ELICITING ACTION UNCERTAINTY VIA REASONING INTERRUPTION FOR TRAINING CALIBRATED TOOL-USING LLMS

Anonymous authors
Paper under double-blind review

ABSTRACT

Reinforcement learning improves the execution accuracy of tool-using LLMs but often leaves them poorly calibrated, causing the models to remain overconfident in incorrect actions and thereby undermining their reliability and trustworthiness. Existing approaches inject a calibration term into the optimization objective using uncertainty estimates aggregated over the entire reasoning-to-action sequence. Such trace-level estimates conflate linguistic fluency with action correctness: because action-relevant information is sparse, the resulting uncertainty signal is diluted by generic reasoning tokens. To recover an action-faithful uncertainty signal, we interrupt the reasoning trace at step boundaries and regenerate tool calls from each prefix. Our pilot analysis shows that the action's log-likelihood, evaluated under such interrupted prefixes, ranks correct tool calls above failures more reliably than its full-trace counterpart. Based on this observation, we propose **UARI** (Uncertainty-aware Reasoning Interruption), an RL framework that turns this interrupted-action signal into an uncertainty-aware training objective: candidates are grouped by uncertainty-induced difficulty and the same signal supplies a bounded reward-shaping term, jointly optimizing execution correctness and calibration. By targeting an uncertainty signal decoupled from generic reasoning tokens, UARI addresses miscalibration at the specific juncture of action commitment. Empirical evaluations on the BFCL-v4 and Nestful benchmarks demonstrate that UARI consistently outperforms the standard GRPO baseline in both execution accuracy and the calibration performance.

1 INTRODUCTION

Reinforcement learning (RL) has substantially improved the performance of tool-use agents, enabling them to leverage external tools and execute complex tasks (Qian et al., 2025; Zeng et al., 2025; Wu et al., 2025; Singh et al., 2025). However, recent studies have shown that such RL training induces severe overconfidence, with agents exhibiting degraded uncertainty estimates and elevated hallucination rates, particularly when trained with reward signals that emphasize correctness alone (Xuan et al., 2026; Hammoud et al., 2025; Li et al., 2025). In high-stakes deployment settings where tool invocations can trigger irreversible external effects, an agent must not only act correctly but also reliably communicate the uncertainty of its predictions so that risky actions can be appropriately deferred (Yao et al., 2024; Zhang et al., 2026; Shi et al., 2025b).

Token-level log-probabilities and entropy along the generation path are widely used for confidence estimation, as they capture the degree of certainty that a model has in its output (Fu et al., 2025; Taubenfeld et al., 2025; Zhang, 2025). In multi-step reasoning, such signals are often informative because the reasoning process approximately follows a causal chain: each step constrains the steps that follow, meaning that uncertainty accumulated along the trace can serve as a proxy for the uncertainty of the final answer (Zhang & Zhang, 2025; Sun et al., 2026).

Nonetheless, this intuition does not transfer cleanly to tool-calling tasks (Xuan et al., 2026; Li et al., 2025). The distinction lies in how the correctness of the output is coupled with the generated trace. For multi-step reasoning, the answer represents the terminal point of a reasoning chain, $y_{\text{reason}} : s_1 \to s_2 \to \cdots \to s_Q$, where uncertainty in an intermediate step s_k may propagate to

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.

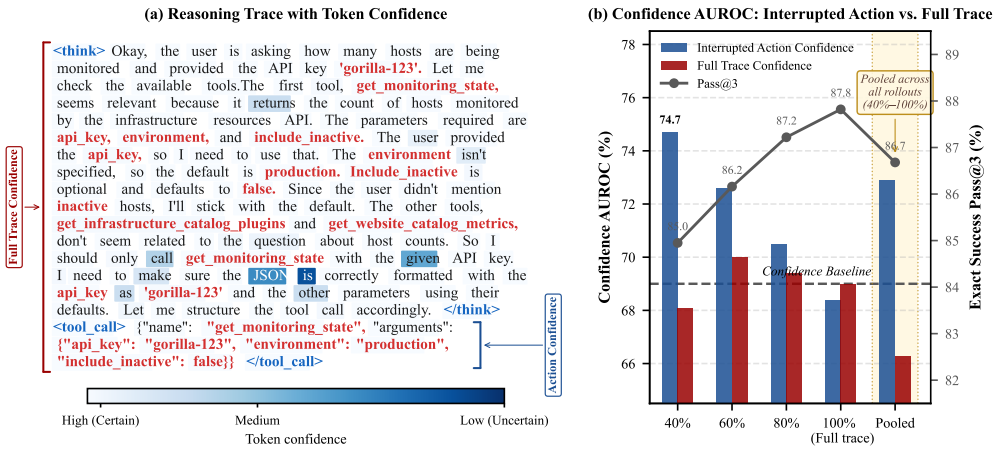

Figure 1: **(a) Generic certainty dilutes confidence signals.** Trace confidence averages over numerous near-certain generic tokens, whereas uncertainty clusters around sparse, action-relevant keywords. Darker backgrounds indicate higher uncertainty; red and blue bold tokens denote action keywords and structural tags, respectively. **(b) Interruption yields cleaner signals.** Interrupted action confidence is more reliable than trace confidence in both individual and pooled rollouts. AUROC denotes the probability that a correct action outranks an incorrect one in confidence. Pass@3 uses 32 inference sampled traces; *Pooled* aggregates candidates from all rollout depths (40%–100%) for use in UARI optimization.

subsequent steps and eventually affect the final answer (Zhou et al., 2026; Yin et al., 2024; Zhang et al.). In contrast, a tool-calling action is a structured output whose components are grounded in the input query rather than derived sequentially along the trace: $y_{\text{tool}} : s_{\text{func}} \oplus s_{\text{arg},1} \oplus \cdots \oplus s_{\text{arg},M}$, where s_{func} denotes the step associated with function prediction, and $\oplus$ represents composition rather than temporal derivation (Bayat et al., 2025; Yao et al., 2024). Figure 1(a) illustrates this structure: uncertainty is concentrated within sparse, action-relevant tokens corresponding to these components, whereas most structural and continuation tokens remain near-certain. Aggregating confidence over the full trace, therefore, dilutes these sparse, action-relevant signals with a much larger pool of generic continuation tokens, weakening the alignment between the resulting uncertainty signals and action correctness (Wang et al., 2026; Jiang et al., 2025).

This raises a question: *which signal better captures action uncertainty and can be optimized for both correctness and calibration?* A useful signal should indicate whether the current reasoning state is sufficient for committing to actions: sufficient prefixes should yield stable, high-confidence actions, while underspecified prefixes should more often lead to lower confidence or divergent actions. We therefore probe intermediate reasoning prefixes by regenerating a tool call from each one. The confidence of these regenerated actions may serve as a more reliable proxy for action-level uncertainty. Figure 1(b) supports this hypothesis: interrupted action confidence achieves higher AUROC than full-trace confidence at useful interrupted prefixes and in the pooled rollout set, indicating stronger ranking of successful tool calls over failures despite modest reductions in execution accuracy. This observation suggests that reasoning interruption exposes action uncertainty and provides a natural candidate pool for jointly optimizing correctness and calibration.

Based on this insight, we propose **UARI** (Uncertainty-aware Reasoning Interruption), an RL framework that turns interrupted action confidence into a training signal for calibrated function-calling LLMs. UARI first locates natural step boundaries within each reasoning trace, regenerates a complete tool call from every interrupted prefix, and evaluates each regenerated action using both execution correctness and interrupted action confidence. This yields a rollout pool whose candidates vary in reasoning state, action commitment, correctness, and confidence.

UARI then uses this pool to make rollout group comparisons more informative. Since interrupted action confidence ranks correct actions above failures more reliably than trace confidence, it provides a proxy for local action difficulty and helps avoid comparisons dominated by either trivial successes or hopeless failures (Yu et al., 2025; Zhang et al., 2025). Finally, UARI updates the policy with a joint advantage that encourages both execution success and confidence calibration. Evaluations on Nestful and BFCL-v4 show that UARI improves execution accuracy over standard GRPO baselines

while reducing the Brier score, and further analyses confirm that interrupted action confidence is an effective optimization signal.

2 PRELIMINARIES

2.1 TOOL-USING RL WITH GRPO

In tool-use settings, each decision-making step generates a reasoning segment $y^{\text{reason}} = (z_1, \ldots, z_{T_r})$ followed by an executable action $y^{\text{act}} = (u_1, \ldots, u_L)$, given context x (Yao et al., 2024; Oh et al., 2025). Although only y^{act} interacts with the environment, its generation is inherently conditioned on the preceding reasoning trace. Following (Zhao et al., 2025; Feng et al., 2025), we adopt the GRPO framework (Shao et al., 2024) for optimization, where a group of K outputs $G = \{o_i\}$ is sampled for each context. The rewards $\{R_i\}$, which incorporate partial credit for correct tool names and arguments (Qian et al., 2025), are used to compute the group-normalized advantage:

$$A_i = \frac{R_i - \mu_G^R}{\sigma_G^R + \epsilon}. \tag{1}$$

The policy is updated by maximizing

$$\mathcal{L}_{\text{GRPO}}(\theta) = \mathbb{E}_{x,G}\left[\frac{1}{K}\sum_{i=1}^{K}\sum_{t=1}^{T_i}\min\Big(\varrho_{i,t}A_i,\ \hat{\varrho}_{i,t}A_i\Big)\right] - \beta\,\mathbb{E}_x\big[D_{\text{KL}}(\pi_\theta \,\|\, \pi_{\text{ref}})\big], \tag{2}$$

where μ_G^R, σ_G^R denote the mean and standard deviation of rewards within the group, respectively. π_{ref} is the frozen reference policy, $\varrho_{i,t} = \pi_\theta(o_{i,t} \mid x, o_{i,<t})/\pi_{\theta_{\text{old}}}(o_{i,t} \mid x, o_{i,<t})$ is the importance ratio, and $\hat{\varrho}_{i,t} = \text{clip}(\varrho_{i,t}, 1-\epsilon_c, 1+\epsilon_c)$ is the clipped ratio. Since A_i is defined relative to the sampled group G, the construction of the group serves as a fundamental component of the optimization signal (Yu et al., 2025; Zhang et al., 2025). To isolate precise interrupted action confidence signals, this study utilizes training data consisting of single-step decisions without environmental interaction.

2.2 UNCERTAINTY SIGNALS FOR TOOL CALLING

Base uncertainty signals. The standard tool-using GRPO objective highlights a potential limitation of correctness-only training: while A_i effectively ranks actions by success, it overlooks the calibration of the model, often resulting in overconfidence during incorrect action commitment (Damani et al., 2025; Ma et al., 2026). To integrate calibration into the tool-using optimization process, it is essential to identify an uncertainty signal that is relevant to the action and comparable across candidates (Wei et al., 2025; Zhang et al., 2026; Xiong et al., 2023). Token log-probabilities are the standard basis for defining such signals (Kadavath et al., 2022; Kang et al., 2025; Fu et al., 2025; Li et al., 2026). For a subsequence $y_{i:j}$ under context x, define the mean log-likelihood:

$$\text{C}(y_{i:j}) = \tfrac{1}{j-i+1}\sum_{\ell=i}^{j}\log p_\theta(y_\ell \mid x, y_{<\ell}),$$

where a larger value of $\text{C}(\cdot)$ indicates higher confidence and lower uncertainty. For a trace decomposed into a reasoning segment s and an action a, we define two primary signals:

$$C_{\text{act}} := \text{C}(a \mid x, s), \qquad C_{\text{trace}} := \text{C}(s, a). \tag{3}$$

Decoupling action from reasoning via interruption. In tool calling settings, both C_{trace} and C_{act} may be suboptimal as signals: the former tends to conflate the reasoning narrative with action commitment, while the latter can be susceptible to interference from generic continuation tokens (see Figure 1). As detailed in Appendix B.1, C_{trace} is highly sensitive to reasoning order even when the correctness of the regenerated action remains unchanged. This supports the interpretation that trace-level confidence captures variations in narrative order that are unrelated to action correctness.

To mitigate this issue, we interrupt the reasoning prefix to probe action uncertainty along the reasoning paths, offering a measure of uncertainty that is less confounded by narrative noise. Formally, let $p_\rho := y^{\text{reason}}_{1:\lfloor \rho T_r \rfloor}$ denote the interrupted prefix retaining a fraction ρ of the T_r-token reasoning prefix. From this prefix, the confidence of interrupted action $\hat{a}_\rho \sim \pi_\theta(\cdot \mid x, p_\rho)$ is defined as:

$$\hat{C}_{\text{act}}(\rho) := \text{C}(\hat{a}_\rho \mid x, p_\rho). \tag{4}$$

Furthermore, we investigate whether the signal $\hat{C}_{\text{act}}$ conveys additional action-relevant information that is not present in C_{trace}. We regress C_{trace} on $\hat{C}_{\text{act}}$ and evaluate the predictive power of the resulting residual. The results in Appendix B.2 demonstrate that this residual contributes negligible improvement to the held-out AUROC for the prediction of action correctness, with the ΔAUROC remaining non-positive or near zero across all prefix ratios. These findings indicate that the additional variation in C_{trace} beyond $\hat{C}_{\text{act}}$ constitutes noise rather than a useful signal. Consequently, we adopt the interrupted action confidence as an efficient optimization signal for the joint improvement of execution correctness and the calibration of the model in tool-using RL.

3 METHODOLOGY

Section 2.2 establishes that reasoning-interrupted action confidence $\hat{C}_{\text{act}}(\rho)$ is a more discriminative predictor of action correctness than standard trace uncertainty. Building on these insights, we propose Uncertainty-aware Reasoning Interruption (UARI), an RL framework that enhances both execution accuracy and model calibration by integrating action-level uncertainty signals derived from reasoning interruption into the optimization objective of GRPO (Figure 2).

3.1 REASONING INTERRUPTION FROM STEP BOUNDARIES

Reasoning interruption proposes intermediate prefixes from which actions can be regenerated. Motivated by confidence-based reasoning segmentation (Liu et al., 2025; Zhang, 2025; Li et al., 2026), we use token-level entropy to propose interruption locations and align the resulting cuts with sentence boundaries. These locations are practical prefix proposals rather than theoretically optimal decision boundaries. The proposal rule determines which prefixes are examined; the definitions of interrupted action confidence, confidence-guided group selection, and bounded shaping remain unchanged.

For each prompt, we sample four complete source rollouts to obtain reasoning prefixes. These source rollouts are not included in the candidate pool; only actions regenerated from interrupted prefixes enter group selection. Formally, let $y^{\text{reason}} = (z_1, \ldots, z_{T_r})$ represent the reasoning segment and h_t denote the entropy of the next-token distribution under $\pi_{\theta_{\text{old}}}$ at position t. High-entropy positions are filtered and aligned with sentence boundaries to obtain interruption points; the thresholds and alignment procedure are specified in Appendix F.1. This yields a set of interrupted prefixes $\{p^{(n)}\}_{n=1}^{N_x}$. For each interrupted prefix, we sample an interrupted action $\hat{a}^{(n)} \sim \pi_{\theta_{\text{old}}}(\cdot \mid x, p^{(n)})$, compute its interrupted action confidence $\hat{C}_{\text{act}}^{(n)}$, and evaluate a correctness reward $R^{(n)} = s(\hat{a}^{(n)}, a^{\star})$ based on Qian et al. (2025). This reward function incorporates partial credit for matches and penalties for formatting errors, thereby providing fine-grained scores. Each interrupted candidate is stored as a tuple $c_n = (\tilde{o}^{(n)}, \hat{C}_{\text{act}}^{(n)}, R^{(n)})$, where $\tilde{o}^{(n)} = (p^{(n)}, \hat{a}^{(n)})$ denotes the reconstructed trace. We aggregate these regenerated candidates across the four source rollouts into a single pool of size N_x for each prompt, from which a group of $K = 6$ is selected.

3.2 UNCERTAINTY-AWARE SAMPLING AND OPTIMIZATION

Uncertainty-aware sampling. Following the reasoning interruption process, the resulting candidate pool $\{c_n\}_{n=1}^{N_x}$ varies across both the reasoning state and the action commitment. Although this variation provides a comprehensive set of candidates, it introduces a critical challenge for the subsequent optimization: an unconstrained sampling from such a diverse pool frequently yields groups that are dominated entirely by either trivial successes or hopeless failures (Wang et al., 2025a; Le et al., 2025; Yu et al., 2025). To construct informative training groups, the discriminative power of the interrupted action confidence $\hat{C}_{\text{act}}^{(n)}$ is leveraged as a guiding signal.

Specifically, UARI ranks all interrupted candidates by $\hat{C}_{\text{act}}^{(n)}$ and confines selection to a median-centered window $\mathcal{W}_x$. Rather than using a fixed window size, the window size M is drawn uniformly at random from $[\min(K{+}1, N_x), \min(2K, N_x)]$ at each training step, introducing stochasticity that prevents the group construction from overfitting to a specific difficulty stratum. The primary motivation for centering on the median is to ensure that group-relative comparisons occur among candidates of comparable difficulty, thereby preventing extreme outliers from distorting the group's statistical baseline (Shi et al., 2025a; Schaul et al., 2015; Yang et al., 2026). Within this window, the

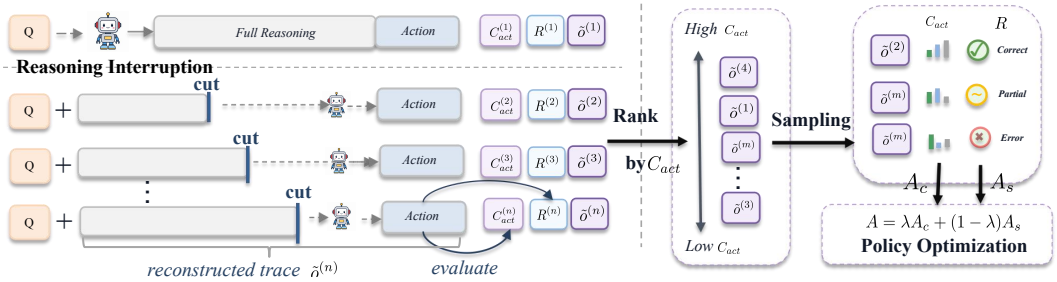

Figure 2: Overview of UARI. UARI interrupts reasoning traces to extract reliable action uncertainty. This signal guides the construction of GRPO groups with high reward contrast and a confidence-based objective, jointly enhancing execution accuracy and the calibration of the model.

subgroup G_x of size K that maximizes correctness reward variance is selected:

$$M \sim \mathcal{U}\{\min(K{+}1, N_x),\ \min(2K, N_x)\}, \quad G_x = \underset{G \subset \mathcal{W}_x,\ |G|=K}{\text{first-argmax}} \operatorname{Var}\big(\{R^{(g)} : g \in G\}\big) \tag{5}$$

Ties are broken by the initial candidate order. If $|\mathcal{W}_x| < K$, sampling with replacement from $\mathcal{W}_x$ maintains a consistent group size for normalization.

Uncertainty-aware policy optimization. Upon the selection of the subgroup G_x via uncertainty-aware sampling, the interrupted action confidence $\hat{C}^{(i)}_{\text{act}}$ is further utilized as a bounded shaping term to facilitate the joint optimization of the action correctness and the model calibration. Given the local group G_x, two group-normalized advantage signals are constructed: the correctness advantage A_s, which captures the quality of the external action, and the confidence advantage $\tilde{A}_c$, which captures the confidence of the model regarding the interrupted actions within the group:

$$A_s(i) = \frac{R^{(i)} - \mu^R_{G_x}}{\sigma^R_{G_x} + \epsilon}, \qquad \tilde{A}_c(i) = \frac{\hat{C}^{(i)}_{\text{act}} - \mu^C_{G_x}}{\sigma^C_{G_x} + \epsilon}.$$

The two signals are combined through a confidence-weighted joint advantage:

$$A_d(i) = (1 - \lambda_{\text{eff}})A_s(i) + \lambda_{\text{eff}} A_c(i), \qquad A_c(i) = \text{clip}\left(\tilde{A}_c(i),\ \min_{j \in G_x} A_s(j),\ \max_{j \in G_x} A_s(j)\right), \tag{6}$$

where $\lambda_{\text{eff}} = \lambda \cdot \mathbf{1}[\exists i : R^{(i)} > 0]$. The confidence advantage is clipped to the same range as the score advantage within the sampled group, so the interpolation combines two normalized signals on a matched scale. When all candidates receive zero reward, $\lambda_{\text{eff}} = 0$ disables confidence shaping, preventing confidence from inducing spurious preferences among uniformly failed actions. We adapt the standard GRPO objective in Eq. 2 to the candidate set G_x by substituting conventional components with the difficulty-aware advantage $A_d(i)$ and reference policy $\pi_{\theta_{\text{old}}}$. The resulting objective encourages the model to improve both action correctness and calibration.

4 Experiments

4.1 Experimental Setup

Benchmarks. We evaluate our method on two tool-use benchmarks that cover different aspects of agentic interaction. The Berkeley Function Calling Leaderboard (BFCL-v4) (Patil et al., 2023) provides a broad assessment of API generalization by testing single- and multi-turn interactions across various tools. For more complex scenarios, we use Nestful (Basu et al., 2025), which requires sequential reasoning and interdependent tool executions in nested REST-style workflows. Together, these datasets allow us to evaluate both general function calling and multi-step planning.

Metrics. We use several metrics to evaluate task success and confidence estimation. For execution performance, we follow the standard success scores defined by each benchmark. To evaluate confidence ranking, we use AUROC to measure the probability that the correct one is assigned a higher confidence score. We also report the Brier score (Brier, 1950) to measure confidence calibration. Defined as the mean squared error between predicted confidence and binary correctness, the Brier score accounts for both calibration and sharpness.

Table 1: Tool-use accuracy (%) on Qwen3-4B and Llama-3.2-3B-Instruct. Unreported results are marked "–". #P: number of parameters. Infer.#: number of inference calls per sample; inference-time methods are applicable only to single step evaluation. Bold marks the best result within each backbone; gray numbers indicate the difference from the corresponding baseline.

Method	#P	Infer.#	Nestful	BFCL Single-Turn		BFCL Multi-Turn
			Full Match	Non-Live	Live	Overall
Reference Open-Source Models						
ToolACE-2-8B (FC) (Zeng et al., 2025)	8B	1	0.10	87.10	77.42	38.38
Arch-Agent-3B	3B	1	0.60	86.67	72.91	34.88
Qwen3-8B	8B	1	10.70	87.58	80.53	41.75
Backbone						
Qwen3-4B (Baseline)	4B	1	10.8	87.48	82.16	32.38
Inference-Time Confidence Selection						
+ DeepConf (Fu et al., 2025)	4B	32	9.50 (−1.3)	87.39 (−0.09)	81.93 (−0.23)	–
+ CICS (Taubenfeld et al., 2025)	4B	32	**11.90** (+1.1)	88.35 (+0.87)	82.68 (+0.52)	–
+ ThinkBrake (Oh et al., 2025)	4B	32	11.80 (+1.0)	76.00 (−11.48)	76.02 (−6.14)	–
Reinforcement Learning						
+ Plain GRPO (Qian et al., 2025)	4B	1	10.40 (−0.4)	88.63 (+1.15)	83.05 (+0.89)	34.88 (+2.50)
+ RLCR (Damani et al., 2025)	4B	1	9.5 (−1.3)	89.48 (+2.00)	83.05 (+0.89)	36.50 (+4.12)
+ CAR (Xuan et al., 2026)	4B	1	9.5 (−1.3)	88.60 (+1.12)	82.98 (+0.82)	31.62 (−0.76)
+ DCPO (Ma et al., 2026)	4B	1	10.8 (+0.0)	87.90 (+0.42)	**83.42** (+1.26)	33.62 (+1.24)
+ UARI (Ours)	4B	1	**11.90** (+1.1)	**90.21** (+2.73)	83.20 (+1.04)	**36.75** (+4.37)
Llama-3.2-3B-Instruct						
Llama-3.2-3B (Baseline)	3B	1	–	82.54	58.40	3.63
+ Plain GRPO	3B	1	–	**83.62** (+1.08)	60.99 (+2.59)	4.00 (+0.37)
+ RLCR	3B	1	–	82.08 (−0.46)	58.25 (−0.15)	3.88 (+0.25)
+ CAR	3B	1	–	80.19 (−2.35)	61.07 (+2.67)	4.62 (+0.99)
+ DCPO	3B	1	–	83.04 (+0.50)	58.25 (−0.15)	3.88 (+0.25)
+ UARI (Ours)	3B	1	–	83.10 (+0.56)	**62.84** (+4.44)	**5.62** (+1.99)

Baselines. The main comparison uses the Qwen3-4B backbone. We additionally evaluate Llama-3.2-3B-Instruct and Granite-3.3-2B-Instruct to assess transfer across model families. **DeepConf** (Fu et al., 2025) computes confidence scores by aggregating the log-probabilities of the top-K candidate tokens, **CICS** (Taubenfeld et al., 2025) applies confidence-weighted voting over multiple traces, and **ThinkBrake** (Oh et al., 2025) is a tool-calling inference method that terminates reasoning when high uncertainty is detected prior to the final response to mitigate overthinking. Regarding the training baselines, Plain GRPO fine-tunes the model exclusively with tool-execution rewards, isolating the effect of outcome-driven RL as described in Qian et al. (2025). **RLCR** (Damani et al., 2025), **CAR** (Xuan et al., 2026) and **DCPO** (Ma et al., 2026) incorporates a calibration reward. We reproduce RLCR, CAR, and DCPO following their respective original papers, preserving their method-specific training objectives under the common tool-calling training and evaluation setting.

Training and inference details. We utilize the training dataset from Qian et al. (2025), which comprises diverse tool-calling instances featuring executable supervision. To improve training efficiency, we sample 20 rollouts per instance and exclude cases where all rollouts succeed, as such instances provide limited learning signals. This filtering process yields 2.3k training instances. To ensure a fair comparison, within each backbone comparison, Plain GRPO, RLCR, CAR, DCPO, and UARI use the same training data and optimization schedule; these methods differ only in their specific sampling strategies or reward terms. During evaluation, all models utilize the standard benchmark inference protocol and identical decoding settings. The specific number of traces utilized for confidence estimation by each method during inference is detailed in Table 1. More details are provided in Appendix F.

4.2 Main Results

Execution accuracy. Table 1 demonstrates that UARI achieves competitive execution performance in both single-step function calling and multi-turn tool-use settings. Improvements over post-hoc confidence selection methods, such as DeepConf, CICS, and ThinkBrake, indicate that selecting or terminating completed traces is insufficient when generic reasoning tokens dilute the confidence signal. Gains over Plain GRPO suggest that action-aligned confidence provides benefits beyond generic reinforcement learning. UARI outperforms other calibration-oriented RL methods in mean

Table 3: Component ablation on BFCL-v4 (Qwen3-4B), with accuracy in %. $\mathcal{I}$: interrupted-candidate group construction; $\mathcal{S}$: confidence-guided selection; $\mathcal{O}$: bounded shaping. All groups use $K = 6$; $\mathcal{I}$-enabled variants share the candidate pool, training examples, and update budget. Random selection samples this pool; trace- and action-guided selection differ only in the signal. Shaping Only retains conventional groups but computes interrupted $\hat{C}_{\text{act}}$ through prefix-based regeneration for shaping. Without $\mathcal{O}$, the objective is correctness-only. Δ: mean accuracy gain over Plain GRPO. Bold marks the best.

Variant	$\mathcal{I}$	$\mathcal{S}$	$\mathcal{O}$	Signal	Non-Live AST	Live AST	Multi-Turn	Avg.	Δ
Plain GRPO (Base)	–	–	–	–	88.63	83.05	34.88	68.85	–
Random selection	✓	–	–	–	87.92	82.31	32.00	67.41	-1.44
+ Shaping Only	–	–	✓	C_{trace}	89.42	83.12	33.50	68.68	-0.17
+ Shaping Only	–	–	✓	$\hat{C}_{\text{act}}$	89.17	82.46	32.00	67.88	-0.97
Trace-guided selection	✓	✓	–	C_{trace}	88.90	82.75	32.25	67.97	-0.88
Action-guided selection	✓	✓	–	$\hat{C}_{\text{act}}$	88.94	82.61	36.25	69.27	+0.42
Full Stack (C_{trace})	✓	✓	✓	C_{trace}	88.96	82.68	29.38	67.01	-1.84
UARI (Ours)	✓	✓	✓	$\hat{C}_{\text{act}}$	**90.21**	**83.20**	**36.75**	**70.05**	**+1.20**

BFCL-v4 accuracy. On Qwen3-4B, UARI achieves a mean BFCL-v4 accuracy of 70.05%, exceeding CAR and DCPO by $+2.32$ and $+1.74$, respectively. Among same-backbone methods, DCPO leads on Live, while UARI leads on Non-Live and Multi-Turn. ToolACE-2-8B and Arch-Agent-3B score poorly on Nestful; frequent parsing errors suggest limited generalization to nested workflows.

Generalization across model families. Table 1 extends the comparison to Llama-3.2-3B-Instruct. UARI achieves the highest mean accuracy (50.52%), improving over all calibration RL methods. Its gains are concentrated in Live and Multi-Turn; Plain GRPO retains the highest Non-Live accuracy. A further evaluation on Granite-3.3-2B-Instruct achieves 35.76% Single-Turn accuracy ($+3.08$ over the 32.68% baseline) (Appendix A). These results support effectiveness across model families.

Calibration performance. Table 2 evaluates final-action confidence C_{act} under standard inference, distinct from the interrupted $\hat{C}_{\text{act}}$ used during training. On BFCL-v4 Single-Turn, UARI outperforms CAR and DCPO in AUROC ($+18.37$ and $+16.20$) and Brier (-2.16 and -1.95), respectively. This pattern supports the central hypothesis in the introduction: training should emphasize uncertainty over the action rather than over the full reasoning narrative. UARI's joint mechanism uses interrupted action confidence, allowing policy optimization to improve execution accuracy while keeping model confidence better aligned with action correctness.

Table 2: Action-confidence quality on BFCL-v4 (Qwen3-4B). AUROC (%)↑; Brier↓ ($\times 100$). Bold marks the best per column.

	Non-Live AST		Live AST		Single-Turn	
Method	AUROC	Brier	AUROC	Brier	AUROC	Brier
Plain GRPO	**66.81**	12.73	66.87	17.16	66.48	15.14
RLCR	65.09	12.47	66.90	17.26	65.99	15.07
CAR	51.16	14.81	47.14	19.26	49.50	17.10
DCPO	55.76	15.42	47.99	18.83	51.67	16.89
UARI	63.71	**12.44**	**70.32**	**17.03**	**67.87**	**14.94**

4.3 Ablation Study

Pool enlargement alone does not explain the gain. Table 3 compares selection and shaping at a fixed group size. Random selection from the enlarged pool achieves 32.00% Multi-Turn accuracy, below Plain GRPO's 34.88%, while UARI reaches 36.75%. Thus, access to interrupted candidates alone does not account for UARI's improvement. The $+4.75$ gain combines selection and shaping; the following comparisons separate their contributions. Shaping Only adds confidence shaping to conventional groups without guided selection. The $\hat{C}_{\text{act}}$ variant uses interruption and regeneration only to compute shaping, so disabling $\mathcal{I}$ does not disable this scoring step. Neither shaping-only variant improves mean accuracy over Plain GRPO in these experiments.

Table 4: Interruption strategy comparison on BFCL-v4 under a matched prefix budget. Acc. (%) ↑: task accuracy; Len. ↓: mean reasoning tokens. All baselines use no confidence signal. Bold marks the best.

	Non-Live AST		Live AST		Multi-Turn		Avg.	
Method	Acc. ↑	Len. ↓	Acc. ↑	Len. ↓	Acc. ↑	Len. ↓	Acc. ↑	Len. ↓
Backbone	87.48	352	82.16	470	32.38	403	67.34	408
ThinkBrake (Oh et al., 2025)	76.00	335	76.02	**374**	—	—	—	—
Plain GRPO (Qian et al., 2025)	88.63	326	83.05	**374**	34.88	316	68.85	339
Fixed cuts (no conf.)	88.73	326	82.38	526	31.25	425	67.45	426
Adaptive cuts (no conf.)	87.92	292	82.31	531	32.00	325	67.41	383
UARI (Ours)	**90.21**	**279**	**83.20**	385	**36.75**	**305**	**70.05**	**323**

Interrupted action confidence improves candidate selection. Without confidence shaping, random, trace-guided, and action-guided selection achieve Multi-Turn accuracies of 32.00%, 32.25%, and 36.25%, respectively. Action-guided selection improves over random selection by $+4.25$ and over trace-guided selection by $+4.00$. The trace- and action-guided variants share candidate generation and median-window, reward-variance selection, differing only in confidence signal. This comparison supports the value of $\hat{C}_{\text{act}}$ for group construction beyond generic confidence-based selection. Selection can act as an implicit curriculum; this comparison isolates the action-level signal within that procedure. Figure 3 shows instability in a run without uncertainty-aware sampling.

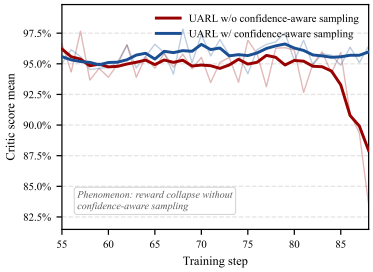

Figure 3: Training collapse in a run without uncertainty-aware sampling.

Bounded shaping adds to action-guided selection. Adding bounded $\hat{C}_{\text{act}}$ shaping to action-guided selection improves Multi-Turn accuracy ($+0.50$) and mean accuracy ($+0.78$). Non-Live and Live accuracy also increase, from 88.94% to 90.21% and from 82.61% to 83.20%, respectively. These results support an incremental contribution from shaping after candidate selection, consistent with Eq. 6, where correctness remains the primary reward signal and the clipped confidence advantage supplies a bounded adjustment. Action-guided selection alone already exceeds Plain GRPO in mean and Multi-Turn accuracy; the full UARI configuration provides a further improvement.

4.4 Analysis

Interruption serves as signal extraction. Table 4 compares fixed and adaptive cuts without confidence guidance. Both underperform Plain GRPO in mean accuracy (67.45% and 67.41% versus 68.85%), although adaptive cuts use fewer reasoning tokens than fixed cuts (383 versus 426). ThinkBrake also lowers Non-Live accuracy to 76.00%. UARI achieves the highest mean accuracy and shortest mean reasoning length. This supports signal extraction rather than compression alone. The cutting rule supplies candidate prefixes; action regeneration, $\hat{C}_{\text{act}}$ scoring, confidence-guided selection, and bounded shaping follow the same procedure under entropy-based or fixed-ratio proposals. UARI therefore does not require a particular cutting heuristic for these components.

Group size trade-off. The ablation of group size K in Table 5 shows a trade-off between within-group diversity and the strength of the reward signal. While performance is relatively stable across different values of K, $K = 6$ provides the best balance, achieving the highest scores in Non-Live and Multi-Turn scenarios. Smaller groups ($K = 4$) lack the diversity to improve Non-Live performance, while larger groups ($K = 8$) introduce cross-prefix bias that reduces accuracy in the Live split. This suggests that a moderate group size is necessary to maintain a distinct advantage signal that effectively handles varying prefix depths during reinforcement learning.

Sensitivity to uncertainty weight λ. Table 5 shows the sensitivity of the joint advantage mechanism to the uncertainty weight λ. A weight of $\lambda = 0.05$ is the most effective default, improving Non-Live and Live accuracy over the $\lambda = 0$ baseline. Increasing the weight to $\lambda = 0.10$ reduces performance, particularly in Multi-Turn accuracy (33.88%), suggesting that excessive interpolation toward the uncertainty signal can interfere with the task-driven reward signal. This indicates that λ must be tuned so that the external reward remains the primary optimization signal.

Table 5: Sensitivity to group size K (left) and confidence weight λ (right) on BFCL-v4 (accuracy in %). NL: Non-Live AST; L: Live AST; MT: Multi-Turn. Bold marks the best per group.

Group Size K	NL	L	MT	**Avg.**
$K = 4$	88.48	**83.57**	35.00	69.02
$K = 6$	**90.21**	83.20	**36.75**	**70.05**
$K = 8$	89.17	82.61	35.75	69.18

Weight λ	NL	L	MT	**Avg.**
$\lambda = 0.00$	88.94	82.61	36.25	69.27
$\lambda = 0.05$	**90.21**	**83.20**	**36.75**	**70.05**
$\lambda = 0.10$	89.08	82.90	33.88	68.62

5 Related Work

Reasoning truncation for LLMs. Recent work uses interruption to reduce unnecessary reasoning through budget control, adaptive stopping, pruning, or compression (Han et al., 2024; Aggarwal & Welleck, 2025; Hou et al., 2025; Arora & Zanette, 2025; Han et al., 2025). These methods show that shorter reasoning can often preserve or even improve accuracy, but their primary target is efficiency or generic reasoning quality rather than reliability of structured tool actions. These methods use interruption for efficiency; UARI uses it to expose prefixes for action-confidence estimation, extracting signals rather than compressing traces.

Uncertainty and confidence estimation in LLMs. Uncertainty estimation in LLMs has predominantly relied on log-probability-based approaches. For instance, Self-Certainty estimates confidence through the divergence from uniform distributions (Kang et al., 2025), DeepConf aggregates top-K token probabilities (Fu et al., 2025), and confidence-weighted voting improves self-consistency (Taubenfeld et al., 2025). These methods primarily address chain-of-thought and multi-step reasoning tasks, where reasoning follows a causal chain and token-level probabilities compound into informative uncertainty signals. In contrast to the continuous causal flow of chain-of-thought reasoning, tool-calling generates discrete actions consisting of function selection and argument bindings, which are often embedded within formulaic scaffolding tokens (Wang et al., 2026; Bayat et al., 2025). These scaffolding tokens inflate the confidence of the full trace without reflecting action correctness, resulting in the decoupling of token-level confidence from execution quality. Consequently, UARI isolates action-relevant uncertainty by utilizing the confidence of actions regenerated from partial prefixes. Intervening before the full narrative avoids scaffolding-induced confidence inflation and grounds the signal at action commitment.

Calibration in tool-use LLMs. Tool-use LLMs select tools and bind arguments to interact with external environments (Schick et al., 2023; Yao et al., 2023; Patil et al., 2023; Qin et al., 2023). Recent work applies reinforcement learning to improve tool-use execution performance (Hao et al., 2025; Zhao et al., 2025; Chen et al., 2026), but stronger execution does not ensure reliability: RL-trained agents exhibit severe overconfidence (Xuan et al., 2026; Li et al., 2025). Confidence-aware optimization addresses this degradation: Damani et al. (2025) jointly optimize correctness and verbalized confidence in general reasoning tasks, while Ma et al. (2026) decouple reasoning and calibration objectives in multi-step tasks. For tool-calling agents, Xuan et al. (2026) further identify tool-type-dependent confidence patterns and incorporate verbalized confidence into RL optimization. These approaches optimize confidence on completed traces without separating sparse action-relevant tokens from formulaic continuation. UARI uses prefix-conditioned action confidence for candidate ranking and bounded shaping, emphasizing action uncertainty over trace fluency.

6 Conclusion

While reinforcement learning has significantly enhanced the performance of tool-use agents, these models remain systematically overconfident. Our empirical analysis demonstrates that full-trace confidence is less effective for predicting action correctness than action confidence conditioned on earlier reasoning prefixes. Based on this insight, UARI optimizes tool-using LLMs by interrupting reasoning traces to regenerate actions, thereby isolating action-level uncertainty from linguistic noise to jointly improve execution accuracy and model calibration.

AI USE STATEMENT

We used generative AI tools for manuscript editing and review, LaTeX formatting, and citation searches and verification. AI assistance included feedback on methodological descriptions and the presentation and interpretation of reported experimental results. AI tools were not used to generate experimental data or results. The authors are responsible for verifying the AI-assisted material and for the final text, citations, scientific claims, and results.

REFERENCES

Pranjal Aggarwal and Sean Welleck. L1: Controlling how long a reasoning model thinks with reinforcement learning. *arXiv preprint arXiv:2503.04697*, 2025.

Daman Arora and Andrea Zanette. Training language models to reason efficiently. *arXiv preprint arXiv:2502.04463*, 2025.

Kinjal Basu, Ibrahim Abdelaziz, Kiran Kate, Mayank Agarwal, Maxwell Crouse, Yara Rizk, Kelsey Bradford, Asim Munawar, Sadhana Kumaravel, Saurabh Goyal, et al. Nestful: A benchmark for evaluating llms on nested sequences of api calls. In *Proceedings of the 2025 Conference on Empirical Methods in Natural Language Processing*, pp. 33526–33535, 2025.

Farima Fatahi Bayat, Pouya Pezeshkpour, and Estevam Hruschka. From proof to program: Characterizing tool-induced reasoning hallucinations in large language models. *arXiv preprint arXiv:2511.10899*, 2025.

Glenn W. Brier. Verification of forecasts expressed in terms of probability. *Monthly weather review*, 78(1):1–3, 1950.

Yifei Chen, Guanting Dong, and Zhicheng Dou. Et-agent: Incentivizing effective tool-integrated reasoning agent via behavior calibration. *arXiv preprint arXiv:2601.06860*, 2026. doi: 10.48550/arXiv.2601.06860. URL http://arxiv.org/abs/2601.06860.

Mehul Damani, Isha Puri, Stewart Slocum, Idan Shenfeld, Leshem Choshen, Yoon Kim, and Jacob Andreas. Beyond binary rewards: Training lms to reason about their uncertainty. *arXiv preprint arXiv:2507.16806*, 2025. doi: 10.48550/arXiv.2507.16806. URL http://arxiv.org/abs/2507.16806.

Tom Fawcett. An introduction to roc analysis. *Pattern Recognition Letters*, 27(8):861–874, 2006. doi: 10.1016/j.patrec.2005.10.010.

Zihao Feng, Xiaoxue Wang, Bowen Wu, Hailong Cao, Tiejun Zhao, Qun Yu, and Baoxun Wang. Toolsample: Dual dynamic sampling methods with curriculum learning for rl-based tool learning. *arXiv preprint arXiv:2509.14718*, 2025.

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

Hasan Abed Al Kader Hammoud, Hani Itani, and Bernard Ghanem. Beyond the last answer: Your reasoning trace uncovers more than you think. *arXiv preprint arXiv:2504.20708*, 2025.

Jinyi Han, Ying Huang, Ying Liao, Zishang Jiang, Xikun Lu, Haiquan Zhao, Xinyi Wang, Guanghao Zhou, Sihang Jiang, Jiaqing Liang, Weikang Zhou, Zeye Sun, Fei Yu, and Yanghua Xiao. Your models have thought enough: Training large reasoning models to stop overthinking. *arXiv preprint arXiv:2509.23392*, 2025.

Tingxu Han, Zhenting Wang, Chunrong Fang, Shiyu Zhao, Shiqing Ma, and Zhenyu Chen. Token-budget-aware llm reasoning. *arXiv preprint arXiv:2412.18547*, 2024.

Bingguang Hao, Zengzhuang Xu, Maolin Wang, Yuntao Wen, Yicheng Chen, Cunyin Peng, Long Chen, Dong Wang, Xiangyu Zhao, Jinjie Gu, Chenyi Zhuang, and Ji Zhang. Reasoning through exploration: A reinforcement learning framework for robust function calling. *arXiv preprint arXiv:2508.05118*, 2025.

Bairu Hou, Yang Zhang, Jiabao Ji, Yujian Liu, Kaizhi Qian, Jacob Andreas, and Shiyu Chang. Thinkprune: Pruning long chain-of-thought of llms via reinforcement learning. *arXiv preprint arXiv:2504.01296*, 2025.

Gangwei Jiang, Yahui Liu, Zhaoyi Li, Wei Bi, Fuzheng Zhang, Linqi Song, Ying Wei, Defu Lian, Christos Christodoulopoulos, Tanmoy Chakraborty, Carolyn Rose, and Violet Peng. What makes a good reasoning chain? uncovering structural patterns in long chain-of-thought reasoning. In *Proceedings of the 2025 Conference on Empirical Methods in Natural Language Processing*, pp. 6501–6525, 2025.

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

Zhewei Kang, Xuandong Zhao, and Dawn Song. Scalable best-of-n selection for large language models via self-certainty. *arXiv preprint arXiv:2502.18581*, 2025.

Thanh-Long V Le, Myeongho Jeon, Kim Vu, Viet Lai, and Eunho Yang. No prompt left behind: Exploiting zero-variance prompts in llm reinforcement learning via entropy-guided advantage shaping. *arXiv preprint arXiv:2509.21880*, 2025.

Changcheng Li, Jiancan Wu, Hengheng Zhang, Zhengsu Chen, Guo An, Junxiang Qiu, Xiang Wang, and Qi Tian. Confidence before answering: A paradigm shift for efficient llm uncertainty estimation. *arXiv preprint arXiv:2603.05881*, 2026. doi: 10.48550/arXiv.2603.05881. URL http://arxiv.org/abs/2603.05881.

Kuan Li, Xixi Wu, Liwen Zhang, Pengjun Xie, Jingren Zhou, Yong Jiang, Litu Ou, Huifeng Yin, Zhongwang Zhang, Rui Ye, and Zile Qiao. Browseconf: Confidence-guided test-time scaling for web agents. *arXiv preprint arXiv:2510.23458*, 2025. doi: 10.48550/arXiv.2510.23458. URL http://arxiv.org/abs/2510.23458.

Yuliang Liu, Junjie Lu, Zhaoling Chen, Chaofeng Qu, Jason Klein Liu, Chonghan Liu, Zefan Cai, Yunhui Xia, Wei Xiong, Li Zhao, Wei Shen, and Chuheng Zhang. Adaptivestep: Automatically dividing reasoning step through model confidence. *arXiv preprint arXiv:2502.13943*, 2025.

Zhengzhao Ma, Xueru Wen, Boxi Cao, Yaojie Lu, Hongyu Lin, Jinglin Yang, Min He, Xianpei Han, and Le Sun. Decoupling reasoning and confidence: Resurrecting calibration in reinforcement learning from verifiable rewards. *arXiv preprint arXiv:2603.09117*, 2026. doi: 10.48550/arXiv.2603.09117. URL http://arxiv.org/abs/2603.09117.

Minjae Oh, Sangjun Song, Seungkyu Lee, Sungmin Jo, and Yohan Jo. Thinkbrake: Mitigating overthinking in tool reasoning. *arXiv preprint arXiv:2510.00546*, 2025.

Shishir G. Patil, Tianjun Zhang, Xin Wang, and Joseph E. Gonzalez. Gorilla: Large language model connected with massive apis. *arXiv preprint arXiv:2305.15334*, 2023.

Cheng Qian, Emre Can Acikgoz, Qi He, Hongru Wang, Xiusi Chen, Dilek Hakkani-Tür, Gokhan Tur, and Heng Ji. Toolrl: Reward is all tool learning needs. *arXiv preprint arXiv:2504.13958*, 2025.

Yujia Qin, Shihao Liang, Yining Ye, Kunlun Zhu, Lan Yan, Yaxi Lu, Yankai Lin, Xin Cong, Xiangru Tang, Bill Qian, et al. Toolllm: Facilitating large language models to master 16000+ real-world apis. *arXiv preprint arXiv:2307.16789*, 2023.

Tom Schaul, John Quan, Ioannis Antonoglou, and David Silver. Prioritized experience replay. *arXiv preprint arXiv:1511.05952*, 2015.

Timo Schick, Jane Dwivedi-Yu, Roberto Dessi, Roberta Raileanu, Maria Lomeli, Eric Hambro, Luke Zettlemoyer, Nicola Cancedda, and Thomas Scialom. Toolformer: Language models can teach themselves to use tools. *arXiv preprint arXiv:2302.04761*, 2023.

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

Taiwei Shi, Yiyang Wu, Linxin Song, Tianyi Zhou, and Jieyu Zhao. Efficient reinforcement finetuning via adaptive curriculum learning. *arXiv preprint arXiv:2504.05520*, 2025a.

Zhengliang Shi, Shen Gao, Lingyong Yan, Yue Feng, Xiuyi Chen, Zhumin Chen, Dawei Yin, Suzan Verberne, and Zhaochun Ren. Tool learning in the wild: Empowering language models as automatic tool agents. In *Proceedings of the ACM on Web Conference 2025*, pp. 2222–2237, 2025b.

Joykirat Singh, Raghav Magazine, Yash Pandya, and Akshay Nambi. Agentic reasoning and tool integration for llms via reinforcement learning. *arXiv preprint arXiv:2505.01441*, 2025.

Lihao Sun, Hang Dong, Bo Qiao, Qingwei Lin, Dongmei Zhang, and Saravan Rajmohan. Llm reasoning as trajectories: Step-specific representation geometry and correctness signals. *arXiv preprint arXiv:2604.05655*, 2026. doi: 10.48550/arXiv.2604.05655. URL http://arxiv.org/abs/2604.05655.

Amir Taubenfeld, Tom Sheffer, Eran Ofek, Amir Feder, Ariel Goldstein, Zorik Gekhman, and Gal Yona. Confidence improves self-consistency in llms. *arXiv preprint arXiv:2502.06233*, 2025.

Chao Wang, Tao Yang, Hongtao Tian, Yunsheng Shi, Qiyao Ma, Xiaotao Liu, Ting Yao, and Wenbo Ding. Learning more with less: A dynamic dual-level down-sampling framework for efficient policy optimization. *arXiv preprint arXiv:2509.22115*, 2025a.

Shenzhi Wang, Le Yu, Chang Gao, Chujie Zheng, Shixuan Liu, Rui Lu, Kai Dang, Xionghui Chen, Jianxin Yang, Zhenru Zhang, et al. Beyond the 80/20 rule: High-entropy minority tokens drive effective reinforcement learning for llm reasoning. *arXiv preprint arXiv:2506.01939*, 2025b.

Zihan Wang, Chi Gui, Xing Jin, Qineng Wang, Licheng Liu, Kangrui Wang, Shiqi Chen, Linjie Li, Zhengyuan Yang, Pingyue Zhang, et al. Ragen-2: Reasoning collapse in agentic rl. *arXiv preprint arXiv:2604.06268*, 2026.

Jason Wei, Zhiqing Sun, Spencer Papay, Scott McKinney, Jeffrey Han, Isa Fulford, Hyung Won Chung, Alex Tachard Passos, William Fedus, and Amelia Glaese. Browsecomp: A simple yet challenging benchmark for browsing agents. *arXiv preprint arXiv:2504.12516*, 2025. doi: 10.48550/arXiv.2504.12516. URL http://arxiv.org/abs/2504.12516.

Junde Wu, Jiayuan Zhu, Yuyuan Liu, Min Xu, and Yueming Jin. Agentic reasoning: A streamlined framework for enhancing llm reasoning with agentic tools. *arXiv preprint arXiv:2502.04644*, 2025.

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

Weihao Xuan, Heli Qi, Qingcheng Zeng, Naoto Yokoya, Yunze Xiao, and Junjue Wang. The confidence dichotomy: Analyzing and mitigating miscalibration in tool-use agents. *arXiv preprint arXiv:2601.07264*, 2026. doi: 10.48550/arXiv.2601.07264. URL http://arxiv.org/abs/2601.07264.

Fengkai Yang, Zherui Chen, Xiaohan Wang, Xiaodong Lu, Jiajun Chai, Guojun Yin, Wei Lin, Shuai Ma, Fuzhen Zhuang, Deqing Wang, et al. Your group-relative advantage is biased. *arXiv preprint arXiv:2601.08521*, 2026.

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 (ICLR)*, 2023.

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

Zhangyue Yin, Qiushi Sun, Qipeng Guo, Zhiyuan Zeng, Xiaonan Li, Junqi Dai, Qinyuan Cheng, Xuanjing Huang, and Xipeng Qiu. Reasoning in flux: Enhancing large language models reasoning through uncertainty-aware adaptive guidance. In Lun-Wei Ku, Andre Martins, and Vivek Srikumar (eds.), *Proceedings of the 62nd Annual Meeting of the Association for Computational Linguistics (Volume 1: Long Papers)*, pp. 2401–2416, Bangkok, Thailand, August 2024. Association for Computational Linguistics. doi: 10.18653/v1/2024.acl-long.131. URL https://aclanthology.org/2024.acl-long.131/.

Qiying Yu, Zheng Zhang, Ruofei Zhu, Yufeng Yuan, Xiaochen Zuo, Yu Yue, Tiantian Fan, Gaohong Liu, Lingjun Liu, Xin Liu, Haibin Lin, Zhiqi Lin, Bole Ma, Guangming Sheng, Yuxuan Tong, Chi Zhang, Mofan Zhang, Wang Zhang, Hang Zhu, Jinhua Zhu, Jiaze Chen, Jiangjie Chen, Chengyi Wang, Hongli Yu, Weinan Dai, Yuxuan Song, Xiangpeng Wei, Hao Zhou, Jingjing Liu, Wei-Ying Ma, Ya-Qin Zhang, Lin Yan, Mu Qiao, Yonghui Wu, and Mingxuan Wang. DAPO: An open-source LLM reinforcement learning system at scale. *arXiv preprint arXiv:2503.14476*, 2025.

Xingshan Zeng, Weiwen Liu, Lingzhi Wang, Liangyou Li, Fei Mi, Yasheng Wang, Lifeng Shang, Xin Jiang, and Qun Liu. Toolace-mt: Non-autoregressive generation for agentic multi-turn interaction. *arXiv preprint arXiv:2508.12685*, 2025.

Boxuan Zhang and Ruqi Zhang. Cot-uq: Improving response-wise uncertainty quantification in llms with chain-of-thought. In *Findings of the Association for Computational Linguistics: ACL 2025*, pp. 26114–26133, 2025.

Jiaxin Zhang. Confidence-aware reasoning: Optimizing self-guided thinking trajectories in large reasoning models. In Saloni Potdar, Lina Rojas-Barahona, and Sebastien Montella (eds.), *Proceedings of the 2025 Conference on Empirical Methods in Natural Language Processing: Industry Track*, pp. 2081–2095, Suzhou (China), November 2025. Association for Computational Linguistics. ISBN 979-8-89176-333-3. doi: 10.18653/v1/2025.emnlp-industry.146. URL https://aclanthology.org/2025.emnlp-industry.146/.

Jiaxin Zhang, Caiming Xiong, and Chien-Sheng Wu. Agentic confidence calibration. *arXiv preprint arXiv:2601.15778*, 2026.

Tunyu Zhang, Haizhou Shi, Yibin Wang, Hengyi Wang, Xiaoxiao He, Zhuowei Li, Haoxian Chen, Ligong Han, Kai Xu, Huan Zhang, et al. Tokur: Token-level uncertainty estimation for large language mode reasoning. In *First Workshop on Foundations of Reasoning in Language Models*.

Xichen Zhang, Sitong Wu, Yinghao Zhu, Haoru Tan, Shaozuo Yu, Ziyi He, and Jiaya Jia. Scaf-grpo: Scaffolded group relative policy optimization for enhancing llm reasoning. *arXiv preprint arXiv:2510.19807*, 2025.

Yida Zhao, Kuan Li, Xixi Wu, Liwen Zhang, Dingchu Zhang, Baixuan Li, Maojia Song, Zhuo Chen, Chenxi Wang, Xinyu Wang, et al. Repurposing synthetic data for fine-grained search agent supervision. *arXiv preprint arXiv:2510.24694*, 2025.

Zhanke Zhou, Zhaocheng Zhu, Xuan Li, Mikhail Galkin, Xiao Feng, Sanmi Koyejo, Jian Tang, and Bo Han. Landscape of thoughts: Visualizing the reasoning process of large language models. *arXiv preprint arXiv:2503.22165*, 2026. doi: 10.48550/arXiv.2503.22165. URL http://arxiv.org/abs/2503.22165.

A Additional Backbone Evaluation

Table 6 reports BFCL-v4 Single-Turn accuracy on two additional model families. For Llama, Single-Turn accuracy is the mean of Non-Live AST and Live AST in Table 1. Granite is evaluated against its base model; a matched Granite Plain GRPO result is not available in this comparison.

Table 6: BFCL-v4 Single-Turn accuracy (%) on additional backbones. Signed gains denote absolute differences from the corresponding base model.

Backbone	Base	UARI	Gain
Granite-3.3-2B-Instruct	32.68	35.76	+3.08
Llama-3.2-3B-Instruct	70.47	72.97	+2.50

B Motivation: Diagnostic Experiments for Uncertainty Signal Selection

Sections 1 and 2.2 argue that full-trace confidence C_{trace} can be diluted by generic reasoning tokens, whereas action correctness in tool calling depends on sparse function and argument tokens. The diagnostics below test this premise from two angles: the step-swap study checks whether C_{trace} changes with local narrative form even when action correctness is stable, and the residual study checks whether C_{trace} contains useful action-relevant information beyond interrupted action confidence $\hat{C}_{\text{act}}$. Together, these results motivate using $\hat{C}_{\text{act}}$ as the main uncertainty signal in UARI.

B.1 Step-Swap Perturbation: Trace Confidence Decoupling from Action Correctness

To directly test whether trace confidence is overly sensitive to local narrative form, we run a controlled step-swap intervention. This diagnostic uses the completed-trace instance of the Section 2.2 notation: *full-trace confidence* refers to $C_{\text{trace}}(y^{\text{reason}}, y^{\text{act}})$ from Eq. 3, i.e. the trace score on the original uninterrupted reasoning segment and its tool call. This is distinct from the prefix-specific reconstructed score $C_{\text{trace}}(\rho) = C_{\text{trace}}(p_\rho, \hat{a}_\rho)$ used in the interruption diagnostic, but it tests the same failure mode: whether confidence attached to reasoning tokens changes with local narrative form beyond what action correctness warrants.

Given a completed reasoning trace $y^{\text{reason}} = (s_1, \ldots, s_Q)$ segmented into sentence-level steps, we uniformly sample a pair of adjacent steps (s_k, s_{k+1}) and swap their order while keeping the reasoning token content fixed. For each modified reasoning trace $\tilde{y}^{\text{reason}}$, we regenerate the action $\tilde{a} \sim \pi_\theta(\cdot \mid x, \tilde{y}^{\text{reason}})$ and measure: (i) $\Delta_c = C_{\text{trace}}(y^{\text{reason}}, y^{\text{act}}) - C_{\text{trace}}(\tilde{y}^{\text{reason}}, \tilde{a})$, the signed trace-confidence shift; and (ii) Δ_a, the *binary* change in action correctness (exact tool-call match: 0 or 1), computed by checking $\tilde{a}$ against the reference $a^\star$. We report mean absolute shifts $\bar{\Delta}_c = \mathbb{E}[|\Delta_c|]$ and $\bar{\Delta}_a = \mathbb{E}[|\Delta_a|]$ over 500 swap trials per task, and define the coupling ratio $\kappa = \bar{\Delta}_a / \bar{\Delta}_c$. Because $\bar{\Delta}_c$ is a continuous log-probability shift and $\bar{\Delta}_a$ is a binary rate change, κ is not scale-invariant; we therefore focus on the direction and magnitude of the contrast rather than the absolute value of κ.

Table 7: Step-swap perturbation: mean shifts in C_{trace} ($\bar{\Delta}_c$, Eq. 3) and action correctness ($\bar{\Delta}_a$) after swapping two adjacent reasoning steps. $\kappa = \bar{\Delta}_a / \bar{\Delta}_c$ is the coupling ratio.

Task	$\bar{\Delta}_c$	$\bar{\Delta}_a$	$\kappa = \bar{\Delta}_a / \bar{\Delta}_c$
Math (GSM8K)	0.198	0.322	1.628
Tool calling (BFCL)	1.139	0.015	0.013
Ratio ($\kappa_{\text{math}} / \kappa_{\text{tool}}$)			$\approx 124\times$

Table 7 shows a sharp contrast between causal-chain reasoning and tool calling. In GSM8K, perturbing the rationale changes confidence and correctness together ($\kappa = 1.63$), consistent with a setting where intermediate steps carry the answer forward. In tool calling, the same perturbation sharply changes C_{trace} ($\bar{\Delta}_c = 1.139$) while leaving action correctness nearly unchanged ($\bar{\Delta}_a = 0.015$), yielding a $\approx 124\times$ weaker coupling. This does not identify a single causal mechanism, and the

coupling ratio κ is scale-sensitive—$\bar{\Delta}_c$ is a continuous log-probability shift while $\bar{\Delta}_a$ is a binary correctness rate change. Taken individually, the BFCL confidence shift is notably larger than in GSM8K (1.139 vs. 0.198) while the correctness shift is negligible (0.015 vs. 0.322). This supports the Section 2.2 claim that trace-level confidence in tool calling can vary with local narrative ordering even when executable action correctness is stable; it is therefore better used as a diagnostic baseline than as the primary optimization signal.

B.2 Residual Diagnostic: Trace Residual Provides Limited Incremental Signal

The step-swap study shows that trace confidence can move with local narrative form even when action correctness remains nearly fixed. The remaining question is whether this extra trace-level variation is nevertheless useful for predicting action correctness. This diagnostic tests the stronger alternative hypothesis: after accounting for reasoning-interrupted action confidence, the residual part of reconstructed trace confidence still contains action-relevant information. If this hypothesis were true, adding the residual to $\hat{C}_{\mathrm{act}}$ should improve held-out AUROC for predicting whether the regenerated action is correct.

For each useful prefix ratio $\rho \in \{0.1, \ldots, 0.8\}$, we use the existing BFCL single-turn segment-truncation records to obtain the action confidence $\hat{C}_{\mathrm{act}}(\rho) = \hat{C}_{\mathrm{act}}(\hat{a}_\rho \mid p_\rho)$, the reconstructed trace confidence $C_{\mathrm{trace}}(\rho) = C_{\mathrm{trace}}(p_\rho, \hat{a}_\rho)$, and the binary action-correctness label $Y_\rho = \mathbf{1}[\hat{a}_\rho \text{ exactly matches } a^\star]$. The trace confidence in this diagnostic is therefore the confidence of the reconstructed prefix-plus-action trace at that prefix depth, not the fixed realized-trace score $\mathrm{C}(y^{\mathrm{reason}}, y^{\mathrm{act}})$ used in Appendix B.1. We measure the *linear incremental predictive value* of the trace residual after partialling out reasoning-interrupted action confidence; this is not a causal identification test, and linear residualization is only a limited proxy for the full conditional relationship. We split examples deterministically by hashing the example identifier: 80% are used to fit

$$C_{\mathrm{trace}}(\rho) = \beta_0 + \alpha_\rho \hat{C}_{\mathrm{act}}(\rho) + Z_\rho,$$

and the held-out 20% are used for all reported residual metrics. Z_ρ is the OLS residual from the above regression (i.e., $Z_\rho = C_{\mathrm{trace}}(\rho) - \hat{\beta}_0 - \hat{\alpha}_\rho \hat{C}_{\mathrm{act}}(\rho)$, where $\hat{\beta}_0$ and $\hat{\alpha}_\rho$ are fit on the 80% split). The ΔAUROC column is the held-out AUROC of a logistic model using $(\hat{C}_{\mathrm{act}}(\rho), Z_\rho)$ as features predicting Y_ρ, minus the AUROC of a logistic model using $\hat{C}_{\mathrm{act}}(\rho)$ alone; both logistic models are also fit on the 80% split and evaluated on the 20% held-out set.

Table 8: Full residual diagnostic statistics. $\Delta\mathrm{AUROC} = \mathrm{AUROC}(C_{\mathrm{act}}, Z_\rho) - \mathrm{AUROC}(C_{\mathrm{act}})$: incremental AUROC from adding the residual Z_ρ to $\hat{C}_{\mathrm{act}}(\rho)$ in held-out logistic evaluation.

Prefix	AUC($\hat{C}_{\mathrm{act}}$)	AUC(C_{trace})	AUC(Z_ρ)	Corr(Z_ρ, Y)	Corr(Z_ρ, C_{act})	ΔAUROC
10%	0.8018	0.6399	0.5827	0.1446	0.0112	-0.1266
20%	0.7755	0.6586	0.5988	0.1755	-0.0589	-0.0966
30%	0.7545	0.6718	0.6131	0.1946	-0.0194	-0.0636
40%	0.7688	0.6831	0.6454	0.2771	0.0521	-0.0466
50%	0.7826	0.6768	0.6375	0.2690	0.0007	-0.0729
60%	0.7461	0.7330	0.6961	0.3645	0.0073	+0.0005
70%	0.7354	0.6984	0.6768	0.3113	0.1709	-0.0156
80%	0.7208	0.7215	0.6862	0.3323	0.0490	+0.0164

Table 8 shows that the residual does not provide the predicted gain: ΔAUROC is non-positive or near zero for most prefix ratios and reaches only $+0.0164$ at the largest retained prefix. Thus, within this linear held-out diagnostic, the extra variation in reconstructed trace confidence beyond $\hat{C}_{\mathrm{act}}(\rho)$ behaves more like narrative noise than robust action-relevant signal. This supports the Section 2.2 claim that the action-conditioned signal captures most of the linearly extractable predictive information in reconstructed trace confidence, while avoiding the extra narrative variation measured by the step-swap diagnostic.

Algorithm 1 UARI

Require: Context x; gold action $a^\star$; old policy $\pi_{\theta_{\text{old}}}$; group size K.
1: Sample four complete source rollouts from $\pi_{\theta_{\text{old}}}(\cdot \mid x)$.
2: $\{p^{(n)}\}_{n=1}^{N_x} \leftarrow$ INTERRUPTEDPREFIXES(source rollouts) ▷ prefixes across all four rollouts; §3.1
3: **for** $n = 1, \ldots, N_x$ **do**
4: $\quad \hat{a}^{(n)} \leftarrow$ SAMPLEACTION$(\pi_{\theta_{\text{old}}}, x, p^{(n)})$
5: $\quad \hat{C}_{\text{act}}^{(n)} \leftarrow \hat{C}_{\text{act}}(\hat{a}^{(n)} \mid p^{(n)})$ ▷ reasoning-interrupted action confidence
6: $\quad R^{(n)} \leftarrow s(\hat{a}^{(n)}, a^\star)$ ▷ partial-credit correctness reward
7: **end for**
8: Form the pool from $(p^{(n)}, \hat{a}^{(n)})$ only; exclude the complete source rollouts.
9: $M \sim \mathcal{U}\{\min(K{+}1, N_x), \min(2K, N_x)\}$
10: $\mathcal{W}_x \leftarrow$ MEDIANWINDOW$\big(\{1, \ldots, N_x\},\ \{\hat{C}_{\text{act}}^{(n)}\}_{n=1}^{N_x},\ M\big)$
11: **if** $|\mathcal{W}_x| < K$ **then**
12: $\quad G_x \leftarrow$ SAMPLEWITHREPLACEMENT$(\mathcal{W}_x, K)$
13: **else**
14: $\quad G_x \leftarrow \text{first-argmax}_{G \subset \mathcal{W}_x,\, |G|=K} \operatorname{Var}\big(\{R^{(n)} : n \in G\}\big)$ ▷ reward-variance subset; §3.2
15: **end if**
16: $A_d \leftarrow$ JOINTADVANTAGE$\big(\{R^{(i)}\}_{i \in G_x},\ \{\hat{C}_{\text{act}}^{(i)}\}_{i \in G_x}\big)$ ▷ Eq. 6
17: Update θ on $\{(p^{(i)}, \hat{a}^{(i)}) : i \in G_x\}$ via the GRPO objective (Eq. 2) with advantage A_d and KL penalty toward $\pi_{\theta_{\text{old}}}$.

C UARI ALGORITHM DETAILS

D EVALUATION METRICS

To rigorously evaluate the performance of our calibration framework, we focus on the following two confidence metrics. Let c_i be the predicted confidence and $y_i \in \{0, 1\}$ be the ground-truth success label for trajectory i over N samples.

Area under the receiver operating characteristic curve (AUROC). AUROC evaluates how well a confidence score ranks correct tool calls above incorrect ones (Fawcett, 2006). Given binary correctness labels, varying a decision threshold over the confidence scores induces a true positive rate (TPR) and a false positive rate (FPR). The AUROC is the area under this ROC curve:

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

where

$$\text{TPR} = \frac{\text{TP}}{\text{TP} + \text{FN}}, \qquad \text{FPR} = \frac{\text{FP}}{\text{FP} + \text{TN}}.$$

A higher AUROC indicates that the confidence signal more reliably assigns larger scores to successful tool calls than to failed ones.

Brier score. The Brier Score is a proper scoring rule that measures the mean squared error between predicted probabilities and actual outcomes (Brier, 1950). Since our confidence scores are mean log-probabilities, we convert each raw score C_i to a probability estimate $c_i = \exp(C_i)$ before computing Brier. The score simultaneously assesses both calibration and discrimination, and is defined as:

$$\text{Brier Score} = \frac{1}{N} \sum_{i=1}^{N} (c_i - y_i)^2.$$

A lower Brier Score indicates better overall prediction quality. For readability, the Brier values in the main tables are multiplied by 100.

E SYSTEM PROMPT

System prompt. The Qwen3 experiments use its official Jinja-based chat template; the prompt below describes this backbone setting. We retain Qwen3's native `#tool` prompt, which standardizes

tool-use formatting by (i) providing available function signatures inside <tools> tags and (ii) requiring the agent to emit each invocation as a JSON object wrapped by <tool_call> tags. If no provided function is applicable, or if required arguments are missing, the agent is instructed to explicitly state the limitation. The dialogue history starts from the initial user query (serialized in <user>), and is extended across steps by appending the agent's intermediate reasoning <think> and the predicted action <tool_call>.

System Prompt

 # System
You are an expert in composing functions. You are given a question and a set of possible functions. Based on the question, you will need to make one or more function/tool calls to achieve the purpose. If none of the functions can be used, point it out. If the given question lacks the parameters required by the function, also point it out.
 # Tools
You may call one or more functions to assist with the user query. You are provided with function signatures within <tools></tools> XML tags:
<tools>
....
</tools>
For each function call, return a json object with function name and arguments within <tool_call></tool_call> XML tags:
<tool_call>
{"name": <function-name>, "arguments": <args-json-object>}
</tool_call>

F Training Details

F.1 Cut-Point Selection: Implementation Details

Section 3.1 uses entropy-based prefix proposals along a reasoning segment of length T_r. Our rollout engine exposes per-token distribution entropy h_t, which supplies the proposal locations. Distribution entropy and the log-probability of the realized token are distinct quantities; the detector does not require their extrema to coincide. Cut-point selection determines the available prefixes, while action regeneration, interrupted-action scoring, group selection, and the shaping objective follow the same UARI procedure. After truncation, we append the explicit closing tag </think> to the retained reasoning segment and feed the reconstructed prefix back into the model so that the subsequent generation starts directly from action prediction. A switch cut_scheme selects the default entropy pipeline used in all reported UARI results, or a deterministic ratio pipeline used only for the interruption-strategy ablation (§4.4). Numeric values for all hyperparameters are listed in Table 9.

Entropy-driven cut (default). Cut selection on $h \in \mathbb{R}^{T_r}$ first retains high-entropy outliers under Tukey's interquartile rule, $h_t > q_3 + \tau\,\mathrm{IQR}$, merges nearby peaks, and adapts the clustering distance to keep the candidate count within a target range. We then keep only peaks above the 80th percentile of h, following the observation that the highest-entropy tokens concentrate much of the useful branching signal in reasoning rollouts (Wang et al., 2025b). Candidate endpoints are moved slightly forward to align with the trailing log-probability valley and restricted to $[s_{\min}, e_{\max}]$, where $s_{\min} = \max(\lfloor r_s T_r \rfloor, 100)$, $r_s \sim \mathcal{U}(0.15, 0.25)$, and $e_{\max} = T_r - \lfloor 0.20 T_r \rfloor$. Each endpoint is snapped to the next sentence boundary, and candidates shorter than $\min(50, \lfloor T_r/10 \rfloor)$ tokens are discarded.

Fixed-ratio cut (ablation only). The ablation pipeline bypasses entropy entirely and places deterministic prefix endpoints at $e_r = \lfloor r T_r \rfloor$ for $r \in \{0.2, 0.4, 0.6, 0.8\}$. The scheme is content-agnostic by design: it isolates the contribution of *where* cuts are placed (entropy-informed vs. uniform) from all other components of UARI, and serves only as the baseline in §4.4.

Table 9: Cut-related configuration.

Parameter	Value (current implementation)
`cut_scheme`	`entropy` (default), `ratio` (optional)
IQR peak threshold τ	1.5
Target peak count range	[3, 12]
Absolute entropy threshold	80th percentile
Start ratio range	[0.15, 0.25] (uniform random)
End buffer ratio	0.20
Minimum segment distance	$\min(50, \lfloor T_r/10 \rfloor)$
Ratio endpoints (`ratio` mode)	0.2, 0.4, 0.6, 0.8

F.2 Reinforcement Learning

All reinforcement learning experiments are implemented with `veRL` and `SGLang`. As described in Section 4.1, the training pool is first filtered by sampling 20 rollouts per instance and removing instances whose rollouts all succeed. This filtering removes prompts whose sampled responses provide no reward contrast, yielding the 2.3k training instances used by all RL methods. During RL optimization, each update samples a batch of 256 prompts. UARI generates four complete source rollouts per prompt solely to obtain interrupted reasoning prefixes. Actions are regenerated from those prefixes and pooled across the four source rollouts. Only these regenerated candidates enter the pool; the original complete rollouts are excluded. Six update candidates are selected from this pool using the procedure in Section 3.2. To keep the GRPO group size matched across training methods, Plain GRPO and RLCR directly generate 6 rollouts per prompt. RLCR, CAR, and DCPO are reproduced following their respective original papers. All methods share the same optimizer, sequence-length limits, KL regularization, and training schedule in our comparison, while retaining their method-specific objectives; the common hyperparameters are reported in Table 10.

Table 10: Hyperparameter configuration for RL training.

Category	**Value**
Data Configuration	
Train batch size	256
Validation batch size	128
Max prompt length	4096
Max response length	8096
Optimization	
Learning rate	1×10^{-6}
PPO mini-batch size	32
KL loss	Enabled
KL coefficient	0.001
Rollout Configuration	
Rollout engine	SGLang
GPU memory utilization	0.45
Source rollouts per prompt	4 (UARI), 6 (Plain GRPO/RLCR)
Filtering rollouts per instance	20
UARI Configuration	
Group size K	6
Candidate pool	Interrupted-prefix regenerated candidates only
Confidence weight λ	0.05
Cut scheme	Entropy-driven
Confidence window	Median-centered, $M \in [\min(K{+}1, N_x), \min(2K, N_x)]$
Training & Logging	
Total epochs	15
Policy loss	GRPO

F.3 Inference-Time Method Evaluation

For the evaluation of inference-time methods on BFCL-v4, we follow the official BFCL evaluation protocol and use its standard generation settings: temperature $= 0.6$, top-$p = 0.95$, top-$k = 20$, and a maximum generation length of 4096 tokens. For the corresponding Nestful evaluation, we use temperature $= 0.0001$ with one in-context example (`icl_count`=1) and the same maximum generation length of 4096 tokens.