ON THE PROVABLE PERFORMANCE GUARANTEE OF EFFICIENT REASONING MODELS

Anonymous authors
Paper under double-blind review

ABSTRACT

Large reasoning models (LRMs) have achieved remarkable progress in complex problem-solving tasks. Despite this success, LRMs typically suffer from high computational costs during deployment, highlighting a need for efficient inference. A practical direction of efficiency improvement is to switch the LRM between thinking and non-thinking modes dynamically. However, such approaches often introduce additional reasoning errors and lack statistical guarantees for the performance loss, which are critical for high-stakes applications. In this work, we propose *Probably Approximately Correct* (PAC) *reasoning* that controls the performance loss under the user-specified tolerance. Specifically, we construct an upper confidence bound on the performance loss and determine a threshold for switching to the non-thinking model. Theoretically, using the threshold to switch between the thinking and non-thinking modes ensures bounded performance loss in a distribution-free manner. Our comprehensive experiments on reasoning benchmarks show that the proposed method can save computational budgets and control the user-specified performance loss.

1 INTRODUCTION

Large reasoning models (LRMs) have shown strong performance in tackling complex problems (DeepSeek-AI et al., 2025; Yang et al., 2025a). However, this strong performance largely depends on long reasoning chains, which substantially increase the computational cost during inference. This phenomenon, often referred to as overthinking (Yue et al., 2025), is evident in mathematical and logic-intensive tasks. And, in applications requiring real-time interaction or large-scale processing, such as text generation (Zhang et al., 2022) and chatbot (Roller et al., 2021), inference efficiency directly determines usability and user experience. Therefore, it is essential to improve the inference efficiency of LRMs.

To address this, some existing works proposed to switch the LRM into a non-thinking mode to avoid overthinking (Chung et al., 2025; Fang et al., 2025; Li et al., 2025b; Liang et al., 2025; Ma et al., 2025; Paliotta et al., 2025; Pan et al., 2025; Xiao & Gan, 2025; Yong et al., 2025). While effective in reducing computational demands, using a non-thinking model often degrades solution quality or introduces additional errors. For instance, in theorem-proving tasks, switching techniques may lead to invalid logical steps, and in mathematical reasoning, it can result in calculation mistakes or overlooked solution paths. Besides, such methods lack a rigorous theoretical guarantee for performance loss. This limitation raises a fundamental issue:

How to improve the efficiency of LRMs, controlling the performance loss provably?

In this work, we formalize this challenge by the concept of a PAC efficient model, where an LRM provides probabilistic guarantees that its performance loss relative to a reference thinking model remains within a user-specified tolerance in Definition 2.1. To achieve this goal, we propose **PAC reasoning**, which constructs a composite model $\hat{f}$ by adaptively routing each input between a high-cost thinking model f and a cheaper non-thinking model $\tilde{f}$. Specifically, PAC reasoning determines an uncertainty threshold on a calibration dataset via a calibration procedure (Algorithm 1). During deployment (Algorithm 2), the composite model uses $\tilde{f}$ for inputs whose uncertainty score falls below the calibrated threshold, and defers to f otherwise. By calibrating the switching rule with respect to

an explicit loss tolerance and confidence level, PAC reasoning provides rigorous statistical guarantees on the resulting performance loss.

Theoretically, we show that PAC reasoning provides distribution-free control of performance loss with high-probability guarantees. We formalize this through a composite model $\hat{f}$, for which the performance loss is non-decreasing with respect to the uncertainty threshold. This monotonicity allows the identification of the largest feasible threshold via an upper confidence bound on the performance loss. Under the i.i.d. assumption, we prove that PAC reasoning controls the loss below a user-specified tolerance with high probability, thereby satisfying PAC efficiency.

We then present comprehensive experimental results in Section 4 that rigorously evaluate the PAC reasoning across diverse reasoning benchmarks, including MATH-500 (Lightman et al., 2023), ZebraLogic(Lin et al., 2025), and Arena-Hard (Li et al., 2025a). The results demonstrate that our approach effectively controls the performance loss and significantly reduces inference cost. For example, on Arena-Hard with tolerance $\epsilon = 0.08$ for the logits uncertainty score, our method controls the average empirical performance loss at 0.06 (below the tolerance), and achieves token savings exceeding 40%. We also find that the logits-based uncertainty score provides more stable performance loss control compared to the verbalized-based score. [1]

Our contributions are as follows:

- We introduce (ε, α)**-PAC efficient**, the first formal framework for quantifying and guaranteeing performance loss in LRM efficiency improvement, establishing a novel theoretical foundation for this domain.
- We propose **PAC reasoning**, a method that combines a thinking-mode model with its non-thinking counterpart via an uncertainty-based mechanism to improve efficiency. The method is model-agnostic and provides *distribution-free* performance guarantees.
- We provide comprehensive experiments on mathematical reasoning, logical deduction, and text generation, demonstrating that PAC reasoning achieves efficiency gains while satisfying the statistical validity of the PAC efficient guarantee.

Notations We begin by introducing key notations. The first is the LRM with thinking-mode f, which is computationally expensive but delivers high performance on its answers. Given an input prompt x, f produces an output $y = f(x)$, which we regard as the "expert answer". The second is the non-thinking LRM $\tilde{f}$, which is computationally cheaper but less accurate, and $\tilde{y} = \tilde{f}(x)$. And for any input x, we use y^{gold} as its "gold reference". Let $\mathcal{I}_{cal} = \{1, \ldots, n\}$ and $\mathcal{I}_{test} = \{n+1, \ldots, n+N\}$ denote the indices of the calibration and test sets, respectively. We define the calibration dataset and the test dataset as:

$$\mathcal{D}_{cal} = \{(x_i, y_i)\}_{i \in \mathcal{I}_{cal}}, \quad \mathcal{D}_{test} = \{(x_i, y_i)\}_{i \in \mathcal{I}_{test}}.$$

It is worth noting that the y_i is not a ground-truth label, but the "expert answer" provided by the LRM f. Finally, let $y_i = (y_{i,1}, \ldots, y_{i,l_{y_i}})$ denote an answer consisting of l_{y_i} tokens, with $y_{i,j}$ representing the j-th token of y_i.

2 Probably Approximately Correct Reasoning

2.1 PAC efficient model

We aim to build a efficient LRM, denoted by $\hat{f}$, that provides probably approximately correct guarantees for its performance loss while improving efficiency. Specifically, given an error tolerance ϵ and a confidence level $1 - \alpha$, $\hat{f}$ ensures its performance loss *relative to the only thinking-mode* LRM f does not exceed ϵ with probability at least $1 - \alpha$. We formulate the PAC guaranteed $\hat{f}$ as follows:

Definition 2.1 ((ϵ, α)-PAC efficient)**.** An LRM $\hat{f}$ is called an (ϵ, α)-probably approximately correct (PAC) efficient model (with respect to loss ℓ) [2], if for given error tolerance $\epsilon > 0$, confidence level

[1] The reproducibility code is placed at an anonymous link: https://anonymous.4open.science/r/pac_reasoning-BD64

[2] For simplicity, we often omit mentioning "with respect to ℓ" since most tasks have their conventional loss functions

$\alpha \in (0, 1)$, it satisfies

$$\mathbb{P}\left(R(\hat{f}) \leq \epsilon\right) \geq 1 - \alpha,$$

where $R(\hat{f}) = \mathbb{E}_{x\sim P}[\ell(\hat{f}(x), f(x))]$ is the risk function, $\ell(\cdot, \cdot)$ is a loss function, x denotes an input prompt drawn from the underlying test sample distribution P.

Remark 2.2. The loss function can be a 0-1 loss for verifiable tasks or a semantic loss for generative tasks. We sometimes term (ϵ, α)-PAC efficient model simply as PAC efficient model. In practice, ϵ controls the performance loss relative to the thinking model f. Therefore, a meaningful range is $\epsilon \in (0, R(\tilde{f}))$, where $R(\tilde{f})$ is the risk of always using the non-thinking model.

2.2 PAC REASONING

Constructing such a controllable LRM $\hat{f}$ is straightforward intuitively. Given an LRM with thinking mode f and a fast LRM without thinking $\tilde{f}$, we create an intermediate model that selectively uses either the LRM with thinking or not based on certain conditions. This condition acts like a "sliding rheostat" that allows us to tune the performance trade-off by adjusting the "position" of the intermediate. We can obtain a model $\hat{f}$ that achieves the desired error tolerance heuristically. However, this approach lacks statistical guarantees on the underlying distribution of performance loss. To build a model with statistical guarantees, a hypothesis test will be used to determine an optimal threshold that balances computational efficiency with output quality while maintaining statistical confidence.

Motivated by this, we present the **PAC reasoning**, which constructs a composite LRM $\hat{f}$ that improves the efficiency of an LRM with thinking f. The composite LRM $\hat{f}$ provides PAC guarantees for the efficiency improvement. The core idea is to use **uncertainty scores** to build an **upper confidence bound** for the performance loss. We could use the upper confidence bound to measure the uncertainty of performance loss for each value of the uncertainty score. Then we **calibrate** an uncertainty threshold to switch between the thinking and non-thinking models. We use the non-thinking LRM $\tilde{f}$ on most inputs and strategically invoke the expensive LRM with thinking f only for inputs whose generation by $\tilde{f}$ has high uncertainty. Mathematically, we could represent the composite model $\hat{f}$ as follows:

$$\hat{f}(x) \equiv \hat{f}_u(x) = \begin{cases} \tilde{f}(x), & U(x) < \hat{u}, \\ f(x), & U(x) \geq \hat{u}. \end{cases} \tag{1}$$

Next, we provide the details of the PAC reasoning.

Uncertainty scores and empirical performance loss We assume that for each input prompt x_i, the non-thinking LRM $\tilde{f}$ produces an output $\tilde{y}_i$, and that there exists a score $U_i \in [0, 1]$ to quantify its uncertainty. This score should ideally correlate with the likelihood of disagreement with the reference model f. The core idea is to use these uncertainty scores to selectively use the expensive model with thinking f. We aim to find a threshold, $\hat{u}$, and accept the non-thinking LRM's output $\tilde{y}_i$ for the instances such that $U_i < \hat{u}$, while querying the model with thinking f for the cases where $U_i \geq \hat{u}$. To formalize, we define the empirical performance loss function as:

$$L(u) = \frac{1}{N} \sum_{i=n+1}^{n+N} \ell(y_i, \tilde{y}_i)\mathbf{1}\{U_i \leq u\}. \tag{2}$$

This function measures the average performance loss for test data points with uncertainty scores no greater than u. If we could compute $L(u)$ for all u, we would choose the largest threshold u^* such that $L(u^*) \leq \epsilon$. However, computing $L(u)$ requires access to all expert answers $y_i = f(x_i)$ in the test set, which is computationally expensive. We try an alternative way to build a bound $\hat{L}_u(\alpha)$ for $\mathbb{E}L(u)$ satisfying the following inequality pointwise w.r.t. α:

$$\mathbb{P}(\hat{L}_u(\alpha) \geq \mathbb{E}L(u)) \geq 1 - \alpha. \tag{3}$$

The bound is also called as upper confidence bound in literature. By construction, $L(u)$ is non-decreasing in u since ℓ is non-negative and $\mathbf{1}\{U_i \leq u\}$ only expands as u increases. This monotonicity lets us obtain the PAC guarantee in Definition 2.1 and apply fixed-sequence single-start without additional corrections; we summarize it in Theorem 3.2.

Algorithm 1 CLT-based UCB for $\hat{L}_u(\alpha)$

Input: Calibration set $\mathcal{S}_{\text{cal}} = \{(x_i, y_i)\}_{i=1}^n$, models f and $\tilde{f}$, uncertainty scores $\{U_i\}_{i=1}^n$, sampling weights $\{\pi_i\}_{i=1}^n$, sample size m, threshold u, confidence level α

Output: Confidence upper bound $\hat{L}_u(\alpha)$

Initialize an empty list of samples $\mathcal{Z} \leftarrow []$.
$\tilde{y}_i \leftarrow \tilde{f}(x_i)$ for all $i = 1, \ldots, n$.
for $j = 1, \ldots, m$ **do**
  Sample an index $i_j \sim \text{Unif}(\{1, \ldots, n\})$.
  Sample a Bernoulli random variable $\xi_{i_j} \sim \text{Bern}(\pi_{i_j})$.
  if $\xi_{i_j} = 1$ **then**
    Query the true label y_{i_j} and compute the importance-weighted loss $Z_j \leftarrow \ell(y_{i_j}, \tilde{y}_{i_j})/\pi_{i_j}$.
  else
    $Z_j \leftarrow 0$.
  end if
  Append Z_j to $\mathcal{Z}$.
end for
For a threshold u, set $Z_j(u) \leftarrow Z_j \cdot \mathbf{1}\{U_{i_j} \le u\}$ for $j = 1, \ldots, m$.
$\hat{\mu}_Z(u) \leftarrow \frac{1}{m}\sum_{j=1}^m Z_j(u)$
$\hat{\sigma}_Z(u) \leftarrow \sqrt{\frac{1}{m-1}\sum_{j=1}^m (Z_j(u) - \hat{\mu}_Z(u))^2}$
$z_{1-\alpha} \leftarrow (1-\alpha)$-quantile of the standard normal distribution.
Return $\hat{L}_u(\alpha) \leftarrow \hat{\mu}_Z(u) + z_{1-\alpha}\frac{\hat{\sigma}_Z(u)}{\sqrt{m}}$.

Constructing the upper confidence bound (UCB) Next we discuss how to construct a valid UCB, via importance sampling. Given a sampling size m, we first collect m indices $\{i_1, \ldots, i_m\}$ by sampling uniformly with replacement from $\{1, \ldots, n\}$. Then, for each selected index i_j, we decide whether to query its expert answer y_{i_j} by performing a Bernoulli trial $\xi_{i_j} \sim \text{Bern}(\pi_{i_j})$, where $\{\pi_1, \ldots, \pi_n\}$ are sampling weights. This procedure yields a dataset of m i.i.d. random variables:

$$Z_j(u) = \ell(y_{i_j}, \tilde{y}_{i_j})\frac{\xi_{i_j}}{\pi_{i_j}}\mathbf{1}\{U_{i_j} \le u\}. \tag{4}$$

The expectation of $Z_j(u)$ is equals the target quantity $L(u)$, since $\mathbb{E}[\xi_{i_j}/\pi_{i_j}|i_j] = 1$. We can therefore estimate an upper bound for $L(u)$ by computing a confidence interval for the mean of $\{Z_j(u)\}_{j=1}^m$. We formally described it in the central limit theorem (CLT) based Algorithm 1.

Remark 2.3. The procedure in Algorithm 1 uses importance sampling to construct an unbiased estimator for the true performance loss $L(u)$. For any fixed threshold u, the random variables $Z_j(u)$ are i.i.d. with expectation $\mathbb{E}[Z_j(u)] = L(u)$. This holds because the sampling process decouples the choice of index i_j from the decision to query the label y_{i_j}. Given the samples $\{Z_j(u)\}_{j=1}^m$, we can form an upper bound for $L(u)$. Algorithm 1 illustrates this using a CLT-based approach, valid for large m. See Appendix C for discussion about its UCB validation as Assumption 3.1. Alternatively, if the importance-weighted losses are bounded, one could use concentration inequalities like Hoeffding's or Bernstein's inequality to construct a valid confidence bound (Bentkus, 2004; Hao et al., 2019; Hoeffding, 1994; Learned-Miller & Thomas, 2020; Ramdas et al., 2022; Waudby-Smith & Ramdas, 2021; 2024), which may provide better guarantees for smaller sample sizes. We give an example based on Hoeffding's inequality (Hoeffding, 1994) in Algorithm 3 in Appendix B.

Calibration Once the UCB $\hat{L}_u(\alpha)$ is constructed, the threshold $\hat{u}$ is the highest one for which this estimated performance loss bound remains below the tolerance ϵ:

$$\hat{u} = \max\{u \in [0, 1] : \hat{L}_u(\alpha) \le \epsilon\}. \tag{5}$$

We calibrate the $\hat{u}$ on the calibration set $\mathcal{D}_{cal}$, and apply it to the test sample. If the score is larger than $\hat{u}$, we use the thinking model to answer; otherwise, we use the non-thinking mode. This procedure ensures we can accept as many outputs from the non-thinking model $\tilde{f}$ while controlling the overall performance loss with high probability. We summarize the PAC reasoning algorithm in Algorithm 2.

Algorithm 2 PAC Reasoning

Input: Calibration set $\{(x_i, y_i)\}_{i=1}^n$, test prompts $x_i, i \in \mathcal{I}_{test}$, model without thinking $\tilde{f}$, model with thinking f, loss function ℓ, error tolerance ϵ, confidence level α

Output: Composite model $\hat{f}$

1: Compute confidence bound function $\hat{L}_u(\alpha)$ via Algorithm 1 on calibration data $\{(x_i, y_i)\}_{i=1}^n$.
2: Threshold $\hat{u} = \max\{u \in [0,1] : \hat{L}_u(\alpha) \leq \epsilon\}$.
3: Composite model $\hat{f}(x) \leftarrow f(x)\mathbb{1}\{U(x) \geq \hat{u}\} + \tilde{f}(x)\mathbb{1}\{U(x) < \hat{u}\}$.
4: **Return** $\hat{f}$.

3 Theoretical Analysis

In this section, we aim to build the PAC guarantee. Our PAC reasoning builds upon the theoretical foundation established by the distribution-free risk control framework (Angelopoulos et al., 2025a). It provides the mathematical foundation for our risk control type method on the characteristics of PAC reasoning. In detail, if the performance loss $L(u)$ is bounded by a UCB, and the UCB based on CLT or concentration inequality is valid as Assumption 3.1, we can prove the PAC guarantee as follows.

First, noting that while the confidence bound $\hat{L}_u(\alpha)$ is constructed to hold for a single, pre-specified threshold u, our algorithm selects the threshold $\hat{u}$ based on the calibration data. Let $\mathcal{D}_{cal}$ be a calibration set, used to construct a threshold $\hat{u}$, and let $\mathcal{D}_{test}$ be an independent test set with i.i.d. samples as $\mathcal{D}_{cal}$. For any threshold u, recalling the deployment strategy in Eq. (1), we denote it by $\hat{f}_u(x)$. We could see that $\hat{f} = \hat{f}_{\hat{u}}$ is the composite model. We re-parameterize $\hat{f}$ and its risk with respect to u, with a slight abuse of notation. Its population risk $R(\hat{f})$ is re-parameterized as:

$$R(u) = \mathbb{E}[\ell(y, \hat{f}_u(x))],$$

and the empirical risk is re-parameterized as:

$$\widehat{R}(u) = \frac{1}{N} \sum_{i \in \mathcal{I}_{test}} \ell(y_i, \hat{f}_u(x_i)).$$

Then we list the assumptions of the PAC reasoning.

Assumption 3.1 (UCB validity)**.** For each threshold u and any $\alpha \in (0,1)$, the UCB $\widehat{L}_u(\alpha)$, computed on $\mathcal{D}_{cal}$, satisfies

$$\mathbb{P}\big(R(u) \leq \widehat{L}_u(\alpha)\big) \geq 1 - \alpha.$$

As discussed in Remark 2.3, we can build the UCB $\widehat{L}_u(\alpha)$ for $\mathbb{E}L(u)$ in two main ways: using the central limit theorem, or using bounds like Hoeffding's or Bernstein's inequality (Bentkus, 2004; Hoeffding, 1994; Waudby-Smith & Ramdas, 2021; 2024), which may provide better guarantees for smaller sample sizes. We prove the validity of the CLT-based method in Appendix C. The risk function $R(u)$ is non-decreasing with u by construction. As u increases, the condition $U(x) \geq u$ becomes harder to satisfy, so we defer to the expert less often. Then, because $\ell(y, y) = 0$ and $\ell \geq 0$, reducing deference to the expert can only increase the total risk. We combine UCB validity and this monotonicity, and provide the PAC guarantee of our proposed method:

Theorem 3.2 (PAC guarantee)**.** *Let $\hat{u}$ be the threshold selected by the PAC reasoning algorithm (Algorithm 2). If calibration set and test set are i.i.d. and Assumption 3.1 holds, then the composite model $\hat{f}$ constructed by Algorithm 2 satisfies the (ϵ, α)-PAC guarantee, i.e.,*

$$\mathbb{P}(R(\hat{f}) \leq \epsilon) \geq 1 - \alpha.$$

Proof sketch. The proof uses the monotonicity of the risk $R(u)$. If the selected threshold $\hat{u}$ is larger than the oracle threshold $u^\star := \inf\{u : R(u) > \epsilon\}$, then the upper bound $\widehat{L}_{u^\star}$ must be less than $R(u^\star)$. This contradicts Assumption 3.1, which happens with probability at most α. A detailed proof is in Appendix D. □

If the loss is bound in $[a, b]$, we provide an empirical version:

Theorem 3.3 (Empirical risk PAC guarantee)**.** *Assume Assumption 3.1 holds, and the test batch $\mathcal{D}_{test}$ is independent of the calibration data $\mathcal{D}_{cal}$. Given $\epsilon, \alpha \in (0, 1)$, $\ell \in [a, b]$, and $\hat{u}$ defined as in Theorem 3.2, then any $t > 0$,*

$$\mathbb{P}\big(\widehat{R}(\hat{u}) \leq \epsilon + t\big) \;\geq\; 1 - \alpha - \exp\Big(-\tfrac{2Nt^2}{(b-a)^2}\Big).$$

Proof sketch. The result follows from the decomposition of the error event: $\{\widehat{R}(\hat{u}) > \epsilon + t\} \subseteq \{R(\hat{u}) > \epsilon\} \cup \{\widehat{R}(\hat{u}) - R(\hat{u}) > t\}$. The first term is bounded by α via Theorem 3.2. The second term is bounded by $\exp(-2Nt^2/(b-a)^2)$ using the conditional Hoeffding's inequality, as the test set is independent of $\hat{u}$. A detailed proof is in Appendix E. □

Remark 3.4. A common special case is a bounded loss $\ell \in [0, 1]$, e.g., 0-1 loss for binary verifiable answers. Then $b - a = 1$ and the bound simplifies to $\mathbb{P}\big(\widehat{R}(\hat{u}) \leq \epsilon + t\big) \geq 1 - \alpha - e^{-2Nt^2}$. It provides exact risk control for $\hat{f}$ with probability at least $1 - \alpha$ by some slacks t.

We prove it in Appendix E. If the i.i.d. assumption does not hold, PAC reasoning can be extended to a transductive setting. This extension is discussed in Appendix F.

4 Experiments

In this section, we evaluate PAC reasoning over multi benchmarks. We study two questions. First, does PAC reasoning control the performance loss at the target tolerance with confidence? Second, how much computation can it save under this control? We report results for two uncertainty scores (logits-based and verbalized) and two efficiency metrics (expert call percentage and saved token percentage). We first show the setup as follows.

4.1 Setup

Large language models In this study, we evaluate the PAC reasoning based on the Qwen3 series models (Yang et al., 2025a) and Llama-3.1-8B–based models. Specifically, we employ the "Qwen3-4B-Thinking-2507" as the thinking model and "Qwen3-4B-Instruct-2507" as the non-thinking model. We also use the "DeepSeek-R1-Distill-Llama-8B" as the thinking model and "Llama-3.1-8B-Instruct" as the non-thinking model, and show the results in Appendix I.1.

Uncertainty score Our theoretical guarantees do not depend on the choice of uncertainty score. Here, we focus on two common uncertainty scores used in LLMs, the **logits-based score** and the **verbalized score**. The logits-based score is a white-box score from model logits. The verbalized score is a black-box score from self-reported confidence. For the white-box score, we use token-level probabilities from the prediction logits (Kwon et al., 2023; Zheng et al., 2024; Zhou et al., 2025). Specifically, we define the uncertainty score of y_i as its average token probability (Hao et al., 2023; Huang et al., 2025):

$$U_{logits}(y_i) = 1 - \frac{1}{l_{y_i}} \sum_{j=1}^{l_{y_i}} \mathbb{P}(y_{i,j} | y_{i,1}, \ldots, y_{i,j-1}, x_i),$$

where $\mathbb{P}(y_{i,j} | y_{i,1}, \ldots, y_{i,j-1}, x_i)$ is the conditional probability of token $y_{i,j}$. Moreover, we use verbalized uncertainty scores from non-thinking models (Xiong et al., 2023; Tian et al., 2023; Yang et al., 2024; Zhou et al., 2025), where the model explicitly states its self-reported confidence. The verbalized uncertainty score is mainly applicable in black-box scenarios, where access to generation logits is restricted, especially in the case of closed-source LLMs. In this study, we report the average confidence over 10 trials, and the corresponding prompts are listed in Appendix H.

Datasets We evaluate PAC reasoning on datasets spanning reasoning and open-ended generation tasks. Specifically, our evaluation covers a high-level mathematics benchmark, MATH-500 (Lightman et al., 2023), a text-based logical reasoning task, ZebraLogic (Lin et al., 2025), and an alignment-focused open-ended writing benchmark, Arena-Hard (Li et al., 2025a). For each dataset, we partition the original test set into a calibration subset and a held-out test subset randomly. Table 3 in Appendix H

provides details on the specific splitting strategies. We also report results on additional benchmarks like GPQA (Rein et al., 2024) and HumanEval (Chen et al., 2021) in Appendix I.2.

Baselines We compare PAC reasoning with several efficiency-oriented baselines, including Naive control, Router Ong et al. (2025), Chain of Draft (CoD) (Xu et al., 2025), and NoThinking (Ma et al., 2025), with details in Appendix H.1. These methods reduce inference cost via heuristic design choices, but lack explicit theoretical guarantees on performance loss.

Loss functions PAC reasoning controls the extra error from switching from the thinking model f to the non-thinking model $\tilde{f}$, i.e., the performance loss caused by accelerating the reasoning procedure. For this reason, we define loss relative to the thinking model, not relative to the gold answer. We use two loss functions in experiments: (1) **Semantic cosine distance** in an embedding space. Given reference output $y_i = f(x_i)$ and PAC output $\hat{y}_i = \hat{f}(x_i)$, let v_{y_i} and $v_{\hat{y}_i}$ be their embeddings. The semantic loss is:

$$\ell(y_i, \hat{y}_i) = 1 - \frac{v_{y_i} \cdot v_{\hat{y}_i}}{\|v_{y_i}\| \|v_{\hat{y}_i}\|}. \tag{6}$$

We use "Qwen3-Embedding-4B" as the embedding model (Yang et al., 2025a). (2) **Binary 0–1 loss** for verifiable tasks. It measures the accuracy drop when the expert answer is correct:

$$\ell(y_i, \hat{y}_i) = \ell(y_i, \hat{y}_i \mid y_i^{gold}) = \mathbb{1}\{\hat{y}_i \neq y_i^{gold}\}\mathbb{1}\{y_i = y_i^{gold}\}. \tag{7}$$

Here y_i^{gold} is the gold answer for the problem x_i. We discuss the choice of loss function in Appendix H.2.

Metrics To evaluate the effectiveness of the PAC reasoning in optimizing budget usage, we define two metrics: **Expert Call Percentage** (ECP) and **Saved Token Percentage** (STP). These metrics are formally defined as follows:

$$\text{ECP} := \frac{|\{i : U_i \geq \hat{u},\, i \in \mathcal{I}_{\text{test}}\}|}{N} \times 100\%,$$
$$\text{STP} := \frac{1}{N} \sum_{i \in \mathcal{I}_{\text{test}}} \left(1 - \frac{l_{\tilde{y}_i} + \mathbb{1}\{U_i \geq \hat{u}\} l_{y_i}}{l_{y_i}}\right) \times 100\%,$$

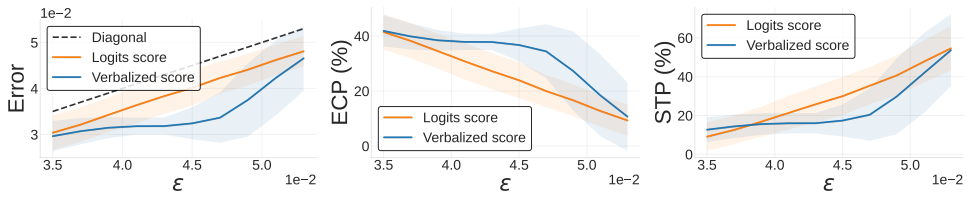
where $l_{\tilde{y}_i}$ and l_{y_i} represent the number of tokens in the candidate answer $\tilde{y}_i$ and the reference answer y_i, respectively. The ECP measures the proportion of test cases requiring expert intervention. At the same time, the STP quantifies the token efficiency by comparing the token counts of candidate and reference answers, accounting for cases where expert calls are triggered. We also report accuracy of our proposed method in Appendix I.6.

We repeat each experiment 100 times and report the mean and standard deviation of the budget savings. We fix $\alpha = 0.05$ throughout all experiments while varying ϵ, and set the sampling weight $\pi = \pi_i = 0.5$ for each $i \in \mathcal{I}_{cal}$ and the sample size $m = n \times \frac{1}{\pi}$.

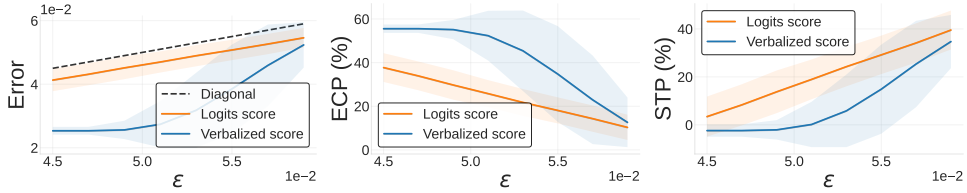
4.2 Results

PAC reasoning improves the efficiency and controls the performance loss. Figure 1 presents the reults of PAC reasoning under the semantic loss. The results show that PAC reasoning consistently ensures validity of performance loss across all the benchmarks: the empirical performance loss remains below the target risk level ϵ while achieving substantial budget savings. For instance, on Arena-Hard with $\epsilon = 0.08$ for the logits uncertainty score, the average empirical performance loss is approximately 0.06, the ECP is about 20%, and the STP is around 40%. Under the 0-1 binary loss (see Table 5 in Appendix I.7), PAC reasoning also maintains error rates within the target risk level and saves computational budget. For example, for the MATH-500 dataset, PAC reasoning saves ECP by 22.50% and STP by 23.13%. The results also align with the one using Llama-based model shown in Appendix I.1. In summary, PAC reasoning bounds the performance loss within the target risk level and significantly improves inference efficiency.

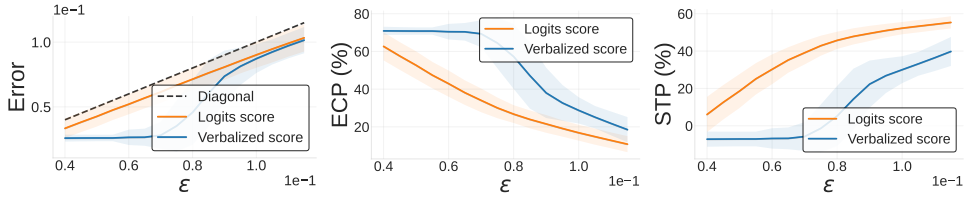
Logits-based uncertainty score is more stable. From the results in Figure 1 and Table 5, the logits-based uncertainty score consistently shows smoother and more stable behavior, with lower variance in both ECP and STP. In contrast, the verbalized uncertainty score exhibits larger fluctuations due to its sparse and clustered distribution. For instance, on ZebraLogic (see Table 5), the standard



(a) MATH-500



(b) ZebraLogic



(c) Arena-Hard

Figure 1: **Error control, ECP and STP of PAC reasoning** for semantic loss across three benchmarks at $\alpha = 0.05$. Uncertainty score includes the logits-based score and the verbalized score. All experiments are repeated 100 times, and the shaded areas represent standard deviations.

Table 1: **PAC reasoning enables safe and effective inference-cost reduction compared to heuristic baselines** under $\epsilon = 0.03$. Cells with green background indicates valid risk control. Experiments are conducted on MATH-500 using the binary loss. We use the logits-based uncertainty score for Naive control and PAC reasoning.

Metric	Naive control	PAC reasoning	Router	CoD	NoThinking
Binary Loss	0.0351 ± 0.0094	0.0206 ± 0.0126	0.0435 ± 0.0107	0.3548 ± 0.0604	0.4445 ± 0.0583
STP (%) ↑	61.53 ± 6.06	37.61 ± 23.19	74.73 ± 2.59	-1.33 ± 0.82	-4.51 ± 1.00
Pass@1	0.93 ± 0.25	$\mathbf{0.95 \pm 0.23}$	0.93 ± 0.26	0.61 ± 0.49	0.50 ± 0.50

deviations of ECP and STP under the verbalized score are 20.68% and 17.20%, considerably higher than the corresponding 7.47% and 9.90% values for the logits-based score. Similar patterns can also be observed in Figure 1, where the variance of the verbalized score is consistently higher than that of the logits-based score across different settings. This difference affects efficiency and variance, but it does not affect the validity of error control for PAC reasoning. Therefore, although the verbalized score occasionally achieves stronger risk control or higher savings, its calibration is less reliable, leading to less consistent performance.

PAC reasoning enables safe and effective inference-cost reduction compared to heuristic baselines. Table 1 summarizes the results under a target tolerance of $\epsilon = 0.03$. PAC reasoning is the **only** method that consistently satisfies the target error tolerance, which is a direct consequence of its explicit PAC-style guarantees on performance loss. In contrast, the naive control baseline and other heuristic methods fail to reliably meet the risk constraint, despite in some cases achieving higher token savings. In particular, learned routing achieves the highest saved token percentage but does not explicitly control downstream prediction risk, resulting in empirical error that exceeds the target tolerance. Similarly, CoD and NoThinking substantially reduce reasoning cost, but incur large performance degradation under the binary loss, reflecting the absence of mechanisms to control or calibrate the induced error. Overall, these results demonstrate that **PAC reasoning enables safe and effective inference-cost reduction** by explicitly controlling performance loss with high probability, clearly

distinguishing it from existing heuristic approaches. We provide further discussion on uncertainty scores and calibration set size in Appendix A.

5 Related work

Efficiency improvement for reasoning models Large Reasoning Models (LRMs) have recently become a research hotspot due to their outstanding performance in handling complex tasks (Yue et al., 2025). However, the problem of overthinking has emerged (Sui et al., 2025; Chen et al., 2025), where LRMs tend to engage in unnecessarily long reasoning chains and redundant computational steps. This increases latency and cost, and may even cause error accumulation through extended reasoning paths. For example, in mathematical problems, LRMs may explore irrelevant solution branches or perform excessive intermediate calculations that do not contribute to the final answer. To alleviate this issue, recent studies propose efficient reasoning strategies such as early exit of thinking (Yang et al., 2025b; Jiang et al., 2025) and adaptive switching between "fast" and "slow" thinking modes to reduce reasoning tokens and avoid redundant steps (Cheng et al., 2025; Chung et al., 2025; Fang et al., 2025; Li et al., 2025b; Liang et al., 2025; Ma et al., 2025; Paliotta et al., 2025; Pan et al., 2024a;b; Xiao & Gan, 2025; Yao et al., 2024; Yong et al., 2025; Zhang et al., 2025b;a). Despite their empirical effectiveness, these techniques lack theoretical guarantees on the performance loss when using the non-thinking mode. We fill this gap by introducing a PAC-based reasoning that provides statistically guaranteed performance loss for efficient reasoning.

PAC-style error rate control Modern PAC-style error rate control (Valiant, 1984) is built on distribution-free risk control. Conformal prediction provides finite-sample coverage guarantees under exchangeability (Angelopoulos et al., 2025b). This idea extends to controlling expected loss on prediction sets, enabling risk control for tasks such as multi-label classification and image segmentation (Bates et al., 2021). Risk control has also been reframed as multiple hypothesis testing, which enables simultaneous control of multiple risks, including false discovery rate (Angelopoulos et al., 2025a). These methods have been applied to various domains, including conformal language modeling for text generation with guaranteed quality (Quach et al., 2024), and automatically adaptive conformal risk control that adjusts to the difficulty of test samples (Blot et al., 2025). Localized adaptive risk control enables online calibration with conditional guarantees (Zecchin & Simeone, 2024). Despite this progress, PAC-style risk control has not addressed efficiency control in reasoning models: the routing problem of choosing between a thinking model and a non-thinking model under a user-specified loss tolerance remains open. We fill this gap by bringing PAC-style risk control to LRM routing, learning a threshold that reduces computation while providing a distribution-free guarantee on performance loss.

6 Conclusion

We propose PAC reasoning, which is designed to provide rigorous, theoretically grounded, and practical efficiency improvement for reasoning models while maintaining probabilistic correctness guarantees. Our approach constructs a composite model that selectively uses either an expert model or a candidate model based on a constructed confidence bound. The PAC reasoning contributes by delivering statistical assurances for model performance and demonstrating that it reliably achieves the specified error rate with probability. We validate our method through extensive experiments on real-world reasoning tasks, showing it can significantly reduce computational costs while maintaining user-specified error tolerances with confidence. Future research will focus on developing more advanced uncertainty estimation techniques, exploring tighter theoretical bounds, and broadening the method's applicability to other large language model efficiency-improvement strategies.

Future work. Our current framework uses a single threshold for all inputs, which may be suboptimal when tasks require different levels of model capability. A promising direction is *multi-level PAC reasoning*, which goes beyond binary routing to multiple model tiers by learning thresholds $u_1 < u_2 < \ldots < u_k$ that route inputs based on uncertainty levels, with PAC guarantees that bound cumulative performance loss across tiers. We discuss the multi-level reasoning in Appendix G.

REFERENCES

Anastasios N. Angelopoulos, Stephen Bates, Emmanuel J. Candès, Michael I. Jordan, and Lihua Lei. Learn then test: Calibrating predictive algorithms to achieve risk control. *The Annals of Applied Statistics*, 19(2):1641–1662, 2025a. ISSN 1932-6157, 1941-7330.

Anastasios N. Angelopoulos, Stephen Bates, Adam Fisch, Lihua Lei, and Tal Schuster. Conformal risk control. *Preprint at arXiv:2208.02814*, 2025b.

Stephen Bates, Anastasios Angelopoulos, Lihua Lei, Jitendra Malik, and Michael Jordan. Distribution-free, risk-controlling prediction sets. *Journal of the ACM*, 68(6):1–34, 2021. ISSN 0004-5411, 1557-735X.

Vidmantas Bentkus. On hoeffding's inequalities. *Annals of Probability*, 32(2):0–26, 2004. ISSN 0091-1798.

Vincent Blot, Anastasios Nikolas Angelopoulos, Michael Jordan, and Nicolas J.-B. Brunel. Automatically adaptive conformal risk control. In *Proceedings of The 28th International Conference on Artificial Intelligence and Statistics*, pp. 19–27. PMLR, 2025.

Chen, Tworek, Jun, Yuan, Pinto, Kaplan, and et al. Evaluating large language models trained on code. *Preprint at arXiv:2107.03374*, 2021.

Chen, Qin, Liu, Peng, Guan, Wang, and et al. Towards reasoning era: A survey of long chain-of-thought for reasoning large language models. *Preprint at arXiv:2503.09567*, 2025.

Xiaoxue Cheng, Junyi Li, Wayne Xin Zhao, and Ji-Rong Wen. Think more, hallucinate less: Mitigating hallucinations via dual process of fast and slow thinking. *Preprint at arXiv:2501.01306*, 2025.

Stephen Chung, Wenyu Du, and Jie Fu. Thinker: Learning to think fast and slow. *Preprint at arXiv:2505.21097*, 2025.

Clark, Cowhey, Etzioni, Khot, Sabharwal, Schoenick, and et al. Think you have solved question answering? Try ARC, the AI2 reasoning challenge. *Preprint at arXiv:1803.05457*, 2018.

Cobbe, Kosaraju, Bavarian, Chen, Jun, Kaiser, and et al. Training verifiers to solve math word problems. *Preprint at arXiv:2110.14168*, 2021.

DeepSeek-AI, Guo, Yang, Zhang, Song, Zhang, and et al. DeepSeek-R1: Incentivizing reasoning capability in LLMs via reinforcement learning. *Preprint at arXiv:2501.12948*, 2025.

Gongfan Fang, Xinyin Ma, and Xinchao Wang. Thinkless: LLM learns when to think. *Preprint at arXiv:2505.13379*, 2025.

Tao Feng, Haozhen Zhang, Zijie Lei, Haodong Yue, Chongshan Lin, Ge Liu, and et al. LLMRouter: An Open-Source Library for LLM Routing, 2025.

Chuan Guo, Geoff Pleiss, Yu Sun, and Kilian Q. Weinberger. On calibration of modern neural networks. In *Proceedings of the 34th International Conference on Machine Learning*, pp. 1321–1330. PMLR, 2017.

Hao, Gu, Ma, Hong, Wang, Wang, and et al. Reasoning with language model is planning with world model. *Preprint at arXiv:2305.14992*, 2023.

Botao Hao, Yasin Abbasi Yadkori, Zheng Wen, and Guang Cheng. Bootstrapping upper confidence bound. In *Advances in Neural Information Processing Systems*, volume 32. Curran Associates, Inc., 2019.

Hendrycks, Burns, Basart, Zou, Mazeika, Song, and et al. Measuring massive multitask language understanding. *Preprint at arXiv:2009.03300*, 2021a.

Hendrycks, Burns, Kadavath, Arora, Basart, Tang, and et al. Measuring mathematical problem solving with the MATH dataset. *Preprint at arXiv:2103.03874*, 2021b.

Wassily Hoeffding. Probability inequalities for sums of bounded random variables. *Journal of the American Statistical Association*, 58(301):13–30, 1963. ISSN 0162-1459.

Wassily Hoeffding. Probability inequalities for sums of bounded random variables. In N. I. Fisher and P. K. Sen (eds.), *The Collected Works of Wassily Hoeffding*, pp. 409–426. Springer, New York, NY, 1994. ISBN 978-1-4612-0865-5. doi: 10.1007/978-1-4612-0865-5_26.

Huang, Song, Wang, Zhao, Chen, Juefei-Xu, and et al. Look before you leap: An exploratory study of uncertainty measurement for large language models. *IEEE Transactions on Software Engineering*, 51(2):413–429, 2025. ISSN 0098-5589, 1939-3520, 2326-3881.

Guochao Jiang, Guofeng Quan, Zepeng Ding, Ziqin Luo, Dixuan Wang, and Zheng Hu. FlashThink: An early exit method for efficient reasoning. *Preprint at arXiv:2505.13949*, 2025.

Mandar Joshi, Eunsol Choi, Daniel Weld, and Luke Zettlemoyer. TriviaQA: A large scale distantly supervised challenge dataset for reading comprehension. In Regina Barzilay and Min-Yen Kan (eds.), *Proceedings of the 55th Annual Meeting of the Association for Computational Linguistics (Volume 1: Long Papers)*, pp. 1601–1611, Vancouver, Canada, 2017. Association for Computational Linguistics. doi: 10.18653/v1/P17-1147.

Kwiatkowski, Palomaki, Redfield, Collins, Parikh, Alberti, and et al. Natural questions: A benchmark for question answering research. *Transactions of the Association for Computational Linguistics*, 7: 452–466, 2019.

Kwon, Li, Zhuang, Sheng, Zheng, Yu, and et al. Efficient memory management for large language model serving with PagedAttention. In *Proceedings of the 29th Symposium on Operating Systems Principles*, SOSP '23, pp. 611–626, New York, NY, USA, 2023. Association for Computing Machinery. ISBN 979-8-4007-0229-7. doi: 10.1145/3600006.3613165.

Erik Learned-Miller and Philip S. Thomas. A new confidence interval for the mean of a bounded random variable. *Preprint at arXiv:1905.06208*, 2020.

Li, Chiang, Frick, Dunlap, Wu, Zhu, and et al. From crowdsourced data to high-quality benchmarks: Arena-hard and benchbuilder pipeline. In *Forty-Second International Conference on Machine Learning, ICML 2025*, Vancouver, BC, Canada, 2025a.

Li, Wei, Huang, Yan, Chen, Kwok, and et al. DynamicMind: A tri-mode thinking system for large language models. *Preprint at arXiv:2506.05936*, 2025b.

Guosheng Liang, Longguang Zhong, Ziyi Yang, and Xiaojun Quan. ThinkSwitcher: When to think hard, when to think fast. *Preprint at arXiv:2505.14183*, 2025.

Lightman, Kosaraju, Burda, Edwards, Baker, Lee, and et al. Let's verify step by step. In *The Twelfth International Conference on Learning Representations*, 2023.

Lin, Bras, Richardson, Sabharwal, Poovendran, Clark, and et al. ZebraLogic: On the scaling limits of LLMs for logical reasoning. In *Forty-Second International Conference on Machine Learning*, 2025.

Wenjie Ma, Jingxuan He, Charlie Snell, Tyler Griggs, Sewon Min, and Matei Zaharia. Reasoning models can be effective without thinking. *Preprint at arXiv:2504.09858*, 2025.

Ong, Almahairi, Wu, Chiang, Wu, Gonzalez, and et al. RouteLLM: Learning to route LLMs with preference data. *Preprint at arXiv:2406.18665*, 2025.

Paliotta, Wang, Pagliardini, Li, Bick, Kolter, and et al. Thinking slow, fast: Scaling inference compute with distilled reasoners. *Preprint at arXiv:2502.20339*, 2025.

Pan, Zhang, Zhang, Liu, Wang, Li, and et al. DynaThink: Fast or slow? A dynamic decision-making framework for large language models. In Yaser Al-Onaizan, Mohit Bansal, and Yun-Nung Chen (eds.), *Proceedings of the 2024 Conference on Empirical Methods in Natural Language Processing*, pp. 14686–14695, Miami, Florida, USA, 2024a. Association for Computational Linguistics. doi: 10.18653/v1/2024.emnlp-main.814.

Pan, Ji, Ding, Li, Chen, Wang, and et al. A survey of slow thinking-based reasoning LLMs using reinforced learning and inference-time scaling law. *Preprint at arXiv:2505.02665*, 2025.

Jiabao Pan, Yan Zhang, Chen Zhang, Zuozhu Liu, Hongwei Wang, and Haizhou Li. DynaThink: Fast or slow? A dynamic decision-making framework for large language models. *Preprint at arXiv:2407.01009*, 2024b.

Quach, Fisch, Schuster, Yala, Sohn, Jaakkola, and et al. Conformal language modeling. In *The Twelfth International Conference on Learning Representations*, Vienna, Austria, 2024.

Aaditya Ramdas, Johannes Ruf, Martin Larsson, and Wouter Koolen. Admissible anytime-valid sequential inference must rely on nonnegative martingales. *Preprint at arXiv:2009.03167*, 2022.

Rein, Hou, Stickland, Petty, Pang, Dirani, and et al. GPQA: A graduate-level google-proof Q&A benchmark. In *First Conference on Language Modeling*, 2024.

Roller, Dinan, Goyal, Ju, Williamson, Liu, and et al. Recipes for building an open-domain chatbot. In *Proceedings of the 16th Conference of the European Chapter of the Association for Computational Linguistics: Main Volume*, pp. 300–325, Online, 2021. Association for Computational Linguistics. doi: 10.18653/v1/2021.eacl-main.24.

Sui, Chuang, Wang, Zhang, Zhang, Yuan, and et al. Stop overthinking: A survey on efficient reasoning for large language models. *Transactions on Machine Learning Research*, 2025. ISSN 2835-8856.

Alon Talmor, Jonathan Herzig, Nicholas Lourie, and Jonathan Berant. CommonsenseQA: A question answering challenge targeting commonsense knowledge. In *Proceedings of the 2019 Conference of the North American Chapter of the Association for Computational Linguistics: Human Language Technologies, Volume 1 (Long and Short Papers)*, pp. 4149–4158, Minneapolis, Minnesota, 2019. Association for Computational Linguistics. doi: 10.18653/v1/N19-1421.

Tian, Mitchell, Zhou, Sharma, Rafailov, Yao, and et al. Just ask for calibration: Strategies for eliciting calibrated confidence scores from language models fine-tuned with human feedback. In Houda Bouamor, Juan Pino, and Kalika Bali (eds.), *Proceedings of the 2023 Conference on Empirical Methods in Natural Language Processing*, pp. 5433–5442, Singapore, 2023. Association for Computational Linguistics. doi: 10.18653/v1/2023.emnlp-main.330.

L. G. Valiant. A theory of the learnable. In *Proceedings of the Sixteenth Annual ACM Symposium on Theory of Computing*, STOC '84, pp. 436–445, New York, NY, USA, 1984. Association for Computing Machinery. ISBN 978-0-89791-133-7. doi: 10.1145/800057.808710.

Ian Waudby-Smith and Aaditya Ramdas. Confidence sequences for sampling without replacement. *Preprint at arXiv:2006.04347*, 2021.

Ian Waudby-Smith and Aaditya Ramdas. Estimating means of bounded random variables by betting. *Journal of the Royal Statistical Society Series B (Statistical Methodology)*, 86(1):1–27, 2024. ISSN 1369-7412.

Wenyi Xiao and Leilei Gan. Fast-slow thinking GRPO for large vision-language model reasoning. *Preprint at arXiv:2504.18458*, 2025.

Xiong, Hu, Lu, Li, Fu, He, and et al. Can LLMs express their uncertainty? An empirical evaluation of confidence elicitation in LLMs. In *The Twelfth International Conference on Learning Representations*, 2023.

Silei Xu, Wenhao Xie, Lingxiao Zhao, and Pengcheng He. Chain of draft: Thinking faster by writing less. *Preprint at arXiv:2502.18600*, 2025.

Yang, Li, Yang, Zhang, Hui, Zheng, and et al. Qwen3 technical report. *Preprint at arXiv:2505.09388*, 2025a.

Yang, Si, Duan, Zhu, Zhu, Li, and et al. Dynamic early exit in reasoning models. *Preprint at arXiv:2504.15895*, 2025b.

Daniel Yang, Yao-Hung Hubert Tsai, and Makoto Yamada. On verbalized confidence scores for LLMs. *Preprint at arXiv:2412.14737*, 2024.

Wenlin Yao, Haitao Mi, and Dong Yu. HDFlow: Enhancing LLM complex problem-solving with hybrid thinking and dynamic workflows. *Preprint at arXiv:2409.17433*, 2024.

Xixian Yong, Xiao Zhou, Yingying Zhang, Jinlin Li, Yefeng Zheng, and Xian Wu. Think or not? Exploring thinking efficiency in large reasoning models via an information-theoretic lens. *Preprint at arXiv:2505.18237*, 2025.

Yue, Du, Wang, Gao, Yao, Wang, and et al. Don't overthink it: A survey of efficient R1-style large reasoning models. *Preprint at arXiv:2508.02120*, 2025.

Matteo Zecchin and Osvaldo Simeone. Localized adaptive risk control. In *Advances Neural Information Processing Systems*, volume 37, 2024.

Zhang, Roller, Goyal, Artetxe, Chen, Chen, and et al. OPT: Open pre-trained transformer language models. *Preprint at arXiv:2205.01068*, 2022.

Zhang, Li, Chen, Zhang, Ye, Bai, and et al. Beyond GPT-5: Making LLMs cheaper and better via performance-efficiency optimized routing. *Preprint at arXiv:2508.12631*, 2025a.

Zhang, Li, Wang, Chen, Zhang, Ye, and et al. The avengers: A simple recipe for uniting smaller language models to challenge proprietary giants. *Preprint at arXiv:2505.19797*, 2025b.

Zhang, Zheng, Wu, Zhang, Lin, Yu, and et al. The lessons of developing process reward models in mathematical reasoning. *Preprint at arXiv:2501.07301*, 2025c.

Zheng, Yin, Xie, Sun, Huang, Yu, and et al. SGLang: Efficient execution of structured language model programs. In *The Thirty-eighth Annual Conference on Neural Information Processing Systems*, 2024.

Zhou, Tan, Li, Yao, Guo, Li, and et al. A theoretical study on bridging internal probability and self-consistency for LLM reasoning. *Preprint at arXiv:2510.15444*, 2025.

A Discussion

In this section, we study how robust PAC reasoning is to practical design. We first consider the reliability of the uncertainty score using expected calibration error (ECE). Then we study the ablation on varying the uncertainty score and the calibration set size. Across these settings, PAC reasoning keeps valid error control and shows stable efficiency.

Quality of uncertainty score To assess the reliability of uncertainty scores, we compute the ECE, which measures the difference between predicted confidence and actual accuracy. Our analysis in Appendix I.3 reveals that logits-based uncertainty scores exhibit lower ECE values compared to verbalized scores on most benchmarks. For example, on MATH-500, the logits-based score achieves an ECE of 0.0450, while the verbalized score has an ECE of 0.0634. Despite the differences in calibration quality, both approaches successfully maintain the PAC guarantee, demonstrating the robustness of our framework to imperfect uncertainty quantification. It means that perfect calibration is not necessary for PAC reasoning.

Calibration set size The size of the calibration set presents a practical trade-off between calibration cost and guarantee of tightness. Our comprehensive ablation study in Appendix I.4 examines calibration ratios ranging from 10% to 90% of the total dataset. The results show that error control is remarkably stable across different calibration sizes, with average loss remaining within the target tolerance even for small calibration sets. For instance, on MATH-500 with logits-based score, the average loss varies only from 0.0331 to 0.0344 as the calibration ratio increases from 10% to 90%. The STP also remains relatively stable, ranging from 14.39% to 18.52%. This demonstrates that PAC reasoning can be deployed with modest calibration costs while still providing meaningful efficiency gains and statistical guarantees.

Alternative uncertainty scores Our framework is flexible and can accommodate any scoring function that correlates with the disagreement rate. An alternative is reward model-based scoring, where a trained reward model evaluates the quality of the non-thinking model's output. We conduct experiments in Appendix I.5 using reward scores on MATH-500 and ZebraLogic, demonstrating that PAC reasoning maintains valid error control even with this alternative scoring method. For example, on MATH-500 with $\epsilon = 0.045$, reward-based PAC reasoning achieves an error of 0.0362 with 22.43% token savings. This flexibility is particularly valuable in scenarios where uncertainty scores are unreliable or unavailable.

B Finite-Sample Upper Confidence Bounds

This section presents an alternative algorithm for computing confidence bounds that provides strict finite-sample guarantees under bounded loss assumptions. This approach leverages concentration inequalities such as the Hoeffding inequality or betting-based confidence intervals (Bentkus, 2004; Hao et al., 2019; Hoeffding, 1994; Learned-Miller & Thomas, 2020; Ramdas et al., 2022; Waudby-Smith & Ramdas, 2021; 2024) to achieve tighter bounds compared to the asymptotic normal approximation used in Algorithm 1.

C Validity of CLT-based upper confidence bound

We show that a CLT-based upper confidence bound computed on the calibration set satisfies Assumption 3.1 asymptotically.

Proposition C.1 (Asymptotic validity of UCB baed on CLT). *Fix a threshold u and let $Z_j(u)$ be the i.i.d. random variables defined in Algorithm 1 for $j = 1, \ldots, m$, with mean $L(u)$ and variance $\sigma_Z^2(u) > 0$. Let $\widehat{\mu}_Z(u) = m^{-1} \sum_{j=1}^m Z_j(u)$ and $\widehat{\sigma}_Z^2(u) = (m-1)^{-1} \sum_{j=1}^m \big(Z_j(u) - \widehat{\mu}_Z(u)\big)^2$. Define the UCB based on CLT*

$$\widehat{L}_u^{\mathrm{CLT}}(\alpha) \;:=\; \widehat{\mu}_Z(u) \;+\; z_{1-\alpha} \sqrt{\widehat{\sigma}_Z^2(u)/m},$$

where $z_{1-\alpha}$ is the $(1-\alpha)$-quantile of the standard normal distribution. Then

$$\liminf_{m \to \infty} \mathbb{P}\big(L(u) \le \widehat{L}_u^{\mathrm{CLT}}(\alpha)\big) \;\ge\; 1 - \alpha.$$

Algorithm 3 Compute Confidence Bound $\hat{L}_u(\alpha)$ Based on Hoeffding Inequality

Input: Calibration set $\{(x_i, y_i)\}_{i=1}^n$, model with thinking f, model without thinking $\tilde{f}$, uncertainty scores $\{U_i\}_{i=1}^n$, sampling weights $\{\pi_i\}_{i=1}^n$, sampling size m, confidence level α, loss upper bound $B > 0$

Output: The finite-sample upper confidence bound $\hat{L}_u(\alpha)$.

Initialize an empty list of samples $\mathcal{Z} = []$.
Let $\tilde{y}_i = \tilde{f}(x_i)$ for all $i = 1, \ldots, n$.
for $j = 1, \ldots, m$ **do**
  Sample an index $i_j \sim \text{Unif}(\{1, \ldots, n\})$.
  Sample a Bernoulli random variable $\xi_{i_j} \sim \text{Bern}(\pi_{i_j})$.
  if $\xi_{i_j} = 1$ **then**
    Query the true label y_{i_j} and compute $Z_j = \min\left(\frac{\ell(y_{i_j}, \tilde{y}_{i_j})}{\pi_{i_j}}, \frac{B}{\pi_{i_j}}\right)$.
  else
    $Z_j = 0$.
  end if
  Append Z_j to $\mathcal{Z}$.
end for
For a threshold u, define the variables $Z_j(u) = Z_j \cdot \mathbf{1}\{U_{i_j} \le u\}$ for $j = 1, \ldots, m$.
$\hat{\mu}_Z(u) \leftarrow \frac{1}{m}\sum_{j=1}^m Z_j(u)$
$R \leftarrow \frac{B}{\min_i \pi_i}$
$\delta_{\text{HB}}(\alpha) \leftarrow \sqrt{\frac{R^2 \log(2/\alpha)}{2m}}$
Return $\hat{L}_u(\alpha) \leftarrow \hat{\mu}_Z(u) + \delta_{\text{HB}}(\alpha)$.

Proof. By the classical Lindeberg–Feller central limit theorem,

$$\frac{\sqrt{m}\left(\widehat{\mu}_Z(u) - L(u)\right)}{\sigma_Z(u)} \to \mathcal{N}(0,1)$$

in distribution as $m \to \infty$ because the variables are i.i.d. Since $\widehat{\sigma}_Z(u) \xrightarrow{p} \sigma_Z^2(u)$ by the weak law of large numbers, Slutsky's theorem yields

$$\frac{\sqrt{m}\left(\widehat{\mu}_Z(u) - L(u)\right)}{\sqrt{\widehat{\sigma}_Z(u)}} \to \mathcal{N}(0,1).$$

Therefore

$$\mathbb{P}\left(\frac{L(u) - \widehat{\mu}_Z(u)}{\sqrt{\widehat{V}_Z(u)/m}} \le z_{1-\alpha}\right) \to 1 - \alpha.$$

Equivalently, define $d_m := (L(u) - \widehat{\mu}_Z(u))/\sqrt{\widehat{V}_Z(u)/m}$ and observe that

$$\{d_m \le z_{1-\alpha}\} = \left\{L(u) \le \widehat{\mu}_Z(u) + z_{1-\alpha}\sqrt{\widehat{V}_Z(u)/m}\right\}.$$

Hence,

$$\liminf_{m\to\infty} \mathbb{P}\big(L(u) \le \widehat{L}_u^{\text{CLT}}(\alpha)\big) \;\ge\; 1 - \alpha.$$

□

D PROOF OF THEOREM 3.2

Proof. Let

$$u^\star := \inf\{u \in \Lambda : R(u) > \epsilon\}.$$

As $R(u)$ is non-decreasing, it holds that

$$R(u) \le \epsilon \text{ for all } u \le u^\star,$$

and

$$R(u^\star) > \epsilon.$$

If $\hat{u} > u^\star$, then

$$\widehat{L}_{u^\star}(\alpha) \le \epsilon < R(u^\star),$$

which contradicts Assumption 3.1 except with probability at most α.

Therefore,

$$\mathbb{P}(\hat{u} \le u^\star) \ge 1 - \alpha,$$

and because $R(u)$ is non-decreasing,

$$R(\hat{u}) \le \epsilon \text{ on this event.}$$

□

E Proof of Theorem 3.3

E.1 Notation recalling and lemma

We present PAC guarantees for the empirical test risk under precise Hoeffding conditions, making explicit the roles of calibration and test randomness. For any threshold u, the deployment rule T_u predicts with the expert when $U(x) \ge u$ and otherwise uses the fast model. The population risk is

$$R(u) \;=\; \mathbb{E}_{(x,y)\sim P}\big[\ell\big(y,\, T_u(x)\big)\big].$$

Given an independent test set $\mathcal{D}_{\text{test}}$ drawn i.i.d. from P, the empirical test risk is

$$\widehat{R}(u) \;=\; \frac{1}{N}\sum_{j=n+1}^{n+N} \ell\big(y_j,\, T_u(x_j)\big).$$

Our guarantee for the empirical test risk relies on Assumption 3.1 from the main text, in addition to the following lemma.

Lemma E.1 (Conditional Hoeffding bound)**.** *Let $\hat{u}$ be a random variable determined by the calibration set $\mathcal{D}_{cal}$. Assume that, conditioned on $\hat{u}$, the test losses $Z_j(\hat{u}) := \ell\big(y_j,\, T_{\hat{u}}(x_j)\big)$ for $j \in \mathcal{I}_{test}$ are i.i.d. and bounded in $[a, b]$. Then for any $t > 0$,*

$$\mathbb{P}\big(\widehat{R}(\hat{u}) - R(\hat{u}) > t \,\big|\, \hat{u}\big) \;\le\; \exp\Big(-\tfrac{2Nt^2}{(b-a)^2}\Big).$$

Proof. Let $Z_j(\hat{u}) = \ell(y_j, T_{\hat{u}}(x_j))$ for $j \in \mathcal{I}_{test}$. The model $\hat{u}$ is fixed when we condition on it. Since the test set $\mathcal{D}_{\text{test}}$ consists of i.i.d. samples and is independent of $\hat{u}$, the random variables $Z_1(\hat{u}), \ldots, Z_N(\hat{u})$ are conditionally independent and identically distributed.

By the boundness of the loss function, each $Z_j(\hat{u})$ is bounded in $[a, b]$. The conditional expectation of each $Z_j(\hat{u})$ is $\mathbb{E}[Z_j(\hat{u}) \,|\, \hat{u}] = \mathbb{E}[\ell(y_j, T_{\hat{u}}(x_j)) \,|\, \hat{u}]$. Since (x_j, y_j) is independent of $\hat{u}$, this is equal to the unconditional expectation over the data distribution, $\mathbb{E}_{(x,y)\sim P}[\ell(y, T_{\hat{u}}(x))]$, which is the definition of the true risk $R(\hat{u})$. The empirical risk is the sample mean:

$$\widehat{R}(\hat{u}) = \frac{1}{N}\sum_{j=1}^{N} Z_j(\hat{u}).$$

Its conditional expectation is

$$\mathbb{E}[\widehat{R}(\hat{u}) \,|\, \hat{u}] = \frac{1}{N}\sum_{j=1}^{N} \mathbb{E}[Z_j(\hat{u}) \,|\, \hat{u}] = R(\hat{u}).$$

We can now apply Hoeffding's inequality (Hoeffding, 1963) to the conditional i.i.d. bounded variables $Z_j(\hat{u})$. For any $t > 0$, the one-sided version states that

$$\mathbb{P}\left(\frac{1}{N}\sum_{j=n+1}^{n+N} Z_j(\hat{u}) - \mathbb{E}\left[\frac{1}{N}\sum_{j=n+1}^{n+N} Z_j(\hat{u}) \,\middle|\, \hat{u}\right] > t \,\middle|\, \hat{u}\right) \le \exp\left(-\frac{2Nt^2}{(b-a)^2}\right).$$

Substituting the empirical risk and true risk, we get

$$\mathbb{P}\left(\widehat{R}(\hat{u}) - R(\hat{u}) > t \,\middle|\, \hat{u}\right) \leq \exp\left(-\frac{2Nt^2}{(b-a)^2}\right).$$

This completes the proof. □

E.2 Proof details for Theorems 3.3

Proof. The independence of $\mathcal{D}_{\text{test}}$ and $\mathcal{D}_{cal}$ ensures Lemma E.1 hold. Use the inclusion

$$\{\widehat{R}(\hat{u}) > \epsilon + t\} \subseteq \{R(\hat{u}) > \epsilon\} \cup \{\widehat{R}(\hat{u}) - R(\hat{u}) > t\}$$

and take probabilities. Combine the result of Theorem 3.2 (i.e., $\mathbb{P}(R(\hat{u}) > \epsilon) \leq \alpha$) with the law of total probability and Lemma E.1 to conclude the claim:

$$\begin{aligned}\mathbb{P}(\widehat{R}(\hat{u}) > \epsilon + t) &\leq \mathbb{P}(R(\hat{u}) > \epsilon) + \mathbb{P}(\widehat{R}(\hat{u}) - R(\hat{u}) > t) \\ &= \mathbb{P}(R(\hat{u}) > \epsilon) + \mathbb{E}\big[\mathbb{P}\big(\widehat{R}(\hat{u}) - R(\hat{u}) > t \,\big|\, \hat{u}\big)\big] \\ &\leq \alpha + \exp\Big(-\tfrac{2Nt^2}{(b-a)^2}\Big).\end{aligned}$$

This is equivalent to the stated guarantee. □

F Transductive PAC Reasoning

In this section, we introduce a transductive version of PAC reasoning. In this setting, the calibration set and the test set are identical. We consider a fixed dataset $\mathcal{D} = \mathcal{D}_{test} = \mathcal{D}_{cal} = \{x_1, \ldots, x_n\}$, and the randomness only comes from the algorithm itself (e.g., which data points are selected to query the expert, the sampling design, and the internal randomization of the mean upper bound estimator). The goal is to provide a guarantee of empirical performance over this fixed dataset. Specifically, the algorithm ensures that the empirical average performance loss is controlled below a user-specified tolerance level ϵ with a confidence of at least $1 - \alpha$.

Let's update the setup for the transductive setting. For a given threshold u, the empirical risk on the dataset $\mathcal{D}$ is defined as

$$L(u) := \frac{1}{n}\sum_{i=1}^{n} \ell(y_i, \hat{y}_i)\, \mathbf{1}\{U_i \leq u\}.$$

This represents the average loss for the data points where the model is used (i.e., uncertainty is below the threshold u), and $L(u)$ is non-decreasing in u by construction. The validity of our transductive PAC reasoning algorithm relies on the following assumptions. For any given threshold u and significance level α, there exists an upper confidence bound (UCB) $\widehat{L}_u(\alpha)$, computable from samples drawn by the algorithm, that satisfies

$$\mathbb{P}\big(L(u) \leq \widehat{L}_u(\alpha)\big) \geq 1 - \alpha.$$

In practice, we instantiate $\widehat{L}_u(\cdot)$ using our UCB procedures. Concretely, one may compute $\widehat{L}_u(\alpha')$ via Algorithm 1 (CLT-based) or Algorithm 3 (Hoeffding/Bentkus/betting-based), depending on sample size and desired conservatives. We summarize our transductive style method in Algorithm 4.

Theorem F.1 (Transductive PAC guarantee)**.** *Suppose* $\mathbb{P}\big(L(u) \leq \widehat{L}_u(\alpha)\big) \geq 1 - \alpha$. *Then the procedure in Algorithm 4, which selects* $\hat{u} = \max\{u : \widehat{L}_u(\alpha) \leq \varepsilon\}$, *achieves*

$$\mathbb{P}\big(L(\hat{u}) \leq \varepsilon\big) \geq 1 - \alpha.$$

Proof. Let E be the event that $L_u \leq \widehat{L}_u(\alpha)$ holds simultaneously for all $u \in \Lambda$. Then $\mathbb{P}(E) \geq 1 - \alpha$. On E, for any u with $\widehat{L}_u(\alpha) \leq \varepsilon$ we have $L(u) \leq \varepsilon$. By the selection rule, either the set is non-empty and $\hat{u}$ is the maximal such u, which implies $L_{\hat{u}} \leq \varepsilon$ by monotonicity, and we take $\hat{u} = -\infty$, in which case $L_{\hat{u}} = 0 \leq \varepsilon$ since no model predictions are used. Therefore $\mathbf{1}\{L(\hat{u}) \leq \varepsilon\} = 1$ on E, and hence $\mathbb{P}(L_{\hat{u}} \leq \varepsilon) \geq \mathbb{P}(E) \geq 1 - \alpha$. □

Algorithm 4 Transductive PAC-Labeling

Inputs: Test ataset $\mathcal{D} = \{x_1, \ldots, x_n\}$, uncertainty scores $U_i, i = 1, ...n$, model predictions $\tilde{y}_i, \tilde{y}_n$, tolerance ϵ, significance level α, number of trials m, sampling probabilities $\pi_i, i = 1, ...n$, UCB function $\hat{L}_u(\alpha)$.
Output: Labeled dataset $\{(X_i, \widetilde{Y}_i)\}_{i=1}^n$ and threshold $\hat{u}$.
Sampling phase:
for $j = 1, \ldots, m$ **do**
  Draw an index i_j according to the sampling design (e.g., uniform or importance-based).
  With probability π_{i_j}, query the expert for y_{i_j}. Let $\xi_j \sim$ Bernoulli(π_{i_j}).
  if $\xi_j = 1$ **then**
    Observe the true label y_{i_j} and compute the loss $\ell(y_{i_j}, \tilde{y}_{i_j})$.
  else
    Mark as not-observed.
  end if
end for
For each candidate threshold, compute UCB $\hat{L}_u(\alpha)$
Choose the estimated threshold $\hat{u} := \max\{\widehat{L}_u \leq \varepsilon\}$.
for $i = 1, \ldots, n$ **do**
  if $U_i \geq \hat{u}$ **then**
    Ensure expert label y_i is obtained (query now if not already queried).
    Set $\widetilde{y}_i := y_i$.
  else
    Set $\widetilde{y}_i := \hat{y}_i$.
  end if
end for

Remark F.2 (Comparison with the inductive setting). In the inductive setting, the calibration and test sets are drawn independently of the same distribution. The goal is to guarantee performance on future, unseen data points from that distribution. The guarantee is of the form $\mathbb{P}(\mathbb{E}L(\hat{u}) \leq \varepsilon) \geq 1-\alpha$, where the probability is over the random draws of both datasets. In contrast, our transductive approach provides a guarantee for a specific, fixed dataset, which can be more suitable in applications where the set of items to be labeled is known in advance or the calibration and test datasets are not exchangeable.

G MULTI LEVEL PAC REASONING

We extend PAC reasoning to a three-tier composite LRM while preserving the statistical guarantees. We introduce two non-thinking LRMs, denoted by $\tilde{f}_1$ and $\tilde{f}_2$, whose computational costs and accuracies lie below the model with thinking f. For each input prompt x, the expert output is $y = f(x)$, and the non-thinking outputs are $\tilde{y}_1 = \tilde{f}_1(x)$ and $\tilde{y}_2 = \tilde{f}_2(x)$. We assume the existence of a unified uncertainty score $U(x) \in [0, 1]$ produced inexpensively, for example by $\tilde{f}_1$, as in Section 3. We define the deployment mapping with two thresholds $u_1, u_2 \in [0, 1]$ and $u_1 \leq u_2$:

$$T_{u_1,u_2}(x) = f(x)\,\mathbf{1}\{U(x) \geq u_2\} + \tilde{f}_2(x)\,\mathbf{1}\{u_1 < U(x) \leq u_2\} + \tilde{f}_1(x)\,\mathbf{1}\{U(x) \leq u_1\}.$$

The composite LRM is $\hat{f} = T_{\hat{u}_1,\hat{u}_2}$ for calibrated thresholds $(\hat{u}_1, \hat{u}_2)$. We re-parameterize the population risk and empirical risk of the composite LRM by the threshold pair:

$$R(u_1, u_2) = \mathbb{E}[\ell(y, T_{u_1,u_2}(x))], \quad \widehat{R}(u_1, u_2) = \frac{1}{N} \sum_{i \in \mathcal{I}_{test}} \ell(y_i, T_{u_1,u_2}(x_i)).$$

Expanding the loss conditional on the tiers yields

$$R(u_1, u_2) = \mathbb{E}\big[\ell(y, \tilde{f}_1(x))\,\mathbf{1}\{U \leq u_1\} + \ell(y, \tilde{f}_2(x))\,\mathbf{1}\{u_1 < U \leq u_2\}\big],$$

because the tier $U > u_2$ uses the expert f and contributes zero loss. The risk function is non-decreasing in both coordinates by construction. Fixing u_2, increasing u_1 assigns more inputs to the cheaper $\tilde{f}_1$, which weakly increases the risk. Fixing u_1, increasing u_2 defers less often to f and

assigns more inputs to $\tilde{f}_2$, which also weakly increases the risk. This bi-variate monotonicity enables valid fixed-sequence calibration and preserves the PAC guarantee as in Definition 2.1. We construct a two-dimensional upper confidence bound$\widehat{L}_{u_1,u_2}(\alpha)$ satisfying

$$\mathbb{P}\big(R(u_1,u_2) \le \widehat{L}_{u_1,u_2}(\alpha)\big) \ge 1-\alpha \quad \text{for all valid pairs } (u_1,u_2) \in [0,1]^2,\ u_1 \le u_2.$$

The UCB can be built on the calibration set via importance sampling, using partial expert queries with sampling probabilities π_i and weights, analogously to Algorithms 1 and 3. Specifically, when an expert label y_i is queried, we record two weighted losses $Z_{i,1} = \ell(y_i, \tilde{y}_{i,1})/\pi_i$ and $Z_{i,2} = \ell(y_i, \tilde{y}_{i,2})/\pi_i$, otherwise we record zeros, and then aggregate tier-wise according to (u_1, u_2). Under a central limit theorem or concentration inequalities, we obtain a valid $\widehat{L}_{u_1,u_2}(\alpha)$. Once the UCB is available, we calibrate the thresholds by searching over the empirical grid of unique uncertainty scores in the calibration set and selecting a pair that minimizes computation under the constraint $\widehat{L}_{u_1,u_2}(\alpha) \le \epsilon$. A simple and effective choice is to maximize u_2 subject to validity and then break ties by maximizing u_1, which prioritizes using $\tilde{f}_2$ and $\tilde{f}_1$ more often while respecting the target tolerance. By monotonicity and the validity of $\widehat{L}_{u_1,u_2}(\alpha)$, the selected pair $(\hat{u}_1, \hat{u}_2)$ ensures $R(\hat{u}_1, \hat{u}_2) \le \epsilon$ with probability at least $1-\alpha$. The multi-level extension therefore preserves the PAC efficiency improvement while enabling finer control over computation across multiple tiers.

Generalization to K-tier PAC reasoning This framework extends directly to K-tier systems with $K-1$ non-thinking LRMs ordered by cost and accuracy. Let $\tilde{f}_1, \ldots, \tilde{f}_{K-1}$ be the non-thinking LRMs and introduce thresholds $0 \le u_1 \le \cdots \le u_{K-1} \le 1$. Define the deployment mapping for a threshold vector $\mathbf{u} = (u_1, \ldots, u_{K-1})$ as

$$T_{\mathbf{u}}(x) = \tilde{f}_1(x)\,\mathbf{1}\{U(x) \le u_1\} + \sum_{k=2}^{K-1} \tilde{f}_k(x)\,\mathbf{1}\{u_{k-1} < U(x) \le u_k\} + f(x)\,\mathbf{1}\{U(x) > u_{K-1}\}.$$

The population risk is

$$R(\mathbf{u}) = \mathbb{E}\Big[\ell\big(y, \tilde{f}_1(x)\big)\,\mathbf{1}\{U \le u_1\} + \sum_{k=2}^{K-1} \ell\big(y, \tilde{f}_k(x)\big)\,\mathbf{1}\{u_{k-1} < U \le u_k\}\Big],$$

which is coordinate-wise non-decreasing in each threshold u_k. A valid upper confidence bound $\widehat{L}_{\mathbf{u}}(\alpha)$ is constructed by recording $K-1$ weighted losses when querying the expert and aggregating tier-wise. Threshold calibration proceeds on the empirical grid to minimize computation under the constraint $\widehat{L}_{\mathbf{u}}(\alpha) \le \epsilon$, and the fixed-sequence strategy applies under monotonicity. The multi-tier deployment then uses $T_{\mathbf{u}}$ on the test set.

H Experimental Details

Hyperparameter settings of LLMs In this study, we configure the decoding parameters as follows: for Qwen/Qwen3-4B-Instruct-2507, we set *Temperature* = 0.7, *TopP* = 0.8, *TopK* = 20, and *MinP* = 0; for Qwen/Qwen3-4B-Thinking-2507, we set *Temperature* = 0.6, *TopP* = 0.95, *TopK* = 20, and *MinP* = 0. Experiments were run on one NVIDIA RTX A6000 Graphics Card.

The prompt for verbalized uncertainty score In Table 2, we present the prompt used to elicit the verbalized confidence scores. After ten trials, we obtained the average confidence score and defined the verbalized uncertainty score as 1 minus this average confidence.

Details of Datasets Table 3 summarizes the datasets employed in our experiments, together with their corresponding splitting strategies. For each dataset, we report its type, overall size, and the partitioning into PAC calibration and PAC test sets.

H.1 Baselines

We compare PAC reasoning with several representative efficiency-oriented baselines, including Naive control, learned routing Feng et al. (2025), prompting-based efficient reasoning (Xu et al., 2025),

Table 2: Prompt for the verbalized confidence scores.

System prompt: You are a reasoning assistant. For each question and proposed answer, you must estimate how likely the proposed answer is correct.
User prompt:
Question: {QUESTION}
Answer: {ANSWER}
Provide a probability (between 0.0 and 1.0) that your answer is correct. Only output the probability.

Table 3: The details of datasets and splitting settings for PAC experiments

Dataset	Dataset Type	Dataset Size	Split Setting	Size
MATH-500	Math Reasoning	500	PAC Calibration	300
			PAC Test	200
ZebraLogic	Text reasoning	1000	PAC Calibration	500
			PAC Test	500
Arena-Hard	Alignment Task	750	PAC Calibration	450
			PAC Test	300

and reasoning-free generation Ma et al. (2025). While these methods have demonstrated empirical effectiveness in reducing inference cost, they are primarily heuristic and do not provide explicit theoretical guarantees on performance loss.

Naive control We include a naive baseline, which closely resembles our method but omits the procedure of UCB. Specifically, given a target error tolerance ϵ, this baseline selects the largest threshold u such that the empirical loss

$$L(u) = \frac{1}{n}\sum_{i=1}^{n} \ell(y_i, \tilde{y}_i)\mathbf{1}\{U_i \leq u\}$$

is below ϵ on the calibration set. This approach directly matches the empirical loss without accounting for the statistical estimation of $L(u)$, and therefore lacks the inductive, high-probability guarantees provided by our PAC-based reasoning.

Router We train a learned router for the model pair using the open-source library LLM-ROUTER (Feng et al., 2025). Specifically, we sample 2,000 instances from eight widely used benchmarks, including ARC-Challenge (Clark et al., 2018), CommonsenseQA (Talmor et al., 2019), GSM8K (Cobbe et al., 2021), MATH (Hendrycks et al., 2021b), HumanEval (Chen et al., 2021), MMLU (Hendrycks et al., 2021a), NaturalQA (Kwiatkowski et al., 2019), and TriviaQA (Joshi et al., 2017). The router is implemented as a lightweight classifier that takes the input prompt as features and outputs a routing score, representing the predicted probability that invoking the thinking model is necessary. At inference time, inputs are routed to the thinking model if the predicted probability exceeds a fixed threshold of 0.5, and otherwise handled by the non-thinking model. Such router does not provide explicit guarantees on the resulting performance loss after routing.

Chain of Draft Chain of Draft (CoD) Xu et al. (2025) is a prompting-based method designed to improve reasoning efficiency by explicitly constraining the verbosity of intermediate reasoning steps. Instead of generating full chain-of-thoughts, CoD enforces a compact "draft" for each reasoning step, typically limited to a few words, thereby reducing token usage while preserving a minimal reasoning structure. The specific prompt template used in our experiments is shown in Table 4. As a prompting strategy, CoD focuses on reducing reasoning length, but does not control the performance loss induced by truncated reasoning.

NoThinking NoThinking Ma et al. (2025) eliminates explicit chain-of-thought reasoning by removing special reasoning markers (e.g., `<think>`) and relying on direct answer generation combined with simple aggregation strategies. The method shows that, for certain reasoning tasks, models can achieve competitive performance without explicitly generating intermediate reasoning steps, leading to reduced inference cost and latency. However, NoThinking does not incorporate uncertainty-aware

Table 4: Prompt for Chain of Draft.

System prompt: Think step by step, but only keep a minimum draft for each thinking step, with at most five words. Finally, put your final answer within `\boxed{}`. **User prompt:** Question: {QUESTION}

decision mechanisms, nor does it provide explicit control or theoretical guarantees over the resulting performance degradation.

H.2 Choice of loss functions

The loss functions, shown as in Eq. (6) and Eq. (7), serve distinct purposes in evaluating the PAC reasoning. The semantic cosine distance captures the degree of semantic alignment between the PAC reasoning's prediction and reference outputs. It is particularly suitable for tasks where nuanced differences in meaning are critical, such as natural language understanding or generation tasks. By leveraging the "Qwen/Qwen3-Embedding4B" model, we ensure that the embeddings capture rich contextual information, robustly comparing semantic content in high-dimensional spaces. In contrast, the binary 0–1 loss is designed for scenarios where the correctness of the generated answer is verifiable, such as in mathematical problem-solving or multiple-choice question answering. This loss function is particularly effective for evaluating the framework's ability to produce exact matches to ground-truth answers, emphasizing precision in verifiable tasks. By testing the PAC reasoning on these two loss functions, we can assess the semantic quality and factual accuracy of the PAC reasoning across diverse tasks.

I Additional experiments

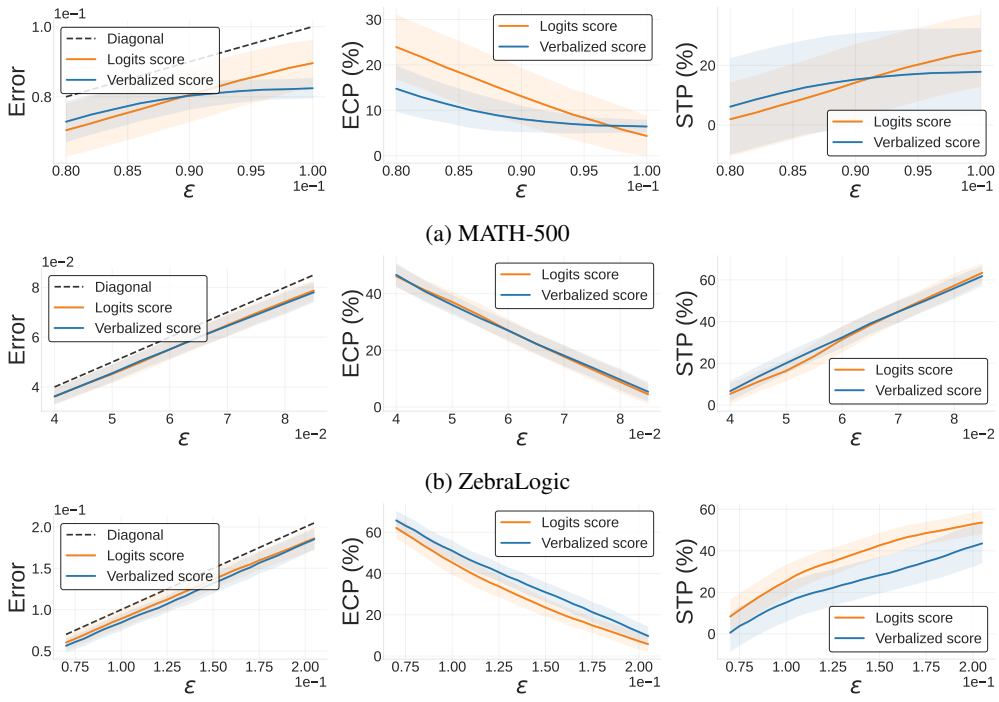
I.1 Experimental results of Llama-based LLMs

In this section, we evaluate PAC reasoning on additional LLM architectures and larger-scale models to further verify the generalizability of our framework. Specifically, we conduct experiments using Llama-3.1-8B–based models: the "DeepSeek-R1-Distill-Llama-8B" as the thinking model and "Llama-3.1-8B-Instruct" as the lower-performance non-thinking model. We configure the decoding parameters as follows: for Llama-3.1-8B-Instruct, we set *Temperature* = 0.6, *TopP* = 0.95, *TopK* = 20, and *MinP* = 0, *max_tokens* = 4096; for DeepSeek-R1-Distill-Llama-8B, we set *Temperature* = 0.6, *TopP* = 0.95, *TopK* = 20, and *MinP* = 0. Experiments were run on one NVIDIA RTX A6000 Graphics Card. Other experimental details are following Appendix H.

Figure 2 summarizes the performance of PAC reasoning on Llama-3.1-8B–based models. Across all three benchmarks, PAC reasoning consistently maintains valid performance loss control, with empirical performance loss staying below the diagonal reference line. For uncertainty estimation, the logits-based score exhibits tighter calibration and lower ECP than the verbalized score, particularly under smaller ϵ values, while the verbalized score shows slightly higher variance but still adheres to theoretical bounds. In terms of efficiency, both scores achieve substantial STP, demonstrating that PAC reasoning can reliably identify confident cases and reduce unnecessary calls to the thinking model. Overall, the results confirm that PAC reasoning generalizes well to larger LLM architectures and continues to deliver stable risk control and efficiency gains.

I.2 Additional benchmarks

In this section, we further validate the effectiveness of our framework on additional datasets, including GPQA (Rein et al., 2024) and HumanEval (Chen et al., 2021). The experimental results shown in Figure 6 evaluate the performance of PAC reasoning on these benchmarks using two types of uncertainty scores: the logits-based score and the verbalized score. Across both datasets, the two uncertainty scores demonstrate valid error control. The ECP increases steadily as the tolerance level ϵ grows, with the verbalized score exhibiting slightly worse ECP performance. For STP, the logits-based score consistently achieves higher performance compared to the verbalized score. Overall,



(a) MATH-500

(b) ZebraLogic

(c) Arena-Hard

Figure 2: **Error control, ECP and STP of PAC reasoning** for semantic loss across three benchmarks under a confidence level of $\alpha = 0.05$, on the Llama-3.1-8B–based models. Uncertainty score includes the logits-based score and the verbalized score. All experiments are repeated 100 times, and the shaded areas represent standard deviations.

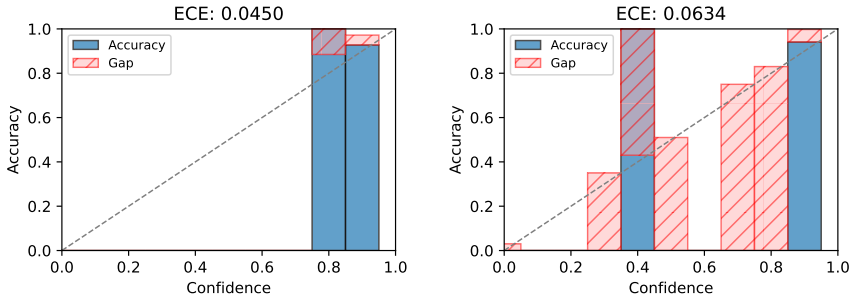
these observations indicate that PAC reasoning remains effective and reliable when applied to a broader range of datasets.

I.3 Expected calibration error of two uncertainty scores

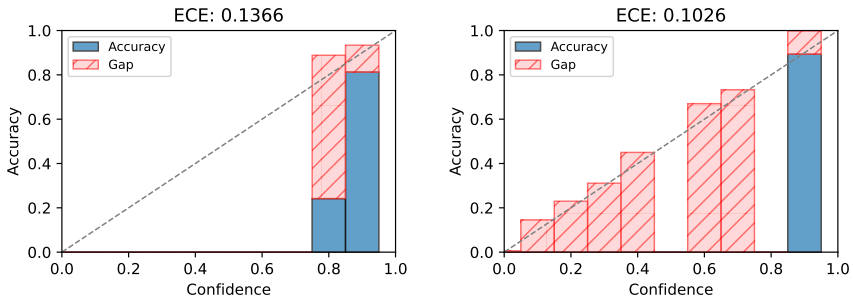
We evaluate the calibration quality of uncertainty estimates on MATH-500 and ZebraLogic using Qwen3-4B-Instruct-2507. We consider two uncertainty scores: a logits-based score derived from the model's predictive distribution and a verbalized score elicited from the model's self-reported confidence. Expected calibration error (ECE) (Guo et al., 2017) quantifies the discrepancy between predicted confidence and empirical accuracy via binning and a weighted average of absolute gaps, where smaller values indicate better calibration. Across both benchmarks, the logits-based score exhibits consistently lower ECE and smoother reliability than the verbalized score, indicating tighter calibration of uncertainty estimates. The verbalized score shows higher variance and mild overconfidence in high-confidence bins. These findings support the use of the logits-based score within PAC reasoning and motivate improved elicitation methods for verbalized confidence. Figure 3 summarizes the reliability plots and aggregated ECE. All experiments are repeated 100 times under the decoding configuration described in Section 4.1.

I.4 Ablation study on the size of the calibration set

We conduct experiments to investigate the stability of efficiency gains under different correction set sizes. Specifically, we repeat the experiments with varying calibration ratios to examine how the size of the correction set influences performance. The results are presented in Figure 4. PAC reasoning maintains stable error control and consistent uncertainty calibration across all benchmarks. Both uncertainty scoring methods are capable of controlling the theoretical risk, though the verbalized score exhibits larger variance. Moreover, the logits-based score consistently outperforms the verbalized



(a) MATH-500 (Logits-based) (b) MATH-500 (Verbalized)



(c) ZebraLogic (Logits-based) (d) ZebraLogic (Verbalized)

Figure 3: Expected Calibration Error across two mathematical benchmarks.

uncertainty score, achieving lower ECP and higher STP. These findings demonstrate that our framework, i.e., PAC reasoning, can effectively maintain valid risk control and stable efficiency gains under varying calibration dataset sizes.

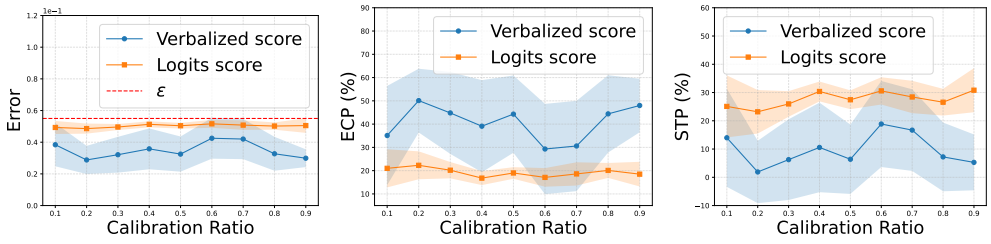
I.5 Reward score as an alternative uncertainty score

In this part, we evaluate whether PAC reasoning remains valid when replacing the original uncertainty score with the reward score. Concretely, we apply PAC reasoning to MATH-500 and ZebraLogic using the reward score as the uncertainty estimate, and we report its error control based on the semantic cosine distance, ECP, and STP under varying ϵ. We follow the experimental setting described in Section 4.1, and the reward model is "Qwen2.5-Math-PRM-7B" (Zhang et al., 2025c).

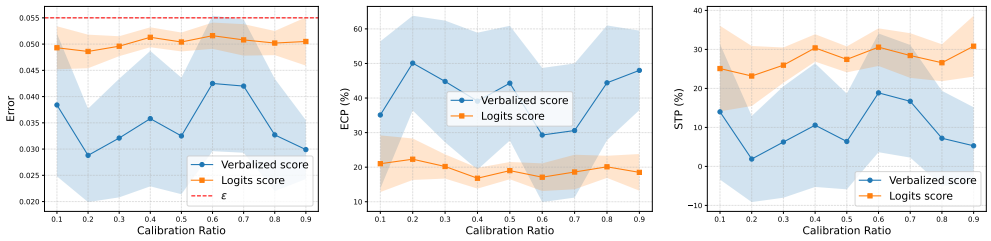
The results are presented in Figure 5. Across both benchmarks, the observed error curves remain below the diagonal baseline, indicating that PAC reasoning still satisfies the theoretical error guarantee even with this alternative scoring method. For efficiency, ECP consistently decreases as ϵ increases, showing that the method becomes more selective. These results show that PAC reasoning is robust to the choice of uncertainty score: using the reward score still ensures valid error control and provides reasonable efficiency gains.

I.6 Accuracy of PAC reasoning

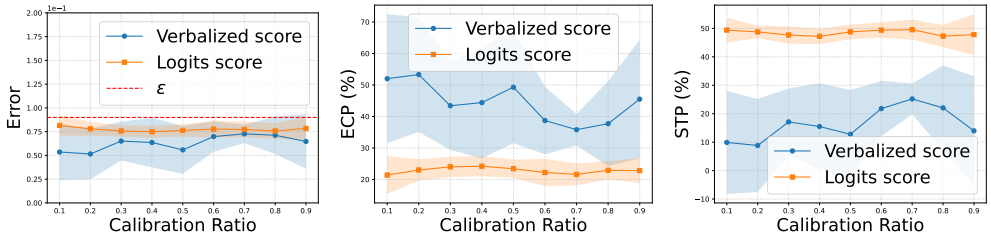
In this section, we investigate the effectiveness of the PAC reasoning framework when controlled by semantic loss. As shown in Figure 7, applying semantic loss to regulate the PAC filtering process leads to consistently improved final accuracies across both MATH-500 and ZebraLogic. Both logits-based and verbalized uncertainty scores yield higher Pass@1 performance than the baseline non-thinking model, with the verbalized score performing the best. The results indicate that PAC reasoning controlled by semantic loss reliably enhances output accuracy while remaining robust to different ϵ settings.



(a) MATH-500

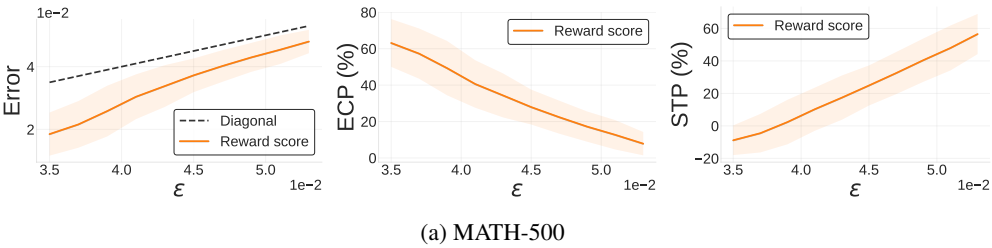


(b) ZebraLogic

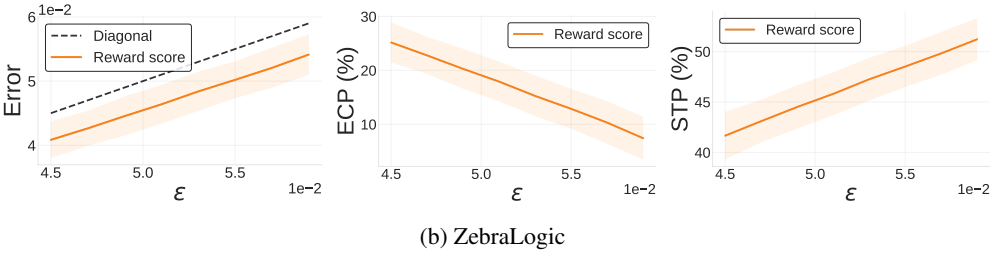


(c) Arena-Hard

Figure 4: Error control, ECP and STP of PAC reasoning for semantic loss for different calibration ratios at a confidence level of $\alpha = 0.05$. Uncertainty score includes the logits-based score and the verbalized score. The red dashed line ϵ means the target risk level, and the shaded areas represent standard deviations. All experiments are repeated 100 times.

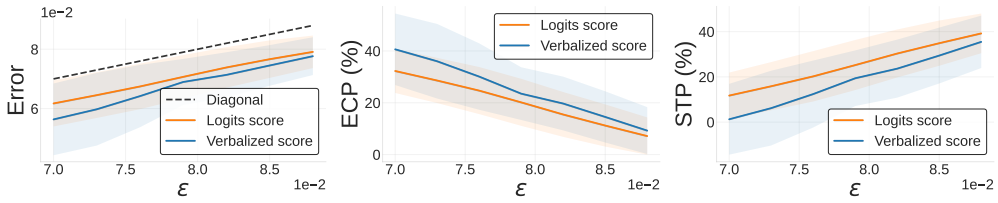


(a) MATH-500

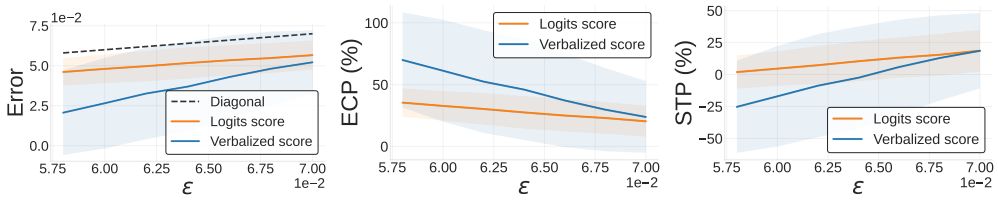


(b) ZebraLogic

Figure 5: Error control, ECP and STP of PAC reasoning for semantic loss across two mathematical benchmarks at a confidence level of $\alpha = 0.05$. Uncertainty score is the **reward score**. All experiments are repeated 100 times, and the shaded areas represent standard deviations.

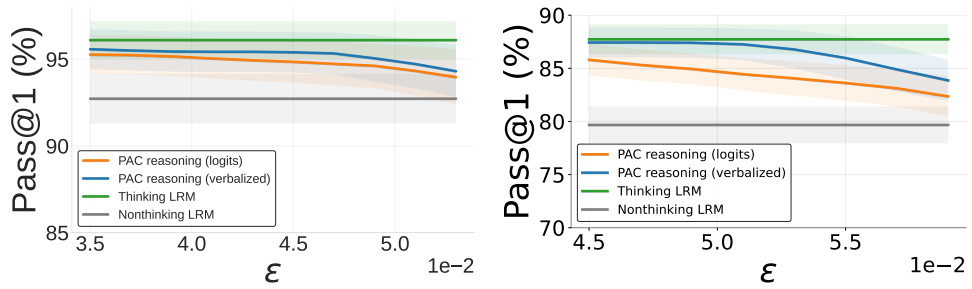


(a) GPQA



(b) HumanEval

Figure 6: Error control, ECP and STP of PAC reasoning for semantic loss for GPQA and HumanEval at a confidence level of $\alpha = 0.05$. Uncertainty score includes the logits-based score and the verbalized score. The red dashed line ϵ means the target risk level, and the shaded areas represent standard deviations. All experiments are repeated 100 times.



(a) MATH-500

(b) ZebraLogic

Figure 7: Accuracy of PAC reasoning based on semantic loss across mathematical benchmarks at a confidence level of $\alpha = 0.05$. Uncertainty score includes the logits-based score and the verbalized score. All experiments are repeated 100 times, and the shaded areas represent standard deviations.

Table 5: Experimental results of the binary loss function on verifiable datasets ($\alpha = 0.05$). For MATH-500, we set $\epsilon = 0.03$, and for ZebraLogic, we set $\epsilon = 0.08$.

Dataset	Metric	Logits-based score		Verbalized score		non-thinking
		PAC reasoning	Naive ($U_i \geq 0.05$)	PAC reasoning	Naive ($U_i \geq 0.05$)	
MATH-500	Error	0.0206 ± 0.0126	0.0179 ± 0.0068	0.0209 ± 0.0141	0.0346 ± 0.0095	0.0435 ± 0.0107
	ECP (%) ↓	21.48 ± 17.85	14.44 ± 2.02	24.59 ± 20.48	2.83 ± 0.94	–
	STP (%) ↑	37.61 ± 23.19	43.58 ± 4.78	36.13 ± 26.44	66.67 ± 4.91	–
ZebraLogic	Error	0.0615 ± 0.0181	0.0062 ± 0.0026	0.0530 ± 0.0246	0.0631 ± 0.0074	0.1163 ± 0.0102
	ECP (%) ↓	22.50 ± 7.47	77.28 ± 1.36	26.95 ± 20.68	12.49 ± 1.07	–
	STP (%) ↑	23.13 ± 9.90	-34.78 ± 1.26	21.21 ± 17.20	32.70 ± 2.11	–

I.7 Additional results for the binary loss

For the binary loss, we evaluate PAC reasoning on the verifiable datasets MATH-500 and ZebraLogic, with target risk levels set to $\epsilon = 0.03$ and $\epsilon = 0.08$, respectively (see Section 4.1 for experimental details). For comparison, we also consider a naive fixed-threshold baseline as well as the approach that relies solely on the non-thinking model.

As shown in Table 5, PAC reasoning consistently keeps the error rates below the target risk, while also achieving substantial efficiency gains. In contrast, the naive baseline exhibits unstable behavior across datasets: on ZebraLogic, although it attains a very small error with logits-based uncertainty, it violates efficiency by yielding a negative STP (-34.78%), meaning it requires even more tokens than fully using the thinking model. Meanwhile, on MATH-500 with verbalized uncertainty, the same method produces a large error (0.0346), which substantially exceeds the target risk $\epsilon = 0.03$. These results highlight that naive threshold fails to provide reliable control over both loss and budget, often swinging between overly conservative and overly risky outcomes. In summary, PAC reasoning strikes a balanced trade-off, keeping the error within ϵ while delivering consistent savings across tasks and datasets.