ADAPTIVE UNCERTAINTY-AWARE TREE SEARCH FOR ROBUST REASONING

Anonymous authors
Paper under double-blind review

ABSTRACT

Inference-time reasoning scaling has significantly advanced the capabilities of Large Language Models (LLMs) in complex problem-solving. A prevalent approach involves external search guided by Process Reward Models (PRMs). However, a fundamental limitation of this framework is the epistemic uncertainty of PRMs when evaluating reasoning paths that deviate from their training distribution. In this work, we conduct a systematic analysis of this challenge. We first provide empirical evidence that PRMs exhibit high uncertainty and unreliable scoring on out-of-distribution (OOD) samples. We then establish a theoretical framework proving that while standard search incurs linear regret accumulation, an uncertainty-aware strategy can achieve sublinear regret. Motivated by these findings, we propose Uncertainty-Aware Tree Search (UATS), a unified method that estimates uncertainty via Monte Carlo Dropout and dynamically allocates compute budget using a reinforcement learning-based controller. Extensive experiments demonstrate that our approach effectively mitigates the impact of OOD errors.

1 INTRODUCTION

In recent years, chain-of-thought (CoT) prompting has substantially improved the performance of large language models (LLMs) on complex reasoning tasks such as math problem solving, question answering, and multi-hop retrieval Wei et al. (2022); Kojima et al. (2022); Yao et al. (2023). Beyond prompting alone, recent studies have shown that allocating additional computation at inference time can further enhance reasoning performance OpenAI (2024); DeepSeek-AI (2025); Team et al. (2025); Snell et al. (2024); Liu et al. (2025); Wu et al. (2024). These approaches can be broadly divided into internal and external ones, with external methods attracting increasing attention due to their ability to improve reasoning performance while keeping the policy model frozen Snell et al. (2024); Liu et al. (2025); Wu et al. (2024).

A prominent line of external reasoning methods relies on a frozen policy model, a search procedure, and an external verifier such as a process reward model (PRM). In PRM-guided search, the policy model proposes candidate reasoning traces, while the PRM provides step-level scores for ranking, pruning, and expansion decisions Yao et al. (2023); Besta et al. (2024). However, test-time search can expose the PRM to reasoning traces beyond its training distribution, which we refer to as PRM-perspective out-of-distribution (OOD) traces. For such traces, the PRM's scalar scores may be unreliable due to high epistemic uncertainty; relying solely on them can lead the search to over-trust incorrect evaluations, prematurely prune promising branches, or allocate inference-time budget to misleading candidates. This raises a key question: *How can external reasoning remain reliable when the verifier is uncertain about test-time reasoning traces?*

We first study this question empirically and theoretically. Empirically, we evaluate representative open-source PRMs under policy-induced distribution shifts and observe that PRM accuracy drops on OOD reasoning traces, accompanied by increased Monte Carlo Dropout variance. Figure 1 illustrates a representative failure case where naive PRM ranking selects an incorrect but high-scoring reasoning step, while its high uncertainty reveals the unreliability of the score. Theoretically, we analyze a stylized search model and show that uncertainty-agnostic greedy selection can suffer linear degradation when OOD candidates appear, whereas uncertainty-aware selection admits a sublinear degradation bound under explicit assumptions.

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.

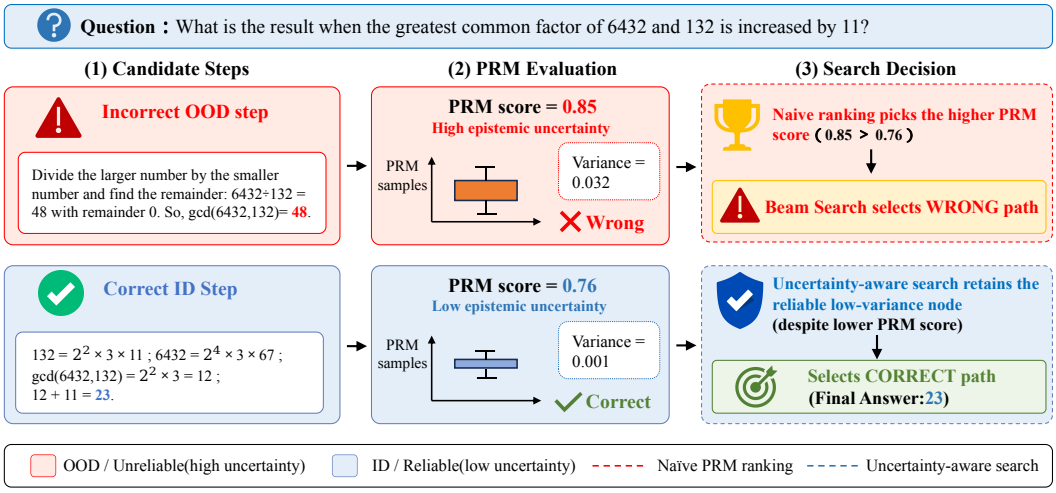


Figure 1: Visualization of a real beam search step on the MATH dataset. The diagram depicts a selection conflict where the PRM assigns a higher reward (0.85) to an incorrect reasoning step containing a calculation error than to a correct step (0.76). Despite the high reward, the epistemic uncertainty (visualized via boxplots) reveals the unreliability of the incorrect node, which exhibits significantly higher score variance ($\sigma^2 = 0.032$) compared to the correct node ($\sigma^2 = 0.001$).

Motivated by the theoretical insights, we propose Uncertainty-Aware Tree Search (UATS), a framework for robust PRM-guided external reasoning. Our method first estimates epistemic uncertainty via Monte Carlo Dropout. We then introduce a heuristic strategy (H-UATS) that leverages this uncertainty to selectively re-evaluate promising but uncertain candidates. To further enhance adaptability, we formalize the search budget allocation as a Markov Decision Process (MDP) and train an adaptive controller using reinforcement learning. This controller dynamically modulates key search hyperparameters under resource constraints.

We evaluate UATS on MATH-500 Hendrycks et al. (2021); Lightman et al. (2023) and AIME24 AI-MO (2024) with multiple frozen policy models, including Qwen2.5 Yang et al. (2024a), Llama 3.1 Grattafiori et al. (2024), and Llama 3.2 Meta AI (2024), as well as several representative PRMs Zhang et al. (2025); Wang et al. (2024); Skywork o1 Team (2024). Across different policy–PRM combinations and compute budgets, UATS consistently improves over standard external reasoning baselines in all evaluated settings.

Our contributions are summarized as follows: 1) We identify the issue of epistemic uncertainty in PRM-based external reasoning, revealing that distribution shifts between the policy and PRM undermine search reliability. 2) We provide empirical evidence of PRM degradation on OOD data and theoretical analysis showing that uncertainty-aware search can reduce variance-induced degradation from linear to sublinear. 3) We propose UATS, integrating uncertainty estimation with a learnable budget controller that consistently outperforms standard baselines across evaluated settings.

2 Related Works

Inference-time Reasoning. CoT prompting is first proposed as a prompting technique that enables LLMs to decompose problems into intermediate steps Wei et al. (2022). Recently, the OpenAI o1 series OpenAI (2024) demonstrate that increasing the length of CoT during inference yields substantial performance gains on tasks like MATH Hendrycks et al. (2021) and AIME AI-MO (2024). External reasoning improves the reasoning performance via sampling or search-based methods with fixed LLMs and an external verifier Lightman et al. (2023); Wu et al. (2024); Snell et al. (2024); Yao et al. (2023); Sel et al. (2024); Besta et al. (2024); Zhang et al. (2022); Brown et al. (2024). Specifically, Tree-of-Thoughts Yao et al. (2023) explores a look-ahead search tree of thought chunks with self-evaluation to achieve large gains on planning-style tasks. Snell et al. Snell et al. (2024) find

that adaptive allocation of verifier-guided search can beat a 14× larger model while using 4× less extra compute than best-of-N sampling.

Process Reward Model. An essential component of external reasoning is the verifier that evaluates different reasoning paths. Verifiers are categorized into two types: Process Reward Models (PRMs) and Outcome Reward Models (ORMs) Uesato et al. (2022). PRMs assess the quality of a reasoning step given the question and partial reasoning trajectory, estimating the likelihood that the process will lead to a correct answer Lightman et al. (2023); Wang et al. (2024). In contrast, ORMs provide a reward signal based on the final answer's correctness, given the full reasoning trace and output Uesato et al. (2022); Lightman et al. (2023). Recent studies have shown that PRMs are generally more effective than ORMs in guiding search Lightman et al. (2023); Uesato et al. (2022); Snell et al. (2024), and PRMs have become a widely adopted tool in external test-time reasoning frameworks Liu et al. (2025); Xie et al. (2023); Snell et al. (2024). Lightman et al. Lightman et al. (2023) train PRMs with human-labeled step-level supervision. Math-Shepherd Wang et al. (2024) automatically constructs step-level supervision and assigns scores based on the proportion of branches that reach the known correct answer.

3 PRELIMINARY

In this section, we first formally define the fundamental workflow of problem-solving via external reasoning. Subsequently, we briefly introduce the training paradigm of PRMs. Finally, we analyze the uncertainty of PRM outputs from a Bayesian perspective.

3.1 PROBLEM SOLVING WITH EXTERNAL REASONING

We consider reasoning problems defined in natural language with fixed answers, where each question $q \in \mathcal{Q}$ is associated with a ground truth answer $a^* \in \mathcal{A}$. These elements are represented as sequences of tokens; specifically, $\mathcal{Q}, \mathcal{A} \subset \mathcal{V}^*$, where $\mathcal{V}$ denotes the token space and $\mathcal{V}^* \triangleq \bigcup_{L=0}^{\infty} \mathcal{V}^L$ represents the set of all possible token sequences. We assume that the question-answer pairs (q, a^*) follow an unknown joint distribution $\mathcal{P}_{\text{data}}(q, a^*)$.

In general, these problems are difficult to answer directly and instead require a step-by-step reasoning process Wei et al. (2022); Lightman et al. (2023); OpenAI (2024). Under this situation, an external reasoning system is designed to address this challenge. Specifically, the system consists of the following components: (i) an LLM $\pi_\theta : \mathcal{V}^* \to \Delta(\mathcal{V})$ with fixed parameters θ is used as the proposer to generate partial reasoning traces $h_k = (q, z_1, \ldots, z_k)$ in an autoregressive manner, where each $z_k \in \mathcal{V}^*$ is a reasoning step and $\Delta(\mathcal{V})$ denotes the probability simplex over $\mathcal{V}$; (ii) a PRM $R_\phi : \mathcal{V}^* \to [0, 1]$ with fixed parameters ϕ is used as the verifier to evaluate the reasoning traces h_k generated by π_θ; (iii) a search algorithm $\mathrm{Alg}(\cdot; \pi_\theta, R_\phi, C) : \mathcal{Q} \to \mathcal{V}^*$ leverages the traces proposed by π_θ, the scores provided by R_ϕ, and a computation budget $C \in \mathbb{N}^+$ to guide, filter, and ultimately output a selected reasoning trace $\hat{h}$; and (iv) an answer extractor $\mathcal{E} : \mathcal{V}^* \to \mathcal{A}$ extracts the final answer from the selected reasoning trace, namely $\hat{a} = \mathcal{E}(\hat{h})$.

In this context, existing research primarily focuses on designing search algorithms that maximize the accuracy of the final answer Snell et al. (2024); Wu et al. (2024). Formally, given a dataset $\mathcal{D}_{\text{tr}} = \{(q_i, a_i^*)\}_{i=1}^{N_{\text{train}}}$ where the samples are drawn from the distribution $\mathcal{P}_{\text{data}}(q, a^*)$, the goal is to find a search algorithm that maximizes the expected accuracy. Consequently, the objective function is defined as:

$$\max_{\text{Alg}} \mathbb{E}_{(q,a^*)\sim\mathcal{P}_{\text{data}}} \left[\mathbb{1}_{\left(\mathcal{E}\left(\mathrm{Alg}(q;\pi_\theta,R_\phi,C)\right)=a^*\right)} \right], \tag{1}$$

where the mathematical term $\mathbb{1}_{(\cdot)}$ denotes the indicator function that equals 1 if the extracted answer matches the ground-truth answer a^*, and 0 otherwise.

3.2 PRM: PROCESS REWARD MODEL

The PRM R_ϕ plays a central role in the external reasoning system. A PRM is typically implemented by appending a linear prediction head to another LLM (different from π_θ) and then fine-tuning the entire network on supervised training data Uesato et al. (2022); Lightman et al. (2023); Zhang

et al. (2025); Wang et al. (2024). The dataset $\mathcal{D}_{\mathrm{PRM}} = \{(q_i, \{(z_{i,t}, y_{h_{i,t}})\}_{t=1}^{T_i})\}_{i=1}^{n}$ for training PRM consists of n questions and corresponding reasoning steps $\{z_{i,t}\}_{t=1}^{T_i}$, where q_i denotes the i-th question, $z_{i,t}$ represents the t-th reasoning step for question q_i, $h_{i,t} = (q_i, z_{i,1}, \ldots, z_{i,t})$ is the reasoning trace and $y_{h_{i,t}} \in \{0, 1\}$ is the quality label for $h_{i,t}$, with 1 indicating "good" and 0 indicating "bad". The label collection process can be found in Lightman et al. (2023); Wang et al. (2024). Given the above training data, the PRM training objective is formulated as:

$$\min_{\phi} \sum_{i=1}^{N} \sum_{t=1}^{T_i} \Big(y_{h_{i,t}} \log R_\phi(h_{i,t}) + (1 - y_{h_{i,t}}) \log \big(1 - R_\phi(h_{i,t})\big) \Big) \tag{2}$$

After training, the parameters of PRM are frozen. Within external reasoning, the trained PRM is expected to assign reliable scores to the reasoning steps produced by the frozen policy π_θ when confronted with unseen questions.

3.3 Prediction Uncertainty in PRMs

The PRM score directly affects trace selection and thus the final performance of external reasoning systems under Equation 1. Since the PRM is trained on a finite dataset $\mathcal{D}_{\mathrm{PRM}}$, we view its parameters from a Bayesian perspective as following a posterior distribution $p(\phi \mid \mathcal{D}_{\mathrm{PRM}})$. The predictive uncertainty of a PRM on a reasoning trace h can then be decomposed by the law of total variance:

$$\mathrm{Var}(y \mid h, \mathcal{D}_{\mathrm{PRM}}) = \mathbb{E}_{\phi \sim p(\phi \mid \mathcal{D}_{\mathrm{PRM}})} \big[\mathrm{Var}(y \mid h, \phi)\big] + \mathrm{Var}_{\phi \sim p(\phi \mid \mathcal{D}_{\mathrm{PRM}})} \big(\mathbb{E}[y \mid h, \phi]\big). \tag{3}$$

The first term is *aleatoric uncertainty*, which reflects the intrinsic ambiguity of a partial trace h, since it may be extended into different future reasoning paths and final answers. The second term is *epistemic uncertainty*, which reflects disagreement among plausible PRM parameters. When h is well covered by $\mathcal{D}_{\mathrm{PRM}}$, different parameter samples are constrained to give similar predictions. When h deviates from the PRM training distribution, this constraint weakens, and the predictions can vary substantially. Thus, epistemic uncertainty provides a useful signal for detecting PRM-perspective OOD reasoning traces Kochenderfer et al. (2022).

This motivates two questions: how reliable are existing PRMs on reasoning traces beyond their training distribution, and how does PRM uncertainty affect external reasoning performance?

4 Problem Analysis

In this section, we analyze the impact of PRM uncertainty on external reasoning. We first show empirically that PRMs become unreliable on OOD reasoning steps, and then provide a theoretical analysis showing that uncertainty-agnostic search accumulates regret, while uncertainty-aware search can provably reduce it.

4.1 Empirical Analysis

To verify our hypothesis regarding PRM behavior under distribution shifts, we investigate whether existing PRMs remain reliable when evaluating reasoning traces generated by policy models whose output distributions differ from those seen during PRM training.

We select two representative open-source PRMs: Math-Shepherd-PRM-7B Wang et al. (2024) and Qwen2.5-Math-PRM-7B Yang et al. (2024b). For the policy models, we employ LLemma-7B and Qwen2.5-Instruct-7B, yielding PRM–Policy pairs with different degrees of distributional mismatch. The evaluation is conducted on the full MATH-500 test subset Hendrycks et al. (2021); Lightman et al. (2023). We use beam search with beam width 4 and retain 8 final candidate answers. We record two primary metrics: (1) **Accuracy**, averaged over 10 independent runs for each PRM–Policy pair; and (2) **Score Variance**, computed as the variance of PRM scores over Monte Carlo Dropout passes on the same reasoning step, serving as a proxy for verifier-side epistemic uncertainty.

As shown in Figure 2, PRMs exhibit non-uniform robustness across policy models. Distributionally mismatched PRM–Policy pairs tend to suffer lower search accuracy and higher Monte Carlo Dropout score variance, indicating that unreliable PRM scoring is correlated with elevated epistemic uncertainty. Additional calibration and depth analyses in Appendix G.1 further show that

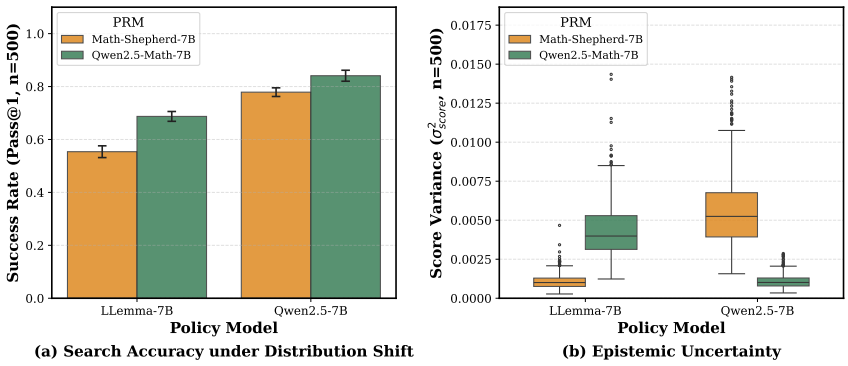


Figure 2: (a) Search accuracy across different PRM–Policy pairs on MATH-500. (b) The distribution of Monte Carlo Dropout score variance for different PRM–Policy pairs.

high-variance steps have larger ECE/Brier errors. Figure 1 illustrates a concrete failure case on MATH using Qwen2.5-Instruct-7B and Math-Shepherd-PRM-7B: the PRM ranks an incorrect OOD step ($R_\phi = 0.85$) above a correct ID step ($R_\phi = 0.76$), while the incorrect step shows much higher variance ($\sigma^2 = 0.032$ vs. 0.001). This suggests that uncertainty provides a useful signal for detecting overconfident PRM errors. More qualitative cases are provided in Appendix H.

4.2 Theoretical Analysis

In this section, we provide a theoretical analysis that connects the uncertainty in PRMs with decision-making strategies for external reasoning. Then, we discuss how these theoretical observations inform the modeling of external reasoning methods.

Influence of Uncertainty In external reasoning, the objective is to iteratively select reasoning prefixes that maximize the true reward, thereby improving final answer correctness. To study the impact of uncertainty on this process, we consider a simplified setting.

At each step $t \in \{1, \dots, T\}$, the proposer generates a set of M candidate reasoning traces $\mathcal{H}_t = \{h_{t,1}, \dots, h_{t,M}\}$. A pretrained PRM $R_\phi(\cdot)$ with parameters $\phi \sim p(\phi \mid \mathcal{D}_{\text{PRM}})$ is used to score candidates, while the true reward function $R^*(\cdot)$ is unobserved. We model the PRM score as $R_\phi(h) = R^*(h) + \xi(h)$, where $\xi(h)$ captures epistemic uncertainty.

At each step, the algorithm selects $\hat{h}_t = \arg\max_{h \in \mathcal{H}_t} R_\phi(h)$ and continues generation from the selected prefix. This procedure is repeated for T steps. We consider the practical regime in which the point estimate R_ϕ is directly used for ranking. Let $\mathcal{O}_t$ denote the event that $\mathcal{H}_t$ contains at least one OOD trace, and assume $\mathbb{P}(\mathcal{O}_t) = \varepsilon$ for all t. We now state the main result under this setting:

Proposition 4.1 (Linear degradation under epistemic uncertainty)**.** *Consider the above scenario. Suppose (i) $\mathbb{P}(\mathcal{O}_t) = \varepsilon$ for all t; (ii) under a fixed continuation policy, if $\hat{h}_t \neq h_t^*$ then the final success probability decreases by at least $\underline{\Delta}$; and (iii) there exists a lower bound $\rho \in (0, 1]$ such that $\mathbb{P}(\hat{h}_t \neq h_t^* \mid \mathcal{O}_t) \geq \rho$, $\mathbb{P}(\hat{h}_t \neq h_t^* \mid \mathcal{O}_t^c) = 0$. Then:*

$$\mathbb{E}[\mathrm{Acc}_T] \;\leq\; R^*(h_0) \;-\; T\,\varepsilon\,\underline{\Delta}\,\rho, \tag{4}$$

where Acc_T is defined as $\mathrm{Acc}_T \triangleq R^*(\hat{h}_T)$.

The proof is provided in Appendix A. When epistemic uncertainty increases, the variance of the PRM noise on OOD traces becomes larger, so the noise gap $\xi(h_t^{(2)}) - \xi(h_t^*)$ is more likely to exceed the true gap Δ_t, making the PRM ranking unreliable at OOD steps. Since OOD appears with probability ε at each iteration, the induced mistake probability accumulates over T steps, leading to a linear degradation in the final expected correctness.

Uncertainty-Aware Regime We now consider a second scenario in which epistemic uncertainty is explicitly taken into account during external reasoning. Following the same settings as above, instead of relying on a single point estimate, we draw K_t i.i.d. samples $\phi_1, \dots, \phi_{K_t} \sim p(\phi \mid \mathcal{D}_{\text{PRM}})$,

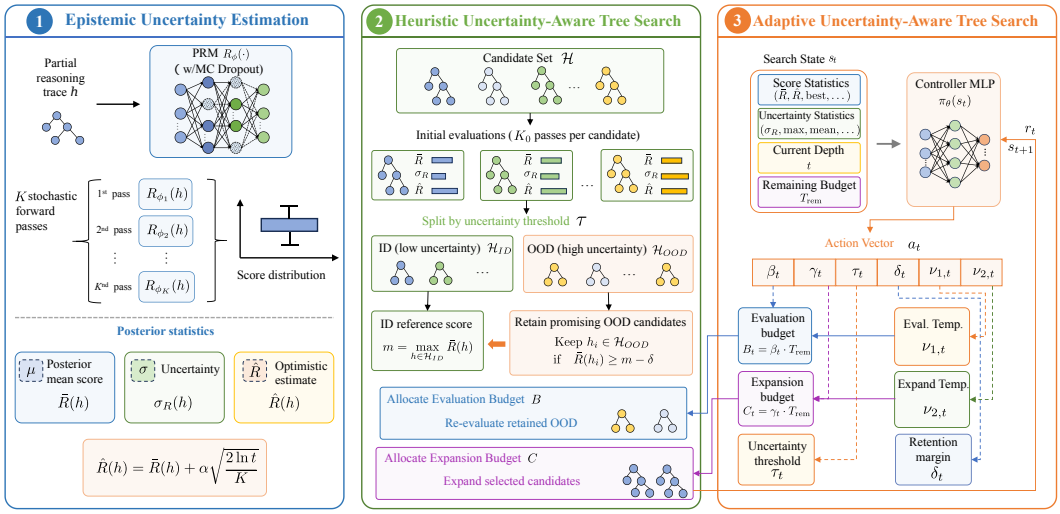


Figure 3: Overview of the proposed Uncertainty-Aware Tree Search (UATS) framework. The pipeline consists of three phases: (1) Approximate the PRM's uncertainty using Monte Carlo Dropout. (2) Use uncertainty thresholds (τ) and optimism margins (δ) to selectively re-evaluate ambiguous nodes and allocate expansion budgets. (3) Observes the search state s_t and dynamically outputs action vectors a_t to modulate budget factors and temperature parameters.

where K_t is allowed to increase with t. For each candidate $h \in \mathcal{H}_t$, we compute the empirical posterior mean $\bar{R}_t(h) \triangleq \frac{1}{K_t}\sum_{k=1}^{K_t} R_{\phi_k}(h)$. We then construct an Upper Confidence Bound (UCB) $U_t \triangleq \sqrt{\frac{2\ln t}{K_t}}$, and estimate the true reward by $\hat{R}_t(h) \triangleq \bar{R}_t(h) + U_t$. The selection is based on the estimated reward. Then we have:

Proposition 4.2 (Sublinear degradation under uncertainty-aware selection)**.** *Consider the second scenario with uncertainty estimation. Suppose (i) $\mathbb{P}(\mathcal{O}_t) = \varepsilon$ for all t; (ii) the PRM estimators are unbiased, i.e., $\mathbb{E}_\phi[\bar{R}_t(h)] = R^*(h)$; and (iii) the selection policy $\hat{h}_t$ follows the UCB criterion: $\hat{h}_t \in \arg\max_{h\in\mathcal{H}_t}\left(\bar{R}_t(h) + \sqrt{\frac{2\ln t}{K_t}}\right)$. Then, we have the lower bound:*

$$\mathbb{E}[\mathrm{Acc}_T] \;\geq R^*(h_0) - 2\varepsilon\sum_{t=2}^{T}\sqrt{\frac{2\ln t}{K_t}} - O(1). \tag{5}$$

In particular, if the sample budget scales as $K_t = \Omega(t)$, then:

$$\mathbb{E}[\mathrm{Acc}_T] \;\geq R^*(h_0) - O\big(\varepsilon\sqrt{T\ln T}\big). \tag{6}$$

The proof is provided in Appendix B. Proposition 4.2 demonstrates that by explicitly modeling epistemic uncertainty, the accumulation of errors shifts from a linear regime ($O(T)$) to a sublinear regime ($O(\sqrt{T})$). The UCB term ensures that even when OOD traces appear with probability ε, the potential loss at each step decreases over time as the sample size K_t grows.

4.3 Theoretical Implications for Method Design

The above analysis shows that uncertainty-agnostic search is vulnerable to PRM epistemic uncertainty: OOD-induced mis-ranking errors accumulate linearly, whereas uncertainty-aware selection reduces this degradation to a sublinear rate. This motivates a practical uncertainty-aware framework.

However, two challenges remain. First, uncertainty estimation requires efficient approximations to posterior sampling from $p(\phi \mid \mathcal{D}_{\mathrm{PRM}})$. Second, under a fixed inference-time budget, the algorithm must balance repeated PRM scoring for reliable evaluation against broader tree expansion. We address these challenges in the next section.

5 METHODOLOGY

To implement uncertainty-aware selection under a fixed inference-time budget, we propose Uncertainty-Aware Tree Search (UATS). UATS first estimates verifier-side epistemic uncertainty with Monte Carlo Dropout on a frozen PRM, and then uses this signal to allocate computation between PRM re-evaluation and tree expansion. We instantiate this idea with a heuristic version, H-UATS, and an adaptive version, A-UATS, where a controller adaptively adjusts allocation.

5.1 ESTIMATION OF EPISTEMIC UNCERTAINTY

Given a partial reasoning trace h, uncertainty-aware search requires multiple evaluations from plausible PRM parameter instances $\phi_1, \ldots, \phi_K \sim p(\phi \mid \mathcal{D}_{\mathrm{PRM}})$. Since direct posterior sampling is intractable for modern PRMs, we approximate it with Monte Carlo Dropout Gal & Ghahramani (2016). Specifically, we keep the PRM frozen, enable dropout during inference, and perform K stochastic forward passes on the same reasoning trace h. Denoting the resulting scores by $\{R_{\phi_k}(h)\}_{k=1}^{K}$, the mean and variance are:

$$\bar{R}(h) \triangleq \frac{1}{K}\sum_{k=1}^{K} R_{\phi_k}(h), \quad \sigma_R^2(h) \triangleq \frac{1}{K-1}\sum_{k=1}^{K}\big(R_{\phi_k}(h) - \bar{R}(h)\big)^2. \tag{7}$$

Here, $\bar{R}(h)$ is the averaged PRM score, and $\sigma_R(h)$ measures verifier-side epistemic uncertainty. We further define an upper-confidence estimate $\hat{R}(h) \triangleq \bar{R}(h) + \alpha\sqrt{\frac{2\ln t}{K}}$ at reasoning step t, where α controls the optimism strength. We use the tuple $(\bar{R}(h), \sigma_R(h), K(h), \hat{R}(h))$ to allocate the budget at test time.

5.2 HEURISTIC UNCERTAINTY-AWARE TREE SEARCH

H-UATS allocates a fixed compute budget between repeated PRM evaluation and tree expansion. At each expansion, a child node is generated by decoding one reasoning segment from the current prefix until a predefined step separator is reached. The accumulated trace is formatted with PRM-specific step tags, and the PRM reward then scores each generated step.

Given a candidate set of partial traces $\mathcal{H}$, we first run K_0 Monte Carlo Dropout forward passes for each $h \in \mathcal{H}$ and compute $(\bar{R}(h), \sigma_R(h), K(h), \hat{R}(h))$, where $K(h) = K_0$. We then split the candidates by an uncertainty threshold τ:

$$\mathcal{H}_{\mathrm{ID}} \triangleq \{h \in \mathcal{H} : \sigma_R(h) \le \tau\}, \quad \mathcal{H}_{\mathrm{OOD}} \triangleq \{h \in \mathcal{H} : \sigma_R(h) > \tau\}. \tag{8}$$

For uncertain candidates, H-UATS only re-evaluates those whose optimistic score remains competitive. Specifically, an OOD candidate $h_i \in \mathcal{H}_{\mathrm{OOD}}$ is retained for additional PRM evaluation if $\hat{R}(h_i) \ge \max_{h\in\mathcal{H}_{\mathrm{ID}}} \bar{R}(h) - \delta$, where δ is a retention margin. Let $\mathcal{S}$ denote the retained subset. The additional evaluation budget B is allocated as $K_i = \mathrm{Round}\left(B \cdot \frac{\exp(\hat{R}(h_i)/\nu_1)}{\sum_{h_j\in\mathcal{S}}\exp(\hat{R}(h_j)/\nu_1)}\right)$, where ν_1 controls the concentration of re-evaluation budget.

After updating the PRM statistics with additional evaluations, H-UATS allocates the child-node expansion budget C following the score-proportional principle of REBASE Wu et al. (2024): $N_i = \mathrm{Round}\left(C \cdot \frac{\exp(\bar{R}(h_i)/\nu_2)}{\sum_{h_j\in\mathcal{H}}\exp(\bar{R}(h_j)/\nu_2)}\right)$, where ν_2 controls the exploration–exploitation trade-off during expansion. Each candidate h_i is then expanded to generate N_i children.

5.3 ADAPTIVE UNCERTAINTY-AWARE CONTROLLER

H-UATS uses fixed hyperparameters, including the budgets B, C, thresholds τ, δ, and temperatures ν_1, ν_2. Since reasoning difficulty varies across questions and intermediate steps, we further introduce Adaptive Uncertainty-Aware Tree Search (A-UATS), which replaces these fixed values with state-dependent decisions. The controller is implemented as a lightweight MLP parameterized by ψ.

We formulate dynamic budget regulation as a Markov Decision Process. At step t, the state s_t summarizes the current candidate set, including PRM score statistics $(\mu_{\bar{R}}, \max \bar{R}, \sigma^2_{\bar{R}})$, uncertainty statistics $(\mu_{\sigma_R}, \max \sigma_R)$, the current depth t, and the remaining global budget T_{rem}. The controller outputs:

$$a_t = \pi_\psi(s_t) = [\beta_t, \gamma_t, \tau_t, \delta_t, \nu_{1,t}, \nu_{2,t}], \tag{9}$$

where β_t and γ_t scale the evaluation and expansion budgets, τ_t and δ_t determine uncertainty filtering, and $\nu_{1,t}, \nu_{2,t}$ control the concentration of re-evaluation and expansion. These dynamic values replace the corresponding fixed hyperparameters.

The episode-level reward is defined as:

$$r = \mathbb{1}(\hat{a} = a^*) - \lambda \cdot \frac{\text{Cost}_{\text{used}}}{\text{Cost}_{\text{total}}}, \tag{10}$$

where $\hat{a}$ is the extracted final answer and λ controls the cost penalty. We first warm-start the controller with behavioral cloning to imitate H-UATS, and then fine-tune it with REINFORCE Williams (1992) to maximize the expected reward. Thus, A-UATS preserves uncertainty-aware search while learning adaptive budget allocation throughout the search process.

6 Experiments

In this section, we evaluate UATS on reasoning benchmarks, analyze its compute efficiency and generalization ability, and verify the role of uncertainty estimation through ablation studies. The appendix contains implementation details, hyperparameters, and full results.

6.1 Mathematical Reasoning Evaluation

Experimental Setup. We evaluate UATS on two mathematical reasoning benchmarks: MATH-500 Hendrycks et al. (2021); Lightman et al. (2023) and AIME24 AI-MO (2024). We use five frozen policy models, including LLaMA3.1-Instruct-8B, LLaMA3.2-Instruct-1B, and Qwen2.5-Instruct models with 0.5B, 3B, and 7B parameters. For PRMs, we use Math-Shepherd-PRM-7B, Qwen2.5-Math-PRM-7B, and Skywork-PRM-1.5B. We compare against Best-of-N, Beam Search, DoRA, DVTS, and REBASE. All methods are evaluated under the same candidate budgets $N \in \{4, 8, 16, 32, 64, 128, 256\}$, and all results are averaged over 5 independent runs. Appendix F reports the complete results.

Main Results. Table 1 summarizes the overall performance across all Policy–PRM combinations and candidate budgets. H-UATS consistently improves over uncertainty-agnostic search methods, outperforming the strongest baseline DoRA by 1.76 points on MATH-500 and 2.29 points on AIME24. A-UATS further improves over H-UATS by 1.81 points on MATH-500 and 1.50 points on AIME24, achieving the best average performance on both benchmarks. Across the per-pair results, A-UATS outperforms all baselines in all 15 model combinations on both MATH-500 and AIME24, showing that adaptive uncertainty-aware allocation is robust across proposers and verifiers.

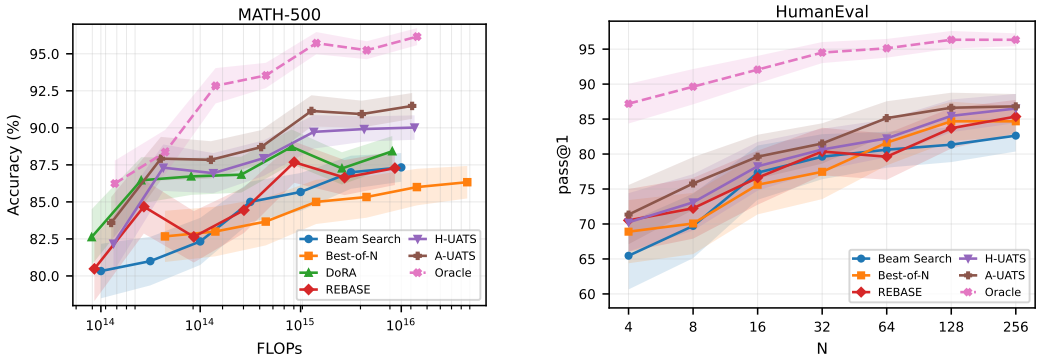
Table 1: Average final-answer accuracy across 15 Policy–PRM pairs and all candidate budgets.

Method	MATH	AIME
Beam Search	64.59	10.56
Best-of-N	64.07	10.63
DoRA	68.82	13.58
DVTS	66.54	12.33
REBASE	65.72	12.22
H-UATS	70.58	15.87
A-UATS	**72.39**	**17.37**

6.2 Compute Efficiency and Task Generalization

We further evaluate UATS from the perspectives of compute efficiency and task generalization. As shown in Figure 4(a), UATS maintains clear advantages under FLOPs-matched inference budgets on a representative MATH-500 setting, indicating that the gains do not simply come from using more test-time compute. The full FLOPs-based curves appear in Appendix F.3.

Figure 4(b) reports results on HumanEval Chen et al. (2021), using Qwen2.5-Coder-7B-Instruct Hui et al. (2024) as the policy model and Skywork-o1-Open-PRM-Qwen-2.5-7B Skywork o1 Team



(a) FLOPs-matched comparison

(b) Evaluation on HumanEval

Figure 4: Compute efficiency and task generalization of UATS. (a) Accuracy under FLOPs-matched inference budgets on MATH-500 using Qwen2.5-Instruct-7B and Qwen2.5-Math-PRM-7B. (b) Evaluation on HumanEval using Qwen2.5-Coder-7B-Instruct as the policy model and Skywork-o1-Open-PRM-Qwen-2.5-7B as the verifier.

Table 2: Ablation study evaluating the impact of Uncertainty Estimation across different Policy and PRM combinations. We report accuracy (%) on MATH-500 and AIME24.

Policy Model	PRM	Method	MATH-500	AIME24
Qwen2.5-7B-Instruct	Qwen2.5-Math-PRM-7B	H-UATS w/o Uncert.	85.62	18.82
		H-UATS	**87.72**	**26.19**
		A-UATS w/o Uncert.	85.52	22.47
		A-UATS	**88.80**	**28.10**
Qwen2.5-7B-Instruct	Math-Shepherd-PRM-7B	H-UATS w/o Uncert.	80.42	21.46
		H-UATS	**84.67**	**24.29**
		A-UATS w/o Uncert.	82.16	21.42
		A-UATS	**86.54**	**26.19**
LLaMA3.1-8B-Instruct	Math-Shepherd-PRM-7B	H-UATS w/o Uncert.	57.72	16.01
		H-UATS	**64.48**	**19.62**
		A-UATS w/o Uncert.	61.02	16.67
		A-UATS	**66.64**	**21.52**

(2024) as the verifier. H-UATS and A-UATS outperform standard search baselines across candidate budgets, showing that uncertainty-aware search generalizes beyond mathematics.

6.3 Ablation Studies

We ablate verifier-side uncertainty in H-UATS and A-UATS by removing Monte Carlo Dropout or uncertainty features, respectively. Table 2 shows that both methods outperform the w/o Uncert. variants on MATH-500 and AIME24, with larger gains on harder AIME24; adaptive allocation further improves over fixed heuristics.

Appendix G gives sensitivity analyses for dropout, K_0, τ, δ, and controller architecture.

7 Conclusion

We identify epistemic uncertainty in PRMs as a critical challenge: inference-time distribution shifts undermine search reliability, while uncertainty-aware strategies reduce search regret from linear to sublinear growth. UATS dynamically allocates compute via a learned controller, mitigates OOD errors, and outperforms baselines, highlighting the value of verifier-side uncertainty modeling for reliable reasoning systems.

AI USE STATEMENT

In this work, we used generative AI tools for language polishing and for generating code for some experimental visualizations. We have not used generative AI tools to generate research ideas, design the main method, or implement the core method code, and the remaining required disclosure tasks are not applicable to this work. Additionally, we used generative AI tools only for manuscript editing and auxiliary plotting-code assistance. All AI-assisted text and code were reviewed by the authors, and the generated code and resulting visualizations were checked for correctness. We take responsibility for the final content of this work, including text, claims, and artifacts produced with the aid of generative AI.

REFERENCES

AI-MO. AIME 2024. https://huggingface.co/datasets/AI-MO/aimo-validation-aime, July 2024. Accessed 2024-07.

Peter Auer, Nicolò Cesa-Bianchi, and Paul Fischer. Finite-time analysis of the multiarmed bandit problem. *Machine Learning*, 47(2–3):235–256, 2002. doi: 10.1023/A:1013689704352.

Maciej Besta, Nils Blach, Ales Kubicek, Robert Gerstenberger, Michal Podstawski, Lukas Gianinazzi, Joanna Gajda, Tomasz Lehmann, Hubert Niewiadomski, and Piotr Nyczyk. Graph of thoughts: Solving elaborate problems with large language models. In *Proceedings of the AAAI Conference on Artificial Intelligence*, volume 38, pp. 17682–17690, 2024.

Bradley Brown, Jordan Juravsky, Ryan Ehrlich, Ronald Clark, Quoc V. Le, Christopher Ré, and Azalia Mirhoseini. Large Language Monkeys: Scaling Inference Compute with Repeated Sampling, December 2024.

Mark Chen, Jerry Tworek, Heewoo Jun, Qiming Yuan, Henrique Ponde de Oliveira Pinto, Jared Kaplan, Harri Edwards, Yuri Burda, Nicholas Joseph, Greg Brockman, Alex Ray, Raul Puri, Gretchen Krueger, Michael Petrov, Heidy Khlaaf, Girish Sastry, Pamela Mishkin, Brooke Chan, Scott Gray, Nick Ryder, Mikhail Pavlov, Alethea Power, Lukasz Kaiser, Mohammad Bavarian, Clemens Winter, Philippe Tillet, Felipe Petroski Such, Dave Cummings, Matthias Plappert, Fotios Chantzis, Elizabeth Barnes, Ariel Herbert-Voss, William Hebgen Guss, Alex Nichol, Alex Paino, Nikolas Tezak, Jie Tang, Igor Babuschkin, Suchir Balaji, Shantanu Jain, William Saunders, Christopher Hesse, Andrew N. Carr, Jan Leike, Josh Achiam, Vedant Misra, Evan Morikawa, Alec Radford, Matthew Knight, Miles Brundage, Mira Murati, Katie Mayer, Peter Welinder, Bob McGrew, Dario Amodei, Sam McCandlish, Ilya Sutskever, and Wojciech Zaremba. Evaluating large language models trained on code, July 2021.

DeepSeek-AI. DeepSeek-R1: Incentivizing Reasoning Capability in LLMs via Reinforcement Learning, January 2025.

Sebastian Farquhar, Jannik Kossen, Lorenz Kuhn, and Yarin Gal. Detecting hallucinations in large language models using semantic entropy. *Nature*, 630(8017):625–630, 2024. doi: 10.1038/s41586-024-07421-0.

Yarin Gal and Zoubin Ghahramani. Dropout as a bayesian approximation: Representing model uncertainty in deep learning. In *international conference on machine learning*, pp. 1050–1059. PMLR, 2016.

Aaron Grattafiori, Abhimanyu Dubey, Abhinav Jauhri, Abhinav Pandey, Abhishek Kadian, Ahmad Al-Dahle, Aiesha Letman, Akhil Mathur, Alan Schelten, Alex Vaughan, et al. The llama 3 herd of models. *arXiv preprint arXiv:2407.21783*, 2024.

Neha Gupta, Harikrishna Narasimhan, Wittawat Jitkrittum, Ankit Singh Rawat, Aditya Krishna Menon, and Sanjiv Kumar. Language Model Cascades: Token-level uncertainty and beyond, April 2024.

Dan Hendrycks, Collin Burns, Saurav Kadavath, Akul Arora, Steven Basart, Eric Tang, Dawn Song, and Jacob Steinhardt. Measuring mathematical problem solving with the math dataset. *NeurIPS*, 2021.

Binyuan Hui, Jian Yang, Zeyu Cui, Jiaxi Yang, Dayiheng Liu, Lei Zhang, Tianyu Liu, Jiajun Zhang, Bowen Yu, Keming Lu, Kai Dang, Yang Fan, Yichang Zhang, An Yang, Rui Men, Fei Huang, Bo Zheng, Yibo Miao, Shanghaoran Quan, Yunlong Feng, Xingzhang Ren, Xuancheng Ren, Jingren Zhou, and Junyang Lin. Qwen2.5-coder technical report, 2024.

Mykel J Kochenderfer, Tim A Wheeler, and Kyle H Wray. *Algorithms for decision making*. MIT press, 2022.

Levente Kocsis and Csaba Szepesvári. Bandit based monte-carlo planning. In *Machine Learning: ECML 2006*, volume 4212 of *Lecture Notes in Computer Science*, pp. 282–293. Springer, 2006. doi: 10.1007/11871842_29.

Takeshi Kojima, Shixiang Shane Gu, Machel Reid, Yutaka Matsuo, and Yusuke Iwasawa. Large language models are zero-shot reasoners. *Advances in neural information processing systems*, 35: 22199–22213, 2022.

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

Balaji Lakshminarayanan, Alexander Pritzel, and Charles Blundell. Simple and scalable predictive uncertainty estimation using deep ensembles. In *Advances in Neural Information Processing Systems*, volume 30, 2017.

Hunter Lightman, Vineet Kosaraju, Yura Burda, Harri Edwards, Bowen Baker, Teddy Lee, Jan Leike, John Schulman, Ilya Sutskever, and Karl Cobbe. Let's Verify Step by Step, May 2023.

Runze Liu, Junqi Gao, Jian Zhao, Kaiyan Zhang, Xiu Li, Biqing Qi, Wanli Ouyang, and Bowen Zhou. Can 1B LLM Surpass 405B LLM? Rethinking Compute-Optimal Test-Time Scaling, February 2025.

Meta AI. Llama 3 to connect 2024: Vision, edge, and mobile devices. https://ai.meta.com/blog/llama-3-2-connect-2024-vision-edge-mobile-devices/, 2024. Accessed: 2025-05-12.

OpenAI. Learning to reason with llms. https://openai.com/index/learning-to-reason-with-llms/, 2024. Accessed: 2025-04-29.

Bilgehan Sel, Ahmad Tawaha, Vanshaj Khattar, Ruoxi Jia, and Ming Jin. Algorithm of Thoughts: Enhancing Exploration of Ideas in Large Language Models. In *Forty-First International Conference on Machine Learning*, June 2024.

Skywork o1 Team. Skywork-o1 open series. https://huggingface.co/Skywork, November 2024. Accessed: 2025-05-12.

Charlie Victor Snell, Jaehoon Lee, Kelvin Xu, and Aviral Kumar. Scaling LLM Test-Time Compute Optimally Can be More Effective than Scaling Parameters for Reasoning. In *The Thirteenth International Conference on Learning Representations*, October 2024.

Richard S Sutton, David McAllester, Satinder Singh, and Yishay Mansour. Policy gradient methods for reinforcement learning with function approximation. *Advances in neural information processing systems*, 12, 1999.

Kimi Team, Angang Du, Bofei Gao, Bowei Xing, Changjiu Jiang, Cheng Chen, Cheng Li, Chenjun Xiao, Chenzhuang Du, Chonghua Liao, Chuning Tang, Congcong Wang, Dehao Zhang, Enming Yuan, Enzhe Lu, Fengxiang Tang, Flood Sung, Guangda Wei, Guokun Lai, Haiqing Guo, Han Zhu, Hao Ding, Hao Hu, Hao Yang, Hao Zhang, Haotian Yao, Haotian Zhao, Haoyu Lu, Haoze Li, Haozhen Yu, Hongcheng Gao, Huabin Zheng, Huan Yuan, Jia Chen, Jianhang Guo, Jianlin Su, Jianzhou Wang, Jie Zhao, Jin Zhang, Jingyuan Liu, Junjie Yan, Junyan Wu, Lidong Shi, Ling Ye, Longhui Yu, Mengnan Dong, Neo Zhang, Ningchen Ma, Qiwei Pan, Qucheng Gong, Shaowei Liu, Shengling Ma, Shupeng Wei, Sihan Cao, Siying Huang, Tao Jiang, Weihao Gao, Weimin Xiong, Weiran He, Weixiao Huang, Wenhao Wu, Wenyang He, Xianghui Wei, Xianqing Jia, Xingzhe Wu, Xinran Xu, Xinxing Zu, Xinyu Zhou, Xuehai Pan, Y. Charles, Yang Li, Yangyang Hu, Yangyang

Liu, Yanru Chen, Yejie Wang, Yibo Liu, Yidao Qin, Yifeng Liu, Ying Yang, Yiping Bao, Yulun Du, Yuxin Wu, Yuzhi Wang, Zaida Zhou, Zhaoji Wang, Zhaowei Li, Zhen Zhu, Zheng Zhang, Zhexu Wang, Zhilin Yang, Zhiqi Huang, Zihao Huang, Ziyao Xu, and Zonghan Yang. Kimi k1.5: Scaling Reinforcement Learning with LLMs, January 2025.

Jonathan Uesato, Nate Kushman, Ramana Kumar, Francis Song, Noah Siegel, Lisa Wang, Antonia Creswell, Geoffrey Irving, and Irina Higgins. Solving math word problems with process-and outcome-based feedback. *arXiv preprint arXiv:2211.14275*, 2022.

Peiyi Wang, Lei Li, Zhihong Shao, R. X. Xu, Damai Dai, Yifei Li, Deli Chen, Y. Wu, and Zhifang Sui. Math-Shepherd: Verify and Reinforce LLMs Step-by-step without Human Annotations, February 2024.

Xinglin Wang, Yiwei Li, Shaoxiong Feng, Peiwen Yuan, Yueqi Zhang, Jiayi Shi, Chuyi Tan, Boyuan Pan, Yao Hu, and Kan Li. Every rollout counts: Optimal resource allocation for efficient test-time scaling. *arXiv preprint arXiv:2506.15707*, 2025.

Jason Wei, Xuezhi Wang, Dale Schuurmans, Maarten Bosma, Fei Xia, Ed Chi, Quoc V. Le, and Denny Zhou. Chain-of-thought prompting elicits reasoning in large language models. *Advances in neural information processing systems*, 35:24824–24837, 2022.

Ronald J Williams. Simple statistical gradient-following algorithms for connectionist reinforcement learning. *Machine learning*, 8(3):229–256, 1992.

Yangzhen Wu, Zhiqing Sun, Shanda Li, Sean Welleck, and Yiming Yang. Inference Scaling Laws: An Empirical Analysis of Compute-Optimal Inference for LLM Problem-Solving. In *The Thirteenth International Conference on Learning Representations*, October 2024.

Yuxi Xie, Kenji Kawaguchi, Yiran Zhao, Xu Zhao, Min-Yen Kan, Junxian He, and Qizhe Xie. Self-Evaluation Guided Beam Search for Reasoning, October 2023.

An Yang, Baosong Yang, Beichen Zhang, Binyuan Hui, Bo Zheng, Bowen Yu, Chengyuan Li, Dayiheng Liu, Fei Huang, Haoran Wei, et al. Qwen2. 5 technical report. *arXiv preprint arXiv:2412.15115*, 2024a.

An Yang, Beichen Zhang, Binyuan Hui, Bofei Gao, Bowen Yu, Chengpeng Li, Dayiheng Liu, Jianhong Tu, Jingren Zhou, Junyang Lin, Keming Lu, Mingfeng Xue, Runji Lin, Tianyu Liu, Xingzhang Ren, and Zhenru Zhang. Qwen2.5-Math Technical Report: Toward Mathematical Expert Model via Self-Improvement, September 2024b.

Shunyu Yao, Dian Yu, Jeffrey Zhao, Izhak Shafran, Thomas L. Griffiths, Yuan Cao, and Karthik R. Narasimhan. Tree of Thoughts: Deliberate Problem Solving with Large Language Models. In *Thirty-Seventh Conference on Neural Information Processing Systems*, November 2023.

Zhenru Zhang, Chujie Zheng, Yangzhen Wu, Beichen Zhang, Runji Lin, Bowen Yu, Dayiheng Liu, Jingren Zhou, and Junyang Lin. The Lessons of Developing Process Reward Models in Mathematical Reasoning, January 2025.

Zhuosheng Zhang, Aston Zhang, Mu Li, and Alex Smola. Automatic Chain of Thought Prompting in Large Language Models, October 2022.

APPENDIX

The appendix is organized as follows:

1. **Appendix A** presents the proof of Proposition 4.1.
2. **Appendix B** presents the proof of Proposition 4.2.
3. **Appendix C** discusses the limitation of this work.
4. **Appendix D** provides comparison and discussion with related works.
5. **Appendix E** provides specific implementation of UATS and Hyperparameters.

6. **Appendix F** provides the full results.
7. **Appendix G** provides additional ablation studies.
8. **Appendix H** provides qualitative results.

A Proof of Proposition 4.1

Proof. We analyze the degradation of the expected accuracy by examining the error probability at each reasoning step $t \in \{1, \dots, T\}$.

Let E_t denote the event that the selected hypothesis is incorrect at step t, i.e., $E_t = \{\hat{h}_t \neq h_t^*\}$. We aim to lower-bound the probability of this error event, $\mathbb{P}(E_t)$. Using the Law of Total Probability with respect to the OOD event $\mathcal{O}_t$, we have:

$$\mathbb{P}(E_t) = \mathbb{P}(E_t \mid \mathcal{O}_t)\mathbb{P}(\mathcal{O}_t) + \mathbb{P}(E_t \mid \mathcal{O}_t^c)\mathbb{P}(\mathcal{O}_t^c). \tag{11}$$

Applying the assumptions given in the proposition:

(i) The OOD probability is fixed: $\mathbb{P}(\mathcal{O}_t) = \varepsilon$.

(iii) The conditional error probabilities satisfy $\mathbb{P}(E_t \mid \mathcal{O}_t) \geq \rho$ and $\mathbb{P}(E_t \mid \mathcal{O}_t^c) = 0$.

Substituting these into the equation, we obtain a lower bound for the error probability at step t:

$$\mathbb{P}(E_t) \geq \rho \cdot \varepsilon + 0 \cdot (1 - \varepsilon) = \varepsilon\rho. \tag{12}$$

Next, we quantify the impact of these errors on the final accuracy. Let L_t be the random variable representing the reduction in success probability caused by the selection at step t. According to assumption (ii), if an error occurs (E_t holds), the success probability decreases by at least $\underline{\Delta}$; otherwise, the reduction is non-negative (we assume no improvement from incorrect steps). Thus, we can bound the expected loss at step t as:

$$\mathbb{E}[L_t] \geq \underline{\Delta} \cdot \mathbb{P}(E_t) + 0 \cdot \mathbb{P}(E_t^c). \tag{13}$$

Using the bound from Eq. equation 12:

$$\mathbb{E}[L_t] \geq \underline{\Delta} \cdot (\varepsilon\rho) = \varepsilon\rho\underline{\Delta}. \tag{14}$$

The final accuracy Acc_T is the initial potential reward $R^*(h_0)$ minus the cumulative losses over all T steps. Taking the expectation:

$$\mathbb{E}[\mathrm{Acc}_T] \leq R^*(h_0) - \sum_{t=1}^{T} \mathbb{E}[L_t]. \tag{15}$$

Substituting the lower bound for the expected loss at each step:

$$\begin{aligned} \mathbb{E}[\mathrm{Acc}_T] &\leq R^*(h_0) - \sum_{t=1}^{T} \varepsilon\rho\underline{\Delta} \\ &= R^*(h_0) - T\,\varepsilon\,\underline{\Delta}\,\rho. \end{aligned} \tag{16}$$

This concludes the proof. □

B Proof of Proposition 4.2

Proof. To analyze the expected performance, we define the instantaneous regret at step t as $r_t \triangleq R^*(h_t^*) - R^*(\hat{h}_t)$. The final accuracy is the initial potential minus the cumulative regret: $\mathbb{E}[\mathrm{Acc}_T] = R^*(h_0) - \sum_{t=1}^{T} \mathbb{E}[r_t]$.

First, we analyze the concentration of the reward estimates. Let $\mathcal{E}_t$ denote the "good event" where the empirical mean estimates for all candidates in $\mathcal{H}_t$ are close to their true values within the confidence interval U_t:

$$\mathcal{E}_t = \left\{\forall h \in \mathcal{H}_t,\ |\bar{R}_t(h) - R^*(h)| \leq U_t\right\}, \tag{17}$$

where $U_t = \sqrt{\frac{2 \ln t}{K_t}}$. Using Hoeffding's inequality for a single candidate h, the probability of deviation is bounded by:

$$\mathbb{P}(|\bar{R}_t(h) - R^*(h)| > U_t) \leq 2 \exp\left(-2K_t U_t^2\right) = 2 \exp(-4 \ln t) = 2t^{-4}. \tag{18}$$

Applying the union bound over all M candidates in $\mathcal{H}_t$, the probability that the good event fails is:

$$\mathbb{P}(\mathcal{E}_t^c) \leq \sum_{h \in \mathcal{H}_t} 2t^{-4} = 2Mt^{-4}. \tag{19}$$

Next, we bound the expected regret $\mathbb{E}[r_t]$. We decompose this expectation based on the occurrence of the OOD event $\mathcal{O}_t$. Recall that $\mathbb{P}(\mathcal{O}_t) = \varepsilon$.

1. **Case 1: No OOD ($\mathcal{O}_t^c$).** We assume the model performs optimally or near-optimally in distribution, contributing negligible regret compared to the OOD case.
2. **Case 2: OOD occurs ($\mathcal{O}_t$).** We further decompose based on the concentration event $\mathcal{E}_t$:
   - If $\mathcal{E}_t$ holds: The UCB selection strategy guarantees a regret bound. By definition of $\hat{h}_t$ (optimizing the upper bound) and $\mathcal{E}_t$ (true value contained in bound):
   $$R^*(h_t^*) \leq \bar{R}_t(h_t^*) + U_t \leq \bar{R}_t(\hat{h}_t) + U_t \leq R^*(\hat{h}_t) + 2U_t. \tag{20}$$
   Thus, $r_t = R^*(h_t^*) - R^*(\hat{h}_t) \leq 2U_t$.
   - If $\mathcal{E}_t^c$ holds: The regret is bounded by the maximum possible reward difference, denoted by $C_{\max}$ (e.g., 1). This occurs with probability at most $2Mt^{-4}$.

Combining these, the expected regret at step t is dominated by the OOD interaction:

$$\begin{aligned} \mathbb{E}[r_t] &\leq \varepsilon \left(\mathbb{P}(\mathcal{E}_t) \cdot 2U_t + \mathbb{P}(\mathcal{E}_t^c) \cdot C_{\max}\right) \\ &\leq 2\varepsilon U_t + 2MC_{\max}\varepsilon t^{-4}. \end{aligned} \tag{21}$$

Finally, we sum the expected regret over $t = 1$ to T to bound the total degradation:

$$R^*(h_0) - \mathbb{E}[\mathrm{Acc}_T] = \sum_{t=1}^{T} \mathbb{E}[r_t] \leq \sum_{t=1}^{T} \left(2\varepsilon U_t + O(\varepsilon t^{-4})\right). \tag{22}$$

Since $\sum_{t=1}^{\infty} t^{-4}$ converges to a constant, the second term is $O(1)$. For the first term, substituting $U_t = \sqrt{\frac{2 \ln t}{K_t}}$:

$$\text{Total Regret} \leq 2\varepsilon \sum_{t=1}^{T} \sqrt{\frac{2 \ln t}{K_t}} + O(1). \tag{23}$$

If $K_t = \Omega(t)$, then $U_t \approx \sqrt{\frac{\ln t}{t}}$. Approximating the sum by an integral $\int_1^T \sqrt{\frac{\ln x}{x}} dx \approx \sqrt{T \ln T}$, we obtain:

$$R^*(h_0) - \mathbb{E}[\mathrm{Acc}_T] = O\left(\varepsilon \sqrt{T \ln T}\right). \tag{24}$$

This confirms the sublinear degradation. □

C LIMITATIONS

This work focuses on PRM-guided external reasoning systems where a frozen proposer, a frozen verifier, and a search algorithm are jointly used at inference time. Therefore, the effectiveness of UATS depends on the availability of a reasonably informative verifier: uncertainty-aware search can improve the reliability of candidate selection, but it is not intended to replace the verifier itself. In addition, our main evaluations are conducted on mathematical reasoning benchmarks, with a preliminary extension to code generation. These settings provide clear step-level verification and final-answer evaluation, making them suitable for controlled analysis. Extending uncertainty-aware search to broader reasoning domains would require reliable task-specific verifiers, which we leave for future work.

D Additional Discussions

D.1 Comparison with REBASE

REBASE Wu et al. (2024) allocates inference compute based on a score-proportional principle, implicitly treating PRM scores as reliable point estimates. However, this assumption fails under distribution shifts, where PRMs often exhibit high confidence on incorrect OOD steps. UATS fundamentally differs by introducing an epistemic uncertainty gate: instead of blindly trusting raw scores, we explicitly model the reliability of the verifier. While REBASE asks "where to expand based on quality," UATS first answers "whether the quality assessment is trustworthy." By selectively re-evaluating high-variance candidates, UATS prevents the over-expansion of hallucinated high-scoring paths, a critical failure mode that REBASE cannot address. Furthermore, A-UATS extends this by dynamically learning the optimal trade-off between verification (re-evaluation) and search breadth, whereas REBASE relies on static hyperparameters.

D.2 Comparison with DoRA

DORA Wang et al. (2025) focuses on macro-level resource allocation, aiming to reduce redundancy by allocating rollout budgets to distinct reasoning directions (i.e., solution clusters). In contrast, UATS addresses micro-level verification reliability, targeting the epistemic errors of the PRM at individual reasoning steps. The two methods solve orthogonal inefficiencies: DORA mitigates the diminishing returns of sampling similar paths, while UATS mitigates the risk of being misled by OOD generalizations. Consequently, they are highly complementary; UATS provides calibrated step-level signals that can enhance the direction-level decision-making in DORA.

D.3 Comparison with MCTS

Monte Carlo Tree Search (MCTS), especially its UCT variant, balances exploitation and exploration by applying upper confidence bounds to tree search Kocsis & Szepesvári (2006); Auer et al. (2002). In standard MCTS, node values are typically estimated through repeated rollouts and back-propagation, which can be computationally expensive when each rollout corresponds to long-form LLM generation. In LLM reasoning, such deep rollouts incur prohibitive latency and compute costs Snell et al. (2024).

UATS differs from MCTS in both the source of uncertainty and the way computation is allocated. Rather than estimating node values by sequential future rollouts, UATS estimates verifier-side epistemic uncertainty through lightweight and parallelizable stochastic PRM forward passes. This design allows UATS to correct unreliable PRM scores and prune risky branches without requiring expensive generation rollouts, making it more suitable for fixed-budget PRM-guided external reasoning.

D.4 Relation to Classical Uncertainty Quantification Tools

UATS uses Monte Carlo Dropout and UCB-style optimism as practical modeling tools, rather than claiming methodological novelty in these classical components themselves. Monte Carlo Dropout provides an efficient approximation to posterior predictive variability by enabling stochastic forward passes of the same frozen PRM Gal & Ghahramani (2016). Compared with ensemble-based uncertainty estimation Lakshminarayanan et al. (2017), this choice is particularly suitable for our setting because it does not require training multiple verifier instances or modifying the PRM training pipeline. The UCB-style term follows the classical confidence-bound principle for balancing exploration and exploitation under uncertainty Auer et al. (2002).

The key contribution of UATS is not the invention of a new uncertainty estimator, but the adaptation of uncertainty estimation to a specific bottleneck in PRM-guided external reasoning: existing search methods typically treat PRM scores as reliable point estimates, which can lead to overconfident mis-ranking when the proposer generates reasoning traces outside the verifier's training distribution. UATS explicitly models this verifier-side uncertainty and uses it to regulate fixed-budget search. In this sense, Monte Carlo Dropout and UCB serve as mechanisms for operationalizing the broader principle that unreliable verifier judgments should receive additional verification before they dominate the search process.

D.5 Relation to Bandit and Reinforcement Learning

Our method is also related to ideas from bandit and reinforcement learning, but the setting differs from standard formulations in important ways. On the bandit side, the theoretical analysis adopts an optimism-in-the-face-of-uncertainty argument Auer et al. (2002). However, external reasoning is not a standard fixed-arm bandit problem. The objects being compared are partial reasoning traces generated online during search, and the uncertainty of interest is not generic reward noise over fixed arms, but verifier-side epistemic uncertainty induced by a mismatch between the proposer-generated traces and the PRM training distribution. Therefore, the UCB-style analysis is used to characterize the role of uncertainty-aware selection in PRM-guided search, rather than to reduce the problem to a conventional bandit setting.

On the reinforcement-learning side, A-UATS formulates dynamic budget regulation as an MDP and uses policy-gradient optimization to learn a state-dependent controller Williams (1992); Sutton et al. (1999). Here, reinforcement learning is used as a practical optimization tool for adaptive test-time compute allocation. The controller learns how to adjust the verification budget, expansion budget, uncertainty threshold, optimism margin, and temperature parameters according to the current state of the search tree. The theoretical analysis in Section 4 establishes the basic mechanism that uncertainty-aware selection can reduce variance-induced search degradation, while the learned controller is positioned as a systems-level extension that improves the efficiency of this principle under a fixed compute budget. Providing formal guarantees for deep RL in the non-stationary search process of LLM generation is beyond the scope of this work.

D.6 Difference from Entropy-Based Uncertainty

Entropy-based uncertainty measures and UATS quantify uncertainty over different random variables. Token-level entropy characterizes uncertainty in the proposer's next-token distribution Gupta et al. (2024), while semantic entropy further groups generations according to semantic equivalence and measures uncertainty at the meaning level Kuhn et al. (2023); Farquhar et al. (2024). Such measures are useful for analyzing generation-side uncertainty, because they describe whether the policy model is confident about what to generate next or whether different sampled outputs express consistent meanings.

In contrast, UATS focuses on verifier-side uncertainty. Given a fixed candidate reasoning trace, we estimate how reliably the PRM evaluates this trace by measuring the variability of PRM scores across stochastic parameter realizations. Therefore, our uncertainty is not about whether the proposer can generate diverse solutions, but about whether the verifier's score for an already generated reasoning step should be trusted. This distinction is important in PRM-guided external reasoning, where the final search decision depends not only on the existence of candidate trajectories, but also on whether the verifier ranks them correctly.

For the same reason, temperature-based sampling is not equivalent to the uncertainty used in UATS. Increasing the proposer's decoding temperature changes the distribution over generated trajectories and mainly increases output diversity. Rescaling PRM logits by a temperature changes the sharpness of a fixed model's output, but it does not sample from the PRM parameter posterior. By contrast, Monte Carlo Dropout evaluates the same reasoning trace under different stochastic realizations of the frozen PRM, which directly targets the epistemic variability of the verifier.

D.7 Scope of the Theoretical Assumptions

The theoretical analysis is intended to isolate the mechanism by which verifier uncertainty induces search errors. The unbiasedness assumption in Proposition 4.2 should therefore be interpreted as an analytical device rather than a claim that real neural PRMs are strictly unbiased. In practice, PRMs can contain systematic biases due to their training data, model architecture, and annotation procedure. UATS does not eliminate such systematic verifier bias. Instead, the theory focuses on the additional degradation caused by variance-induced mis-ranking, showing that explicitly modeling uncertainty can mitigate this source of error.

Similarly, the additive-noise model $R_\phi(h) = R^*(h) + \xi(h)$ is a simplified statistical abstraction for representing score dispersion on uncertain reasoning traces. It separates the latent quality of a trace from the PRM's epistemic fluctuation around that quality, making it possible to analyze how

uncertainty affects ranking. The fixed OOD probability assumption $P(O_t) = \epsilon$ is also an abstraction. It decouples the occurrence of OOD traces from the complex transition dynamics of LLM generation, thereby allowing a tractable regret comparison between uncertainty-agnostic and uncertainty-aware selection.

These assumptions are not meant to fully characterize all failure modes of real external reasoning systems. Rather, they define a controlled setting for studying a specific vulnerability: when a PRM's point estimate is unreliable on OOD reasoning steps, uncertainty-agnostic search can accumulate mis-ranking errors over the reasoning horizon. Under this abstraction, the analysis explains why additional verification of uncertain candidates can change the degradation pattern from linear to sublinear. The empirical results then evaluate whether this mechanism remains useful under realistic frozen-proposer and frozen-PRM settings.

E Implementation Details

E.1 Implementation of UATS

H-UATS is a training-free inference strategy, while A-UATS additionally trains a lightweight budget controller. The hyperparameters are listed in Appendix E.2. For fair comparison, all methods are evaluated under matched inference-time budgets. We calibrate the relative cost of policy generation and PRM evaluation using wall-clock latency on the same A100 hardware. In our implementation, generating one reasoning segment takes 571 ms on average, while one stochastic PRM forward pass takes 32 ms, yielding a latency ratio of $571/32 \approx 17.8$. Thus, we count one generation step as approximately 18 stochastic PRM passes. Baselines spend this budget on generating more candidates, whereas UATS allocates an equivalent portion to uncertainty estimation and re-evaluation.

For A-UATS, we train the budget controller on the training split of the original MATH dataset Hendrycks et al. (2021). We use LLaMA-3.1-Instruct-8B and Qwen2.5-Instruct-3B as policy models, and Math-Shepherd-PRM-7B and Qwen2.5-Math-PRM-7B as PRMs. In each training round, we sample a question and a Policy–PRM pair, generate 10 complete reasoning trajectories under the current controller, and update the controller using the final-answer reward defined in Section 5.3. The controller is trained for 500 update rounds. MATH-500 and AIME24 are used only for evaluation.

E.2 Implementation of Hyperparameters

We provide a detailed listing of the hyperparameters used for both the Heuristic Uncertainty-Aware Tree Search (H-UATS) and the Adaptive Uncertainty-Aware Tree Search (A-UATS) in Table 3.

Table 3: Hyperparameter settings for H-UATS and A-UATS experiments.

Method	**Hyperparameter**	**Value**
Common	Dropout Rate (p)	$0.07 \sim 0.10$
	Max Candidate Paths (N)	256
	Exploration parameter (α)	0.3
H-UATS	Initial Sampling Count (K_0)	7
	Uncertainty Threshold (τ)	0.003
	Optimism Margin (δ)	0.04
	Re-evaluation Temperature (ν_1)	0.5
	Expansion Temperature (ν_2)	0.2
A-UATS	Learning Rate	1×10^{-4}
	Optimizer	AdamW
	Cost Penalty (λ)	0.05

F Full Results

F.1 Visualization of Full Results

The full results of all policy models and PRMs in the MATH-500 dataset are provided in Figure 5, and the results of AIME are provided in Figure 6.

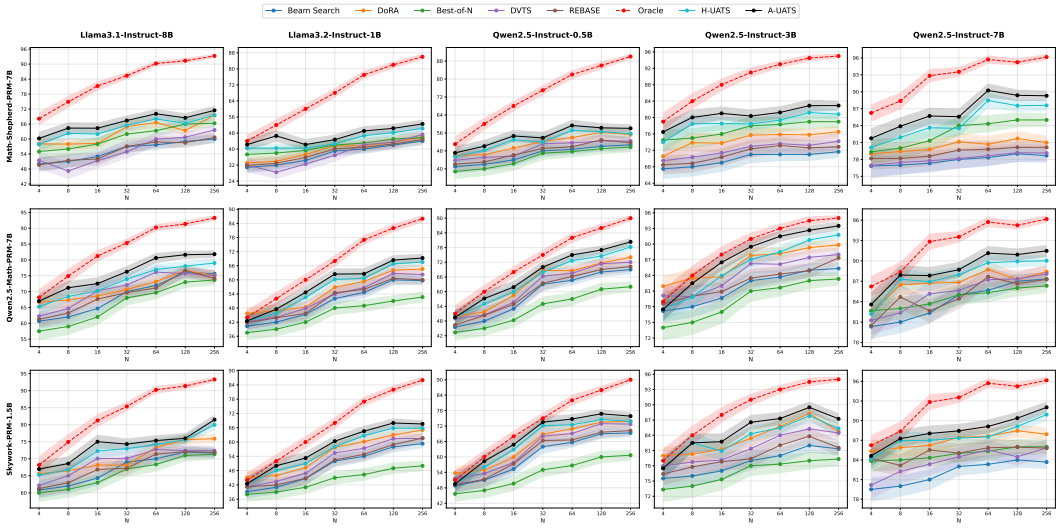


Figure 5: Full results on the MATH-500 dataset.

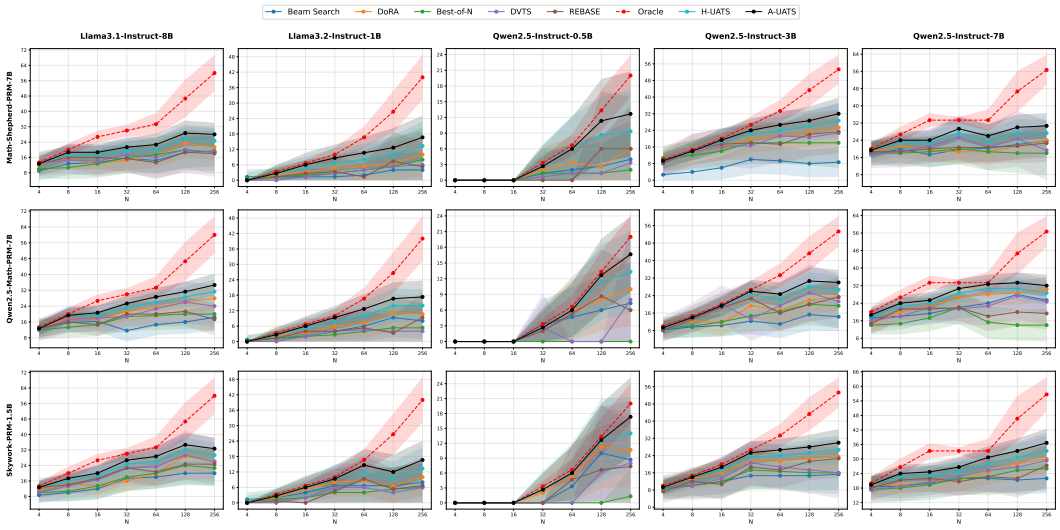


Figure 6: Full results on the AIME24 dataset.

F.2 Table of Full Results

Table 4 and Table 5 summarize the average performance across all 15 model combinations, averaged over all budget levels $N \in \{4, 8, 16, 32, 64, 128, 256\}$. On AIME24, H-UATS achieves an average accuracy of 15.87%, surpassing the strongest baseline DoRA (13.58%) by 2.29 points. A-UATS further improves the accuracy to 17.37%, yielding gains of 3.79 points over DoRA and 1.50 points over H-UATS. A similar trend is observed on MATH-500, where H-UATS (70.58%) and A-UATS (72.39%) consistently outperform the strongest baseline DoRA (68.82%). Specifically, A-UATS improves over DoRA by 3.57 points and over H-UATS by 1.81 points. These aggregated results show

Table 4: Average performance across all number of N for different model pairs on the MATH dataset.

Model Pair		MATH						
Policy	Reward	Beam Search	Best-of-N	DoRA	DVTS	REBASE	H-UATS	A-UATS
Llama3.1-Instruct-8B	Math-Shepherd-PRM-7B	55.38 ± 2.27	60.95 ± 2.43	62.67 ± 2.08	55.59 ± 2.81	55.45 ± 2.49	64.48 ± 2.22	**66.64 ± 2.10**
Llama3.1-Instruct-8B	Qwen2.5-Math-PRM-7B	68.72 ± 2.02	66.12 ± 2.32	71.19 ± 1.93	70.87 ± 2.34	69.11 ± 2.15	73.14 ± 1.87	**75.88 ± 1.78**
Llama3.1-Instruct-8B	Skywork-PRM-1.5B	67.12 ± 1.96	65.95 ± 2.18	70.46 ± 1.90	69.26 ± 2.24	67.66 ± 2.09	72.43 ± 1.88	**73.96 ± 1.80**
Llama3.2-Instruct-1B	Math-Shepherd-PRM-7B	37.45 ± 2.51	41.48 ± 2.81	39.58 ± 2.53	37.52 ± 3.10	38.40 ± 2.69	44.15 ± 2.15	**46.80 ± 2.57**
Llama3.2-Instruct-1B	Qwen2.5-Math-PRM-7B	50.57 ± 2.50	45.60 ± 2.80	55.12 ± 2.37	52.81 ± 2.88	51.77 ± 2.62	56.57 ± 2.47	**58.19 ± 2.44**
Llama3.2-Instruct-1B	Skywork-PRM-1.5B	49.74 ± 2.51	44.05 ± 2.70	55.17 ± 2.35	52.47 ± 2.79	50.83 ± 2.56	56.27 ± 2.42	**58.05 ± 2.35**
Qwen2.5-Instruct-0.5B	Math-Shepherd-PRM-7B	46.33 ± 2.54	44.93 ± 2.89	51.30 ± 2.39	48.87 ± 2.88	47.82 ± 2.78	52.57 ± 2.62	**54.46 ± 2.53**
Qwen2.5-Instruct-0.5B	Qwen2.5-Math-PRM-7B	58.74 ± 2.54	53.09 ± 2.84	63.29 ± 2.47	61.77 ± 2.84	60.04 ± 2.74	65.40 ± 2.48	**67.26 ± 2.37**
Qwen2.5-Instruct-0.5B	Skywork-PRM-1.5B	60.45 ± 2.44	53.53 ± 2.69	65.09 ± 2.29	63.71 ± 2.80	61.52 ± 2.60	65.94 ± 2.40	**67.64 ± 2.31**
Qwen2.5-Instruct-3B	Math-Shepherd-PRM-7B	69.88 ± 2.00	77.12 ± 2.09	74.59 ± 1.97	72.19 ± 2.35	71.25 ± 2.05	78.70 ± 1.77	**80.71 ± 1.67**
Qwen2.5-Instruct-3B	Qwen2.5-Math-PRM-7B	81.69 ± 1.66	79.29 ± 1.96	86.33 ± 1.50	84.27 ± 1.71	82.79 ± 1.72	85.63 ± 1.53	**87.66 ± 1.40**
Qwen2.5-Instruct-3B	Skywork-PRM-1.5B	78.69 ± 1.74	76.76 ± 1.98	83.39 ± 1.52	81.56 ± 1.91	79.96 ± 1.80	83.40 ± 1.62	**84.76 ± 1.52**
Qwen2.5-Instruct-7B	Math-Shepherd-PRM-7B	77.88 ± 1.59	82.71 ± 1.69	80.38 ± 1.52	78.19 ± 1.88	79.24 ± 1.59	84.67 ± 1.47	**86.54 ± 1.36**
Qwen2.5-Instruct-7B	Qwen2.5-Math-PRM-7B	84.09 ± 1.40	84.57 ± 1.46	86.72 ± 1.31	85.29 ± 1.61	84.83 ± 1.57	87.72 ± 1.26	**88.80 ± 1.21**
Qwen2.5-Instruct-7B	Skywork-PRM-1.5B	82.07 ± 1.44	84.93 ± 1.38	86.98 ± 1.25	83.73 ± 1.62	85.08 ± 1.40	87.57 ± 1.17	**88.55 ± 1.16**

Table 5: Average performance across all number of N for different model pairs on the AIME dataset.

Model Pair		AIME						
Policy	Reward	Beam Search	Best-of-N	DoRA	DVTS	REBASE	H-UATS	A-UATS
Llama3.1-Instruct-8B	Math-Shepherd-PRM-7B	14.71 ± 6.65	15.05 ± 8.13	16.67 ± 6.03	16.19 ± 9.49	15.71 ± 7.56	19.62 ± 5.54	**21.52 ± 5.10**
Llama3.1-Instruct-8B	Qwen2.5-Math-PRM-7B	15.05 ± 5.46	16.86 ± 6.61	20.95 ± 5.10	19.81 ± 8.16	17.43 ± 6.23	22.95 ± 4.70	**24.67 ± 4.29**
Llama3.1-Instruct-8B	Skywork-PRM-1.5B	15.14 ± 5.49	16.95 ± 7.28	18.57 ± 5.47	20.19 ± 9.01	19.53 ± 6.62	22.76 ± 5.08	**24.67 ± 4.64**
Llama3.2-Instruct-1B	Math-Shepherd-PRM-7B	2.19 ± 4.44	3.43 ± 6.85	4.57 ± 5.09	2.76 ± 6.48	3.05 ± 5.84	6.48 ± 5.24	**8.19 ± 5.05**
Llama3.2-Instruct-1B	Qwen2.5-Math-PRM-7B	4.48 ± 4.79	3.05 ± 5.36	5.90 ± 4.65	2.67 ± 5.14	3.90 ± 5.21	7.71 ± 4.31	**9.24 ± 4.19**
Llama3.2-Instruct-1B	Skywork-PRM-1.5B	5.05 ± 5.50	3.33 ± 5.90	5.81 ± 4.85	3.43 ± 6.41	3.81 ± 4.92	7.52 ± 4.90	**8.76 ± 4.61**
Qwen2.5-Instruct-0.5B	Math-Shepherd-PRM-7B	1.43 ± 3.16	0.86 ± 3.34	2.00 ± 3.27	0.95 ± 3.92	1.71 ± 2.98	3.81 ± 3.69	**4.67 ± 3.73**
Qwen2.5-Instruct-0.5B	Qwen2.5-Math-PRM-7B	3.05 ± 3.57	0.00 ± 0.00	3.81 ± 3.27	1.52 ± 2.97	3.14 ± 4.18	4.76 ± 3.43	**5.43 ± 3.23**
Qwen2.5-Instruct-0.5B	Skywork-PRM-1.5B	3.14 ± 3.34	0.19 ± 0.80	4.19 ± 3.67	2.05 ± 3.65	2.67 ± 3.68	5.05 ± 3.64	**5.52 ± 3.44**
Qwen2.5-Instruct-3B	Math-Shepherd-PRM-7B	6.95 ± 5.84	15.52 ± 9.14	18.47 ± 6.62	17.14 ± 10.76	17.24 ± 8.56	20.38 ± 6.09	**22.00 ± 5.68**
Qwen2.5-Instruct-3B	Qwen2.5-Math-PRM-7B	11.62 ± 5.70	14.57 ± 7.42	16.47 ± 5.79	17.43 ± 9.62	17.43 ± 7.31	20.57 ± 5.26	**22.00 ± 4.82**
Qwen2.5-Instruct-3B	Skywork-PRM-1.5B	12.95 ± 5.87	13.81 ± 7.75	18.28 ± 5.74	15.05 ± 9.23	15.71 ± 7.65	20.00 ± 5.25	**21.71 ± 4.86**
Qwen2.5-Instruct-7B	Math-Shepherd-PRM-7B	20.02 ± 6.43	18.48 ± 7.76	20.29 ± 5.73	21.05 ± 9.91	20.38 ± 6.80	24.29 ± 5.43	**26.19 ± 5.05**
Qwen2.5-Instruct-7B	Qwen2.5-Math-PRM-7B	22.00 ± 5.45	15.90 ± 6.40	24.43 ± 4.88	21.71 ± 7.83	19.52 ± 5.97	26.19 ± 4.49	**28.10 ± 4.11**
Qwen2.5-Instruct-7B	Skywork-PRM-1.5B	20.67 ± 5.30	21.52 ± 7.02	23.33 ± 5.01	22.95 ± 8.11	22.00 ± 6.44	26.00 ± 4.65	**27.90 ± 4.34**

that explicitly modeling uncertainty improves robustness over point-estimate search methods, while adaptive budget allocation further improves the utility of the fixed inference-time compute budget across diverse reasoning scenarios.

F.3 FLOPs-based Evaluation

In the main text, we report final-answer accuracy as a function of the candidate budget N, following standard test-time scaling protocols. To further evaluate compute efficiency, we additionally compare all methods under FLOPs-matched inference budgets. For each method, we estimate the total inference cost by accounting for both policy-model generation and PRM evaluation. The same cost model is used for all baselines and UATS variants, so the x-axis reflects comparable computational budgets rather than the raw number of generated candidates.

Figures 7 and 8 show the complete FLOPs-based results on MATH-500 and AIME24 across all 15 Policy–PRM combinations. The trends are consistent with the candidate-budget results in the main text: H-UATS improves over uncertainty-agnostic search baselines in most settings, and A-UATS further achieves the strongest or near-strongest performance under comparable FLOPs. This indicates that the gains of UATS are not merely due to using more inference-time computation, but come from allocating the fixed budget more effectively between verification and expansion.

G Additional Analysis

G.1 Calibration analysis

We further examine whether Monte Carlo Dropout variance reflects the reliability of PRM scores. Using Qwen2.5-Instruct-7B as the policy model and Math-Shepherd-PRM-7B as the verifier on MATH-500, we compute the Expected Calibration Error (ECE) and Brier score over 10 runs. As shown in Figure 9a, the PRM exhibits non-negligible calibration error. We then divide reasoning steps into low-, mid-, and high-variance groups according to their Monte Carlo Dropout score variance. Figure 9b shows that both ECE and Brier score increase with variance: ECE rises from around 0.08 to 0.13, while the Brier score increases from around 0.09 to 0.145. This indicates that high Monte

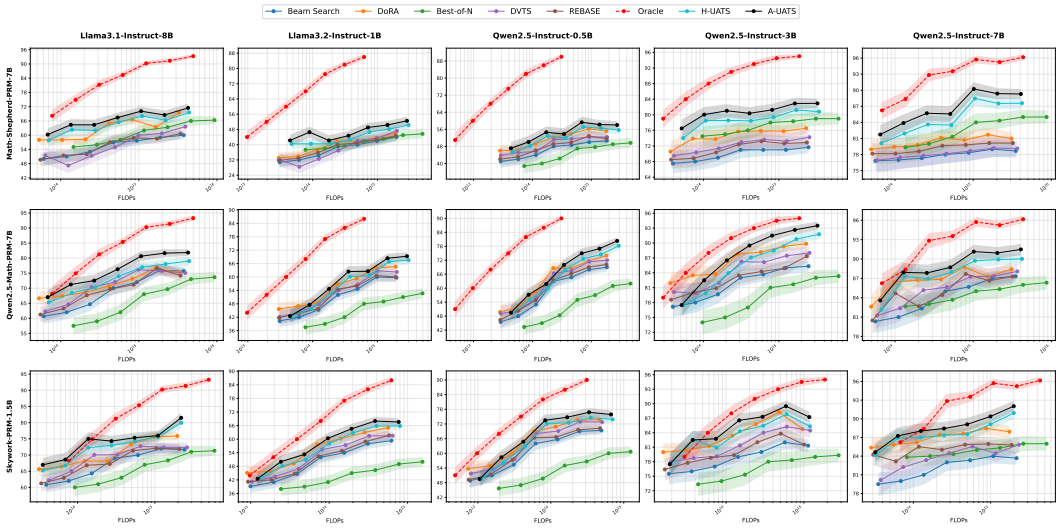


Figure 7: FLOPs-based evaluation on MATH-500 across all Policy–PRM combinations. The x-axis denotes the estimated total inference FLOPs, and the y-axis denotes final-answer accuracy. Shaded regions denote the standard deviation over 5 runs.

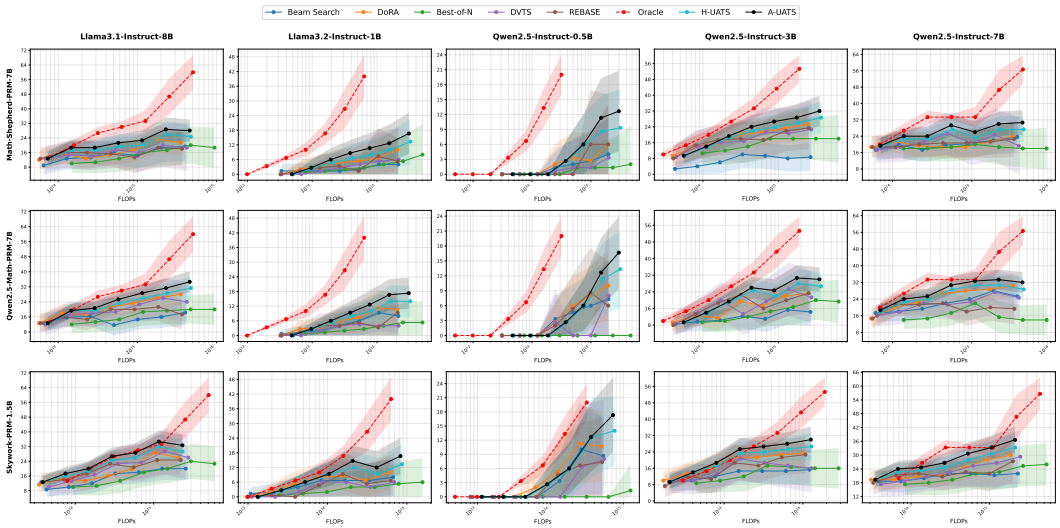
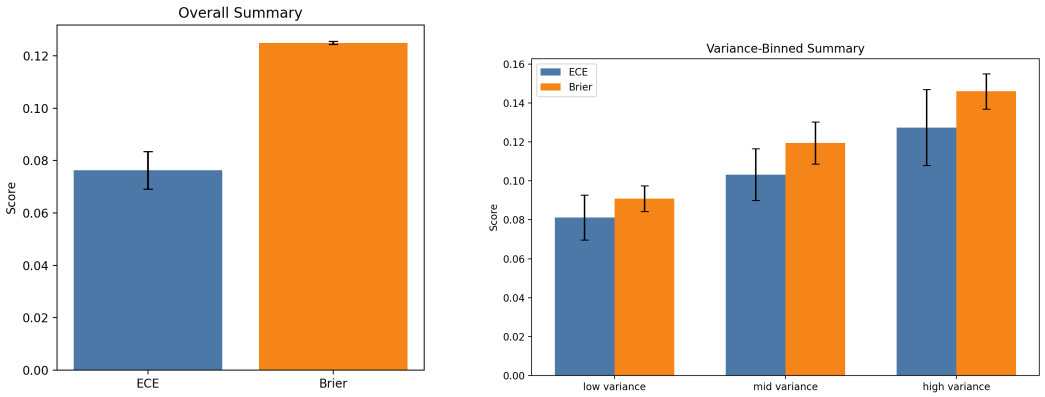


Figure 8: FLOPs-based evaluation on AIME24 across all Policy–PRM combinations. The x-axis denotes the estimated total inference FLOPs, and the y-axis denotes final-answer accuracy. Shaded regions denote the standard deviation over 5 runs.

Carlo Dropout variance effectively identifies regions where PRM scores are less calibrated and thus less reliable.

G.2 Uncertainty along reasoning depth

A potential concern is that high uncertainty may simply arise at early reasoning steps due to short prefixes. To examine this, we group PRM evaluations by reasoning depth and compute the average Monte Carlo Dropout score variance at each depth. As shown in Figure 10, the earliest step has the lowest average variance. The variance increases over the first few steps and remains elevated across most middle-to-late depths. This suggests that the observed uncertainty is not merely an artifact of short prefixes, but is more consistent with uncertainty accumulating as reasoning traces become longer and more diverse.



(a) Overall calibration

(b) Variance-binned calibration

Figure 9: Calibration analysis of Math-Shepherd-PRM-7B on MATH-500 using Qwen2.5-Instruct-7B as the policy model. (a) Overall calibration results, where ECE and Brier score are averaged over 10 runs. (b) Variance-binned calibration results. Reasoning steps with higher Monte Carlo Dropout variance exhibit larger ECE and Brier scores, suggesting that variance is correlated with PRM miscalibration.

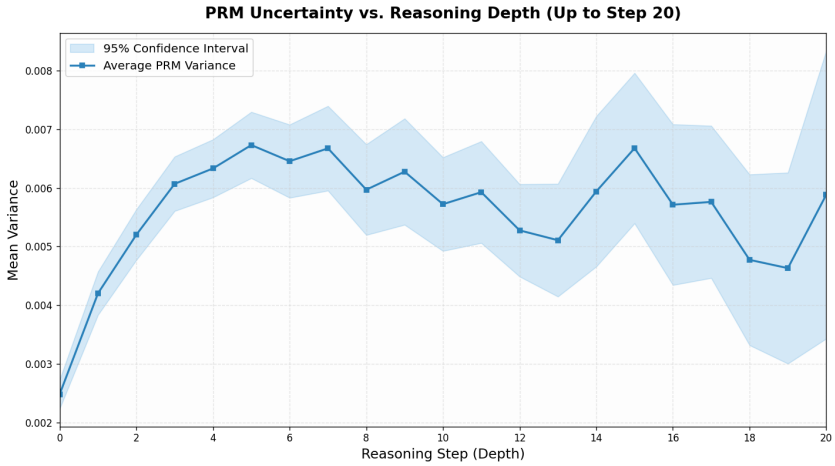
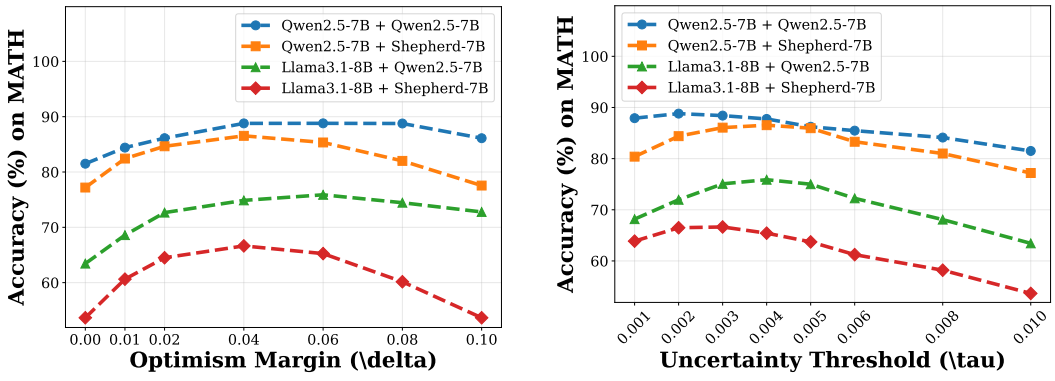


Figure 10: PRM uncertainty across reasoning depths. The shaded region denotes the 95% confidence interval. The earliest reasoning step has the lowest variance, while uncertainty increases and remains elevated in later steps.

G.3 Ablation of Uncertainty Threshold

The uncertainty threshold τ plays a pivotal role in distinguishing between ID and OOD reasoning steps. It acts as a gatekeeper, determining which candidates warrant expensive re-evaluation. We analyze the sensitivity of H-UATS performance to varying τ. Empirically, setting τ too low results in an overly aggressive classification, where even stable ID steps are flagged as uncertain, causing the system to waste its limited evaluation budget on unnecessary verifications. Conversely, a high τ fails to capture genuine epistemic uncertainties, allowing hallucinated or high-error OOD steps to propagate through the search tree. Our results, visualized in Figure 11b, indicate that an optimal trade-off is achieved around $\tau \approx 0.003$, where the system effectively filters high-risk nodes while maintaining search efficiency.



(a) Optimism Margin δ

(b) Uncertainty Threshold τ

Figure 11: Ablation studies for τ and δ.

G.4 Ablation of Optimism Margin

The optimism margin δ defines the degree of leniency extended to potentially underestimated candidates during the filtering phase. It allows the system to rescue steps that have low initial PRM scores but high uncertainty. Increasing δ encourages exploration by retaining more uncertain candidates for re-evaluation. Figure 11a shows that excessive optimism (large δ) introduces significant noise, as the system begins to invest resources in truly incorrect steps. The performance curve suggests an optimal value when $\delta \approx 0.04$.

G.5 Ablation of Network Structure

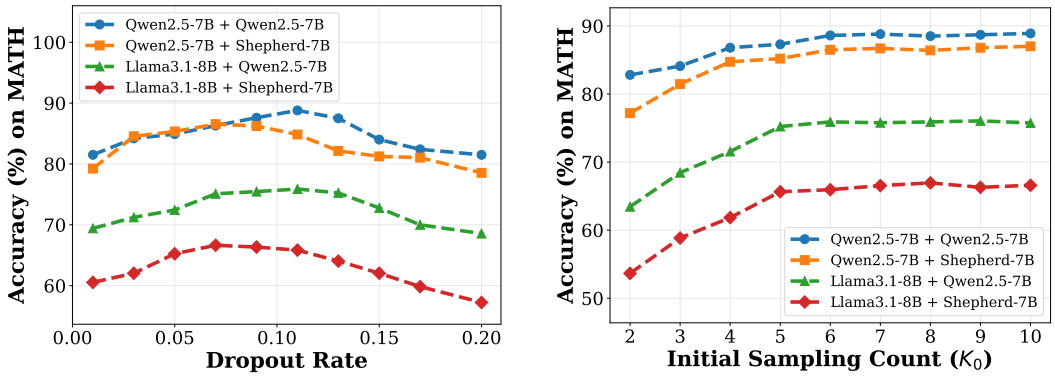
In this section, we investigate how the architecture of the A-UATS controller affects performance. We perform an ablation study on the MATH-500 dataset using Qwen2.5-Instruct-7B as the policy model and Math-Shepherd-PRM-7B as PRM. For each configuration, we report the average accuracy across all reward models and compute budgets. Specifically, we vary the number of layers and hidden dimensions of the controller MLP to assess their impact on overall performance.

Table 6: Mean accuracy of A-UATS on MATH-500 under different network architectures. Each result is averaged over all compute budgets.

Network Layers	Network Dim	Accuracy (%)
2	128	84.73
2	256	**86.54**
2	512	85.12
3	128	86.32
3	256	85.61
3	512	85.03

G.6 Ablation of Dropout Rate

We evaluate the effect of the dropout rate p by varying it from 0.01 to 0.20 across four Policy–PRM combinations, involving the policy models Qwen2.5-Instruct-7B and LLaMA3.1-Instruct-8B, and the PRMs Qwen2.5-Math-PRM-7B and Math-Shepherd-PRM-7B. This parameter controls the stochasticity of the uncertainty estimates. As illustrated in Figure 12a, insufficient noise with a lower p fails to capture epistemic uncertainty, while excessive noise with higher p destabilizes the reward signal. Notably, the optimal rate depends on the verifier's architecture; Qwen2.5-Math-PRM peaks at $p \approx 0.10$, whereas Math-Shepherd-PRM requires a more conservative $p \approx 0.07$, reflecting different tolerance levels to feature perturbations.



(a) Ablation on dropout rate

(b) Ablation on K_0

Figure 12: Ablation studies on: (a) Dropout rate. (b) Initial Sampling Count for uncertainty estimation.

G.7 Ablation of Initial Sampling Count

We evaluate the initial sampling count $K_0 \in [2, 10]$, which establishes the baseline uncertainty for candidate filtering. As shown in Figure 12b, accuracy improves consistently with higher K_0, as more samples reduce the variance of the initial estimators. However, the marginal gains diminish rapidly, with the performance curve plateauing around $K_0 = 7$. This suggests that a modest number of initial passes is sufficient to distinguish ID from OOD candidates.

H Qualitative Results

To provide deeper insights into the behavior of the PRM and the utility of epistemic uncertainty, we analyze four representative reasoning steps encountered during the standard beam search process. These examples, visualized in Figures 13 to 16, illustrate how uncertainty estimates serve as a critical calibration signal, enabling the system to distinguish between reliable judgments and potential distribution shifts.

1. High Score, Low Uncertainty (Figure 13). This figure represents the ideal scenario where the policy model generates a correct, high-quality reasoning step that aligns well with the PRM's training distribution. The PRM confidently assigns a high score with low variance. In this case, both standard and uncertainty-aware methods correctly prioritize this step for extension.

2. High Score, High Uncertainty (Figure 14). Here, the step is actually incorrect, yet the PRM assigns it a high score, likely due to "hallucinated" correctness or superficial plausibility (e.g., correct-looking LaTeX formatting but flawed logic). Crucially, the *high epistemic uncertainty* reveals the PRM's lack of confidence in this OOD input. While a standard greedy search would be misled into selecting this step, our UATS framework uses the high variance signal to downweight its priority or trigger additional sampling, effectively filtering out this potential trap.

3. Low Score, High Uncertainty (Figure 15). This is the most critical scenario demonstrating the value of our method. The step is factually correct but phrased in an unconventional way (OOD), causing the PRM to assign a low score. However, the associated *high uncertainty* indicates that the PRM is unreliable in this region. Unlike standard methods that would prematurely prune this valid path, UATS identifies the high uncertainty and allocates additional compute budget (or applies an optimism bonus), successfully "rescuing" the correct path and allowing the reasoning to proceed to the correct solution.

4. Low Score, Low Uncertainty (Figure 16). This figure shows a clear reasoning error. The PRM correctly identifies it as incorrect with a low score, and the low uncertainty confirms this judgment is robust (in-distribution error). Both baseline and uncertainty-aware methods efficiently prune this step, conserving computational resources for more promising branches.

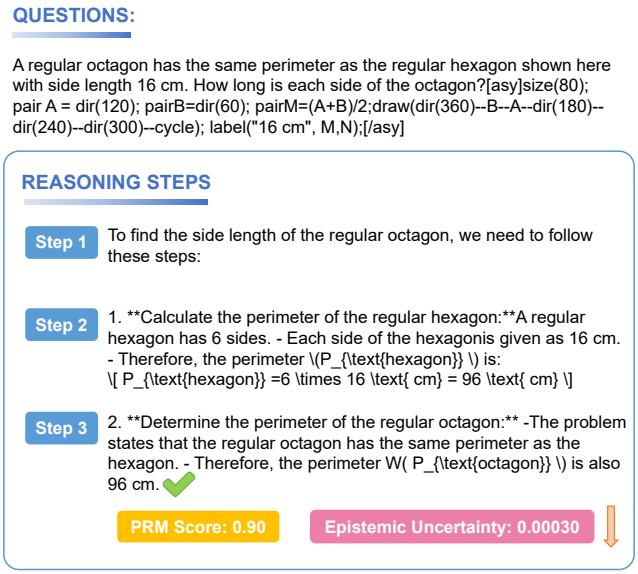


Figure 13: Reasoning step with high PRM score but low epistemic uncertainty.

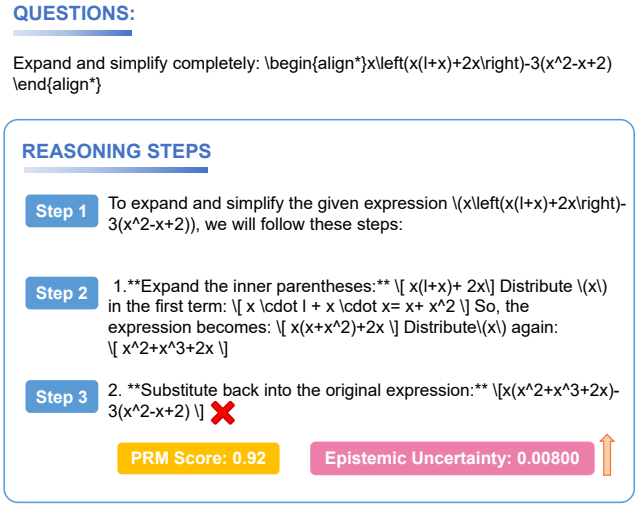


Figure 14: Reasoning step with high PRM score and high epistemic uncertainty.

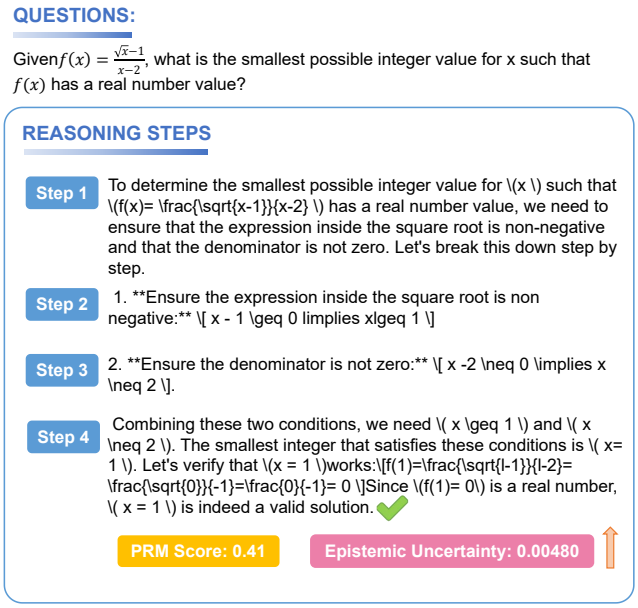


Figure 15: Reasoning step with low PRM score but high epistemic uncertainty.

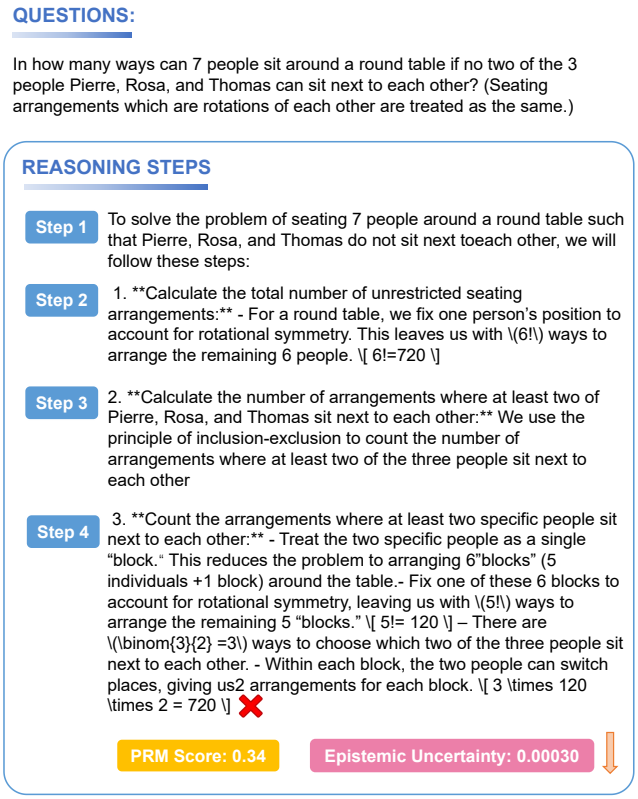


Figure 16: Reasoning step with low PRM score and also low epistemic uncertainty.