029	0.6238
nq	3B-Base	0.6869	0.5852	0.5845	0.8513	0.7977	0.7812	0.6602	0.5648	0.6679	0.7342	0.6681	0.5683	0.5321	0.5847	0.8579	0.6174
	Base-RL	0.6722	0.6077	0.5175	0.8703	0.7611	0.7562	0.6607	0.5761	0.6443	0.5929	0.6535	0.5857	0.4601	0.5098	0.8816	0.5703
	3B-Instruct	0.5858	0.4723	0.5650	0.9212	0.6850	0.6522	0.6244	0.5665	0.7101	0.5465	0.5656	0.4613	0.4828	0.5664	0.9274	0.5348
	Ins-RL	0.6729	0.5914	0.5561	0.8857	0.7640	0.7569	0.7018	0.5347	0.6221	0.5983	0.6547	0.5767	0.6064	0.5532	0.8892	0.5916
triviaqa	3B-Base	0.6221	0.5380	0.5686	0.8735	0.7385	0.7228	0.6043	0.5545	0.6626	0.6937	0.6027	0.5279	0.4917	0.5678	0.8817	0.6537
	Base-RL	0.5612	0.4902	0.5049	0.8796	0.6552	0.6414	0.5547	0.5355	0.6233	0.6340	0.5444	0.4780	0.4269	0.5007	0.8876	0.5424
	3B-Instruct	0.4617	0.3658	0.5300	0.9207	0.5373	0.4879	0.4789	0.5159	0.7125	0.5377	0.4568	0.3602	0.3665	0.5350	0.9250	0.5271
	Ins-RL	0.4841	0.4221	0.5485	0.9133	0.5872	0.5706	0.5391	0.5167	0.6218	0.5092	0.4675	0.4104	0.4686	0.5512	0.9209	0.4482

Table 8: Unified appendix calibration table (KL-weighted, Max, Mean) across models without external information.

Dataset	Model	GMean						RMS					
		ECE-F	ECE-B	ECE-V	AUC-F	AUC-B	AUC-V	ECE-F	ECE-B	ECE-V	AUC-F	AUC-B	AUC-V
hotpotqa	Ins-Format	0.5211	0.3804	0.3533	0.5486	0.9408	0.5433	0.5383	0.4737	0.4068	0.5438	0.9292	0.5349
	Format-RL	0.3986	0.2688	0.2390	0.6284	0.9459	0.7019	0.4252	0.3433	0.2951	0.6228	0.9253	0.7132
popqa	Ins-Format	0.3977	0.2900	0.2928	0.6225	0.9836	0.6108	0.4232	0.3840	0.3153	0.6150	0.9786	0.6174
	Format-RL	0.3133	0.1992	0.1725	0.7453	0.9809	0.7124	0.3322	0.2902	0.2218	0.7436	0.9782	0.6916
2wikimultihopqa	Ins-Format	0.5157	0.3871	0.3559	0.5134	0.9161	0.5224	0.5284	0.4636	0.4202	0.5083	0.8928	0.5127
	Format-RL	0.4269	0.3003	0.2851	0.6360	0.9162	0.5435	0.4490	0.3835	0.3342	0.6328	0.8976	0.5666
bamboogle	Ins-Format	0.5704	0.3874	0.4163	0.4860	0.9652	0.5465	0.6002	0.5178	0.4843	0.4924	0.9596	0.5733
	Format-RL	0.5435	0.3690	0.3902	0.7048	0.9444	0.7238	0.5639	0.4806	0.4171	0.6944	0.9380	0.7274
nq	Ins-Format	0.4619	0.3439	0.3196	0.5357	0.9413	0.6038	0.4761	0.4234	0.3347	0.5331	0.9324	0.6175
	Format-RL	0.3270	0.2254	0.1926	0.6983	0.9570	0.6920	0.3482	0.3069	0.2106	0.6935	0.9530	0.7013
triviaqa	Ins-Format	0.3114	0.2461	0.2611	0.5706	0.9663	0.5599	0.3257	0.2940	0.2431	0.5675	0.9571	0.5767
	Format-RL	0.2256	0.1533	0.1463	0.6414	0.9354	0.6731	0.2276	0.2020	0.1384	0.6338	0.9263	0.6798

Table 9: Unified calibration table (GMean + RMS) On Format and Format-RL.

		KL-Weighted				Max						Mean					
Dataset	**Model**	ECE-F	ECE-B	AUC-F	AUC-B	ECE-F	ECE-B	ECE-V	AUC-F	AUC-B	AUC-V	ECE-F	ECE-B	ECE-V	AUC-F	AUC-B	AUC-V
hotpotqa	Ins-Format	0.5522	0.4469	0.5445	0.9337	0.6539	0.6200	0.5629	0.5169	0.6825	0.5510	0.5334	0.4357	0.3778	0.5464	0.9369	0.5396
	Format-RL	0.4346	0.3177	0.6213	0.9342	0.5367	0.4814	0.4455	0.5963	0.7071	0.7229	0.4043	0.3104	0.2734	0.6252	0.9377	0.7044
popqa	Ins-Format	0.4346	0.3540	0.6137	0.9800	0.5534	0.5201	0.4712	0.5509	0.7071	0.6108	0.4069	0.3455	0.2963	0.6188	0.9817	0.6117
	Format-RL	0.3418	0.2605	0.7442	0.9790	0.4666	0.4360	0.3838	0.6849	0.7402	0.6722	0.3260	0.2517	0.1993	0.7443	0.9801	0.7025
2wikimultihopqa	Ins-Format	0.5415	0.4453	0.5039	0.9015	0.6377	0.6017	0.5560	0.4801	0.6153	0.5283	0.5245	0.4333	0.3802	0.5112	0.9075	0.5172
	Format-RL	0.4613	0.3582	0.6331	0.9049	0.5683	0.5261	0.4711	0.5928	0.6293	0.6313	0.4351	0.3444	0.3097	0.6346	0.9093	0.5540
bamboogle	Ins-Format	0.6140	0.4828	0.5064	0.9630	0.7433	0.7018	0.6340	0.6138	0.7570	0.6557	0.5845	0.4617	0.4496	0.4856	0.9641	0.5616
	Format-RL	0.5728	0.4517	0.6892	0.9408	0.6682	0.6423	0.5530	0.6256	0.6916	0.7040	0.5651	0.4361	0.3902	0.6980	0.9424	0.7208
nq	Ins-Format	0.4907	0.4044	0.5313	0.9351	0.5838	0.5569	0.4636	0.4920	0.6705	0.6161	0.4671	0.3883	0.3234	0.5347	0.9376	0.6103
	Format-RL	0.3627	0.2829	0.6958	0.9545	0.4575	0.4286	0.3422	0.6528	0.7140	0.6949	0.3395	0.2731	0.1985	0.6961	0.9555	0.6943
triviaqa	Ins-Format	0.3316	0.2830	0.5721	0.9607	0.4018	0.3715	0.3314	0.5472	0.7400	0.5950	0.3159	0.2767	0.2251	0.5687	0.9632	0.5671
	Format-RL	0.2410	0.1927	0.6306	0.9290	0.3213	0.2894	0.2367	0.6014	0.7200	0.7053	0.2162	0.1813	0.1414	0.6379	0.9318	0.6750

Table 10: Unified appendix calibration table (KL-weighted, Max, Mean) On Format and Format-RL.

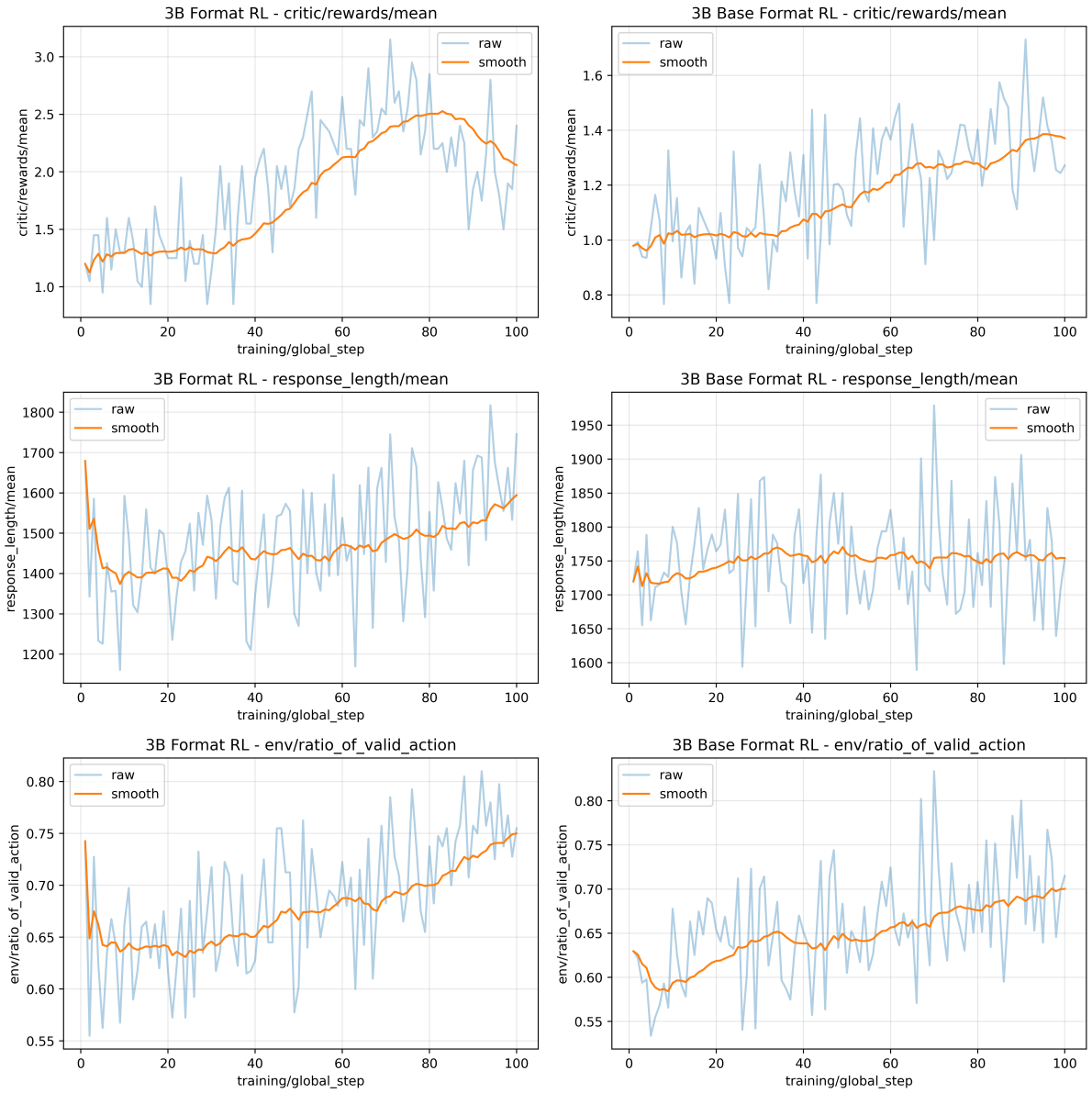

Figure 4: Comparison of training dynamics between the 3B Format RL policy and the 3B Base Format RL policy. Each row plots one metric over the first 100 global training steps: *(top)* critic reward (`critic/rewards/mean`), *(middle)* average response length (`response_length/mean`), and *(bottom)* ratio of valid actions in the environment (`env/ratio_of_valid_action`). Within each subplot we show the raw logged values (light blue) and their moving average (orange), with the left column corresponding to 3B Format RL and the right column to 3B Base Format RL. Here, *3B Format RL* denotes a 3B policy trained from scratch with a format-only reward for cold start, while *3B Base Format RL* continues training from this format-cold-start checkpoint by adding the calibration reward on top.

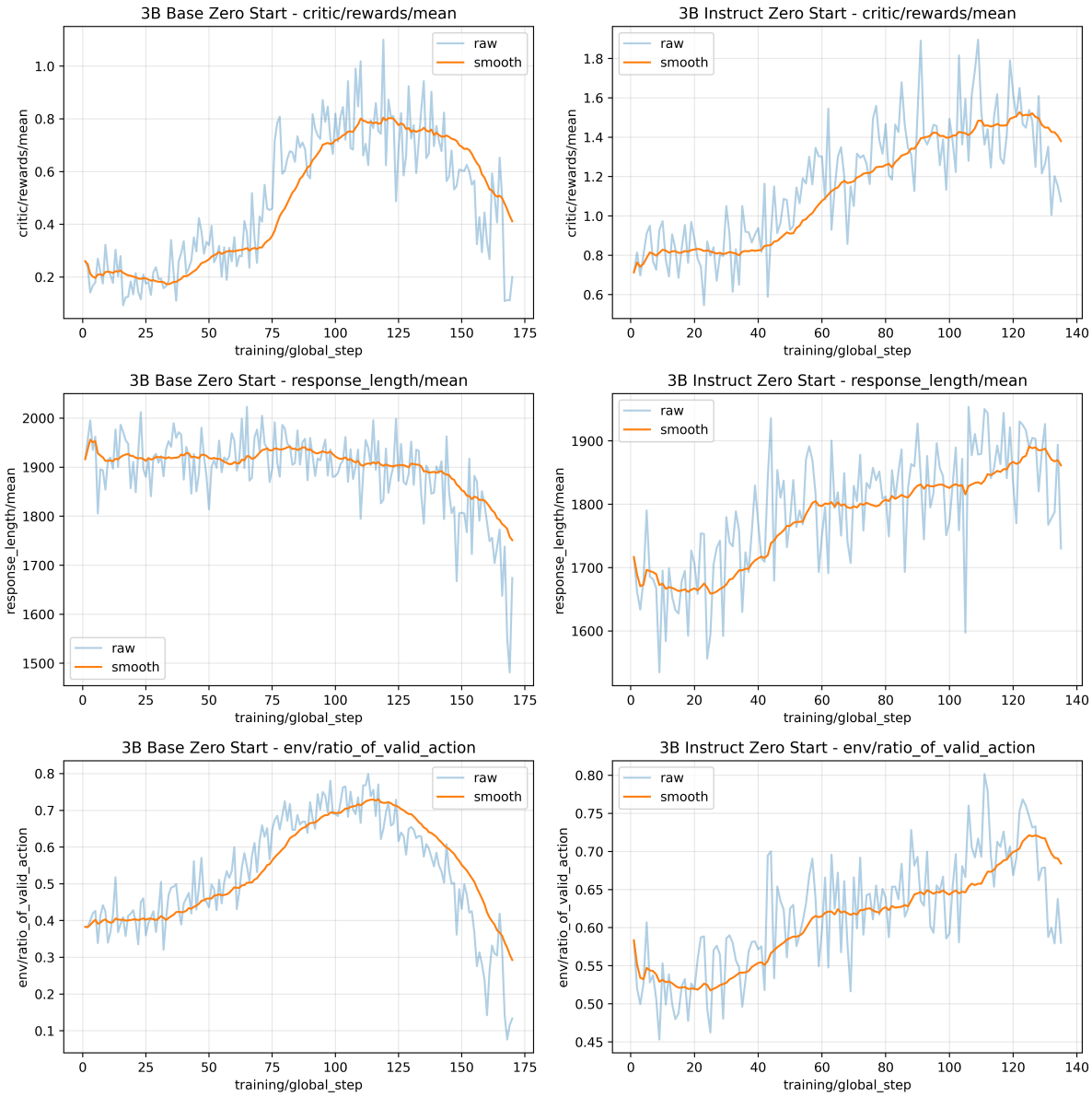

Figure 5: Training dynamics of the 3B Base Zero Start and 3B Instruct Zero Start policies. Each row reports a different metric as a function of the global training step: *(top)* critic reward (`critic/rewards/mean`), *(middle)* average response length (`response_length/mean`), and *(bottom)* ratio of valid actions taken in the environment (`env/ratio_of_valid_action`). For each metric we plot the raw measurements (light blue) and a smoothed trend (orange), with the left column showing the 3B Base Zero Start run and the right column showing the 3B Instruct Zero Start run. Here, *3B Base Zero Start* denotes a policy initialized from the **base** model, while *3B Instruct Zero Start* is initialized from the **instruct** model.

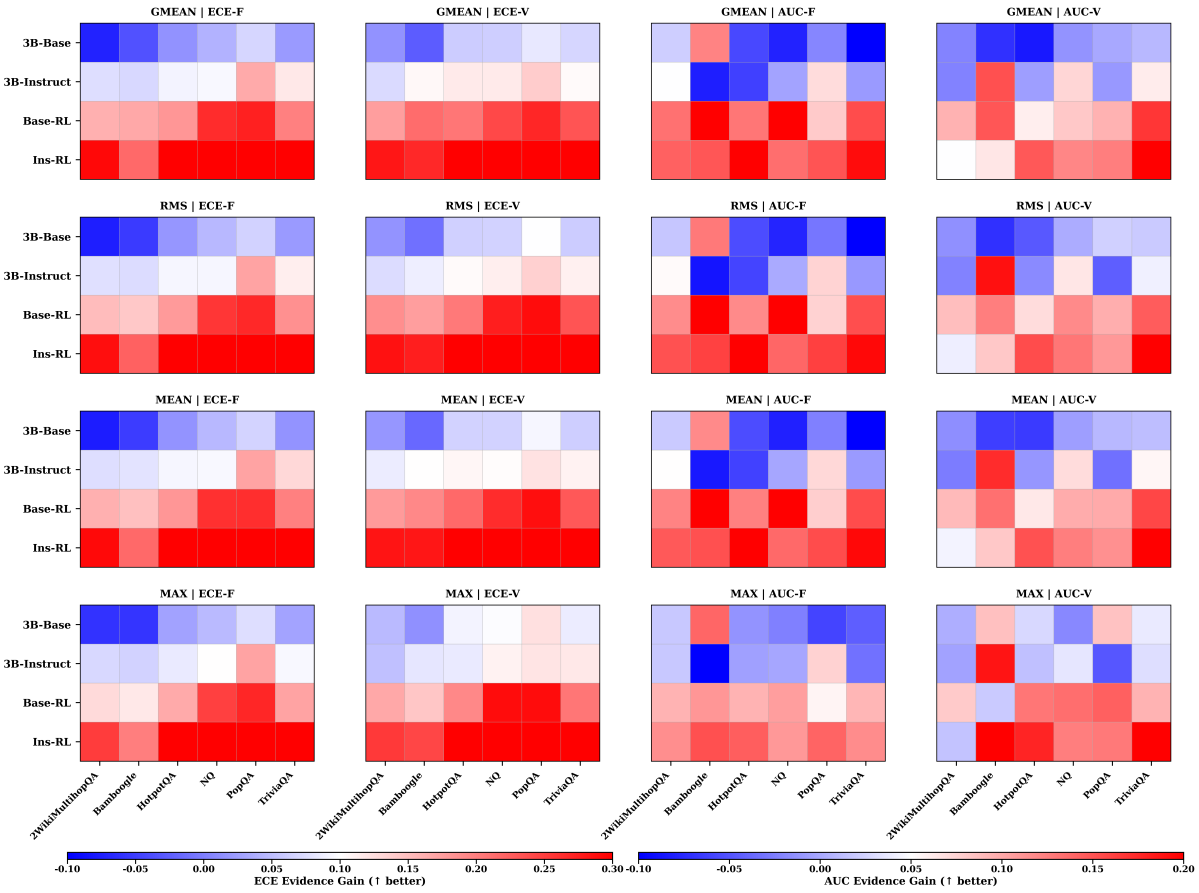

Figure 6: Evidence gain under the GMEAN, RMS, MEAN, MAX aggregation setting. Heatmaps report **ECE-Gain** (left two; ECE-F and ECE-V) and **AUC-Gain** (right two; AUC-F and AUC-V) for four model variants across six QA datasets. Positive ECE-Gain indicates reduced miscalibration with retrieval evidence, while positive AUC-Gain indicates improved correct incorrect separability with retrieval evidence.