

# Conditional Performance Guarantee for Large Reasoning Models

Jianguo Huang<sup>1\*</sup> Hao Zeng<sup>2\*</sup> Bingyi Jing<sup>3,2</sup> Hongxin Wei<sup>2</sup> Bo An<sup>1</sup>

## Abstract

Large reasoning models have shown strong performance through extended chain-of-thought reasoning, yet their computational cost remains significant. Probably approximately correct (PAC) reasoning provides statistical guarantees for efficient reasoning by adaptively switching between thinking and non-thinking models, but the guarantee holds only in the marginal case and does not provide exact conditional coverage. We propose G-PAC reasoning, a practical framework that provides PAC-style guarantees at the group level by partitioning the input space. We develop two instantiations: Group PAC (G-PAC) reasoning for known group structures and Clustered PAC (C-PAC) reasoning for unknown groupings. We prove that both G-PAC and C-PAC achieve group-conditional risk control, and that grouping can strictly improve efficiency over marginal PAC reasoning in heterogeneous settings. Our experiments on diverse reasoning benchmarks demonstrate that G-PAC and C-PAC successfully achieve group-conditional risk control while maintaining substantial computational savings.

## 1. Introduction

Large language models (LLMs) demonstrate strong reasoning capabilities through extended chain-of-thought generation (DeepSeek-AI et al., 2025; Yang et al., 2025a). However, these improvements come at the cost of overthinking (Sui et al., 2025; Yue et al., 2025; Aggarwal et al., 2025), resulting in increased latency and computational overhead when deploying LLMs at scale. This shows the importance of balancing reasoning performance and computational efficiency. Recent work on efficient reasoning (Zeng et al., 2025), called *PAC reasoning*, addresses this challenge by using a statistical framework based on adaptive model se-

lection. Specifically, the framework maintains two models: a thinking model that performs full reasoning and a non-thinking model that produces fast but potentially less accurate responses. PAC reasoning constructs a composite system that adaptively routes inputs to either model based on uncertainty estimates, while providing statistical guarantees on the expected performance loss.

While PAC reasoning provides a statistical solution for efficient reasoning, its guarantees are marginal: the performance bound holds on average over the entire input distribution, allowing arbitrarily large errors on subpopulations. Such marginal guarantees are insufficient in many high-stakes applications. For instance, in medical diagnosis, practitioners require reliable guarantees for specific diseases, rather than guarantees that hold only on average across the population. This raises a fundamental question:

*Can we provide performance loss guarantees conditional on each group?*

We propose a novel statistical method, termed *Group PAC (G-PAC) reasoning*. Instead of seeking impossible per-input guarantees (Zeng, 2025), G-PAC partitions the input space into groups and provides PAC guarantees at the group level, enabling explicit control of performance loss within each group. When group information is not available a priori, we augment G-PAC with a clustering step, referred to as *Clustered PAC (C-PAC) reasoning*, which learns the grouping from calibration data using uncertainty scores. We establish theoretical guarantees for G-PAC under both known and learned grouping settings. For known groupings, we prove group-wise PAC guarantees together with finite-sample risk bounds. For learned groupings, we characterize the oracle optimal partition and show that grouping strictly improves efficiency under heterogeneity. Finally, we provide coverage guarantees for the learned partitions and show that the resulting coverage gap vanishes asymptotically.

We conduct comprehensive experiments in Section 5 evaluating G-PAC reasoning across diverse reasoning benchmarks, including MATH-500 (Lightman et al., 2023), ZebraLogic (Lin et al., 2025), and GPQA (Rein et al., 2024). Our results show that G-PAC reasoning effectively controls group-conditional loss while substantially reducing inference cost. For instance, on MATH-500 with tolerance

\*Equal contribution <sup>1</sup>Nanyang Technological University, Singapore <sup>2</sup>Southern University of Science and Technology, China <sup>3</sup>The Chinese University of Hong Kong, Shenzhen, China. Correspondence to: Hongxin Wei <weihx@sustech.edu.cn>.

$\epsilon = 0.05$  under the logits-based score, G-PAC achieves zero group-conditional error gap, whereas vanilla PAC fails to control group-wise risk. Moreover, even when group partitions are not available a priori, G-PAC remains effective by learning group structure directly from data.

Our main contributions are as follows.

- We formalize group-conditional PAC efficiency, a practical relaxation that provides statistical guarantees at the group level while enabling fine-grained control over the accuracy-efficiency trade-off.
- We design a practical PAC reasoning method, *G-PAC reasoning*, together with a clustering-based extension, *C-PAC reasoning*, for settings where group information is unavailable, and show that both satisfy group-conditional PAC efficiency in practice.
- We establish theoretical guarantees showing that group-conditional risk can be provably controlled under both known and learned groupings. Building on this result, we show that grouping strictly improves efficiency under heterogeneity, characterize the oracle optimal partition, and prove that the coverage gap vanishes as the learned partition approaches the oracle.

### 1.1. Related work

**Efficient reasoning for large language models.** LLMs (DeepSeek-AI et al., 2025; Yang et al., 2025a) have shown strong performance in complex reasoning tasks, often at a high computational cost, i.e., *overthinking* (Chen et al., 2025; Sui et al., 2025). To address this issue, recent works propose efficient reasoning strategies such as early exit (Yang et al., 2025b; Jiang et al., 2025) and adaptive switching of a single LLM between thinking and non-thinking modes (Cheng et al., 2025; Chung et al., 2025; Fang et al., 2025; Ma et al., 2025; Xiao & Gan, 2025; Yong et al., 2025). Model routing and cascading approaches (Dekoninck et al., 2025) select among multiple LLMs based on query difficulty. Despite their empirical effectiveness, these techniques lack theoretical guarantees on the resulting performance degradation. PAC reasoning (Zeng et al., 2025) addresses this limitation by providing statistical guarantees that characterize the trade-off between computational efficiency and accuracy. However, these guarantees are *marginal*, holding only in expectation over the entire input distribution. We move beyond marginal analysis and propose *group-conditional PAC-efficient reasoning*, which enables fine-grained control of performance loss across different groups of inputs.

**PAC learning and distribution-free risk control.** PAC learning (Valiant, 1984) establishes how algorithms can generalize from training data with probabilistic guaran-

tees. The Learn-then-Test (LTT) framework (Angelopoulos et al., 2025a; Bates et al., 2021) provides a modern, distribution-free approach to PAC-style guarantees, enabling explicit specification of error tolerances. Conformal risk control (Angelopoulos et al., 2025b) extends this framework to control the expected value of monotone loss functions. Despite their generality, these methods primarily yield *marginal* guarantees that hold only on average over the input distribution, and may fail to control performance loss for particular subpopulations. This limitation naturally raises the question of whether we can achieve conditional guarantees that hold for each input group. Our group-conditional PAC-efficient reasoning framework addresses this challenge by providing guarantees at the group level, so it enables clear computational efficiency gains across subpopulations.

## 2. PAC reasoning

We introduce *PAC reasoning* (Zeng et al., 2025), which only achieves marginal PAC efficiency. However, this marginal guarantee may lead to uncontrolled risk on individual input, motivating the need for a conditional PAC efficiency.

*PAC reasoning* formalizes the LLM’s selective thinking with a PAC guarantee on the performance risk. Let  $x \sim \mathcal{P}$  denote an input prompt drawn from a data distribution  $\mathcal{P}$ ,  $f$  a thinking LLM that generates extended reasoning chains, and  $\tilde{f}$  a non-thinking LLM that generates direct answers without thinking parts. Then, PAC reasoning constructs a composite model  $\hat{f}$  that routes inputs between  $f$  and  $\tilde{f}$  using an uncertainty score  $U(x) \in [0, 1]$  and a calibrated threshold  $u$ . Formally, the routing rule is given by:

$$\hat{f}(x) = \begin{cases} \tilde{f}(x) & \text{if } U(x) \leq u, \\ f(x) & \text{else,} \end{cases}$$

where  $U(x)$  is a score to quantify the uncertainty of the non-thinking LLM on the input  $x$ . Then, the performance loss of the composite model  $\hat{f}$  is defined as:

$$R(\hat{f}) = \mathbb{E}_{x \sim \mathcal{P}}[\ell(\hat{f}(x), f(x))],$$

where  $\ell$  is a loss function that measures the performance degradation of  $\hat{f}$  relative to the thinking LLM  $f$ , such as the 0-1 loss for verifiable answers.

**Remark 1.** Notably, the loss is measured *relative to the thinking model’s output* (rather than the ground truth), as our goal is to bound the additional degradation induced by routing away from the thinking model.

To control performance loss with a theoretical guarantee, we introduce the definition of PAC-efficient:

**Definition 2.1** (PAC-efficient). Given an error tolerance  $\varepsilon > 0$  and a confidence level  $\alpha \in (0, 1)$ , a composite model

$\hat{f}$  is  $(\varepsilon, \alpha)$ -PAC-efficient if

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

Definition 2.1 provides a marginal guarantee that controls the expected performance loss averaged over the entire input distribution. PAC reasoning offers a model-free procedure to achieve such PAC-efficient bounds. However, the marginal guarantee does not impose any constraint on the performance loss within specific input subpopulations. As a result, the composite model may satisfy the global PAC guarantee while still exhibiting substantially larger losses on certain groups, potentially violating the target tolerance at the group level. This limitation motivates the need for a finer-grained notion of PAC efficiency that provides conditional guarantees beyond the marginal setting.

**Impossible full conditional PAC efficiency** A natural strengthening is to require the guarantee to hold for each input  $x$  separately. For a specific input  $x$ , define the conditional risk  $R(\hat{f} | x) = \mathbb{E}_{\mathcal{D}_{\text{cal}}}[\ell(\hat{f}(x), f(x)) | x]$ , where the expectation is taken over the randomness of the calibration set. A composite model  $\hat{f}$  is  $(\varepsilon, \alpha)$ -pointwise PAC efficient if for every  $x \in \mathcal{X}$ ,

$$\mathbb{P}(R(\hat{f} | x) \leq \varepsilon) \geq 1 - \alpha. \quad (1)$$

Unfortunately, achieving such per-instance guarantees is impossible without sacrificing efficiency (Zeng, 2025): *the only way to satisfy pointwise PAC efficiency is to use the expert model for almost every  $x \in \mathcal{X}$* .

### 3. Group-conditional PAC-efficient reasoning

To address the limitations of standard PAC reasoning, we introduce the *group-conditional PAC efficiency* and propose methods termed *Group PAC (G-PAC) reasoning* and *Clustered PAC (C-PAC) reasoning*, to control performance loss within known and unknown groups of partitions.

#### 3.1. Group PAC (G-PAC) reasoning

Let  $\mathcal{G} = \{G_1, \dots, G_k\}$  be a partition of the input space  $\mathcal{X}$  with  $k$  groups. For each group  $G_j$ , we define the group-conditional risk function:

$$R(\hat{f} | G_j) = \mathbb{E}_{x \sim \mathcal{P}}[\ell(\hat{f}(x), f(x)) | x \in G_j].$$

We now introduce a group-conditional PAC efficiency:

**Definition 3.1** (Group-conditional PAC efficiency). Given an error tolerance  $\varepsilon > 0$  and a confidence level  $\alpha \in (0, 1)$ , a composite model  $\hat{f}$  is  $(\varepsilon, \alpha)$  group-conditional PAC-efficient with respect to partition  $\mathcal{G}$  if for  $G_j \in \mathcal{G}$ ,

$$\mathbb{P}_{\mathcal{D}_j}(R(\hat{f} | G_j) \leq \varepsilon) \geq 1 - \alpha,$$

where  $\mathcal{D}_j = \{x_i \sim \mathcal{P} : x_i \in G_j\}$  is the data for group  $G_j$ .

This definition is stronger than the marginal PAC guarantee, as it controls performance loss within each group  $G_j \in \mathcal{G}$ , but weaker than the full conditional PAC guarantees in Eq. (1). Group-conditional PAC efficiency offers a practical way to control group-specific performance loss. It relaxes the conception of fully conditional PAC efficiency by balancing feasibility with efficiency. Figure 1 illustrates the trade-off between feasibility and the strength of conditional guarantees, from marginal PAC efficiency to full conditional PAC efficiency.

**Remark 2.** (Group partition) The group partition  $\mathcal{G}$  may be specified a priori or inferred from data, depending on the availability of group information. Many benchmarks provide group labels, such as subject areas in MATH-500 (Lightman et al., 2023) and difficulty-level annotations in ZebraLogic (Lin et al., 2025). When no grouping information is available, the partition must be learned from data via clustering (Section 3.2).

**Group-conditional calibration.** To achieve group-conditional PAC efficiency, we introduce a statistical framework termed *G-PAC reasoning*. Suppose we are given a calibration dataset  $\mathcal{D}_{\text{cal}} = \{(x_i, y_i)\}_{i=1}^n$ , where  $y_i = f(x_i)$  denotes the output of the thinking model.

To control the group-conditional risk below a tolerance  $\varepsilon$  with confidence level  $1 - \alpha$ , we construct an upper confidence bound (UCB)  $\widehat{L}_{j,u}(\alpha)$  for each group  $G_j$  on the calibration set such that

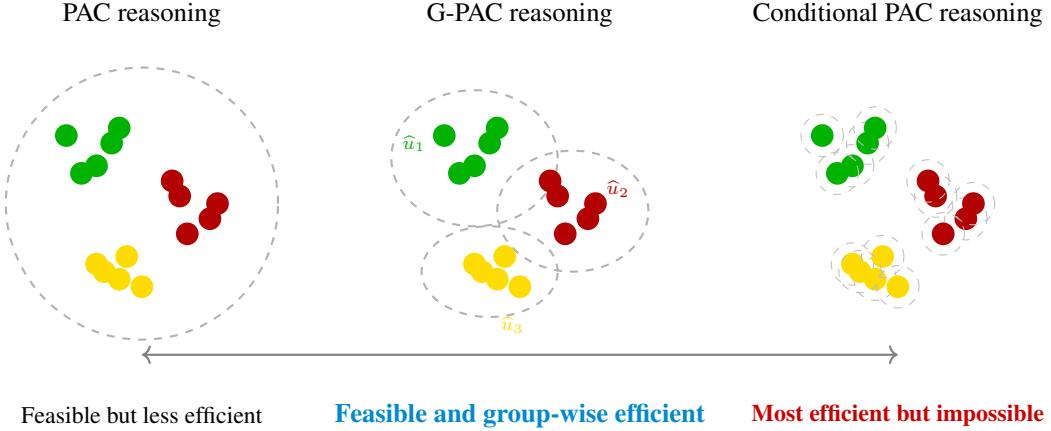
$$\mathbb{P}\left(\widehat{L}_{j,u}(\alpha) \geq R(\hat{f} | G_j)\right) \geq 1 - \alpha,$$

where the probability is taken over the randomness of the calibration data and the importance sampling procedure.

The calibrated threshold  $\widehat{u}_j$  for group  $G_j$  is then defined as the largest uncertainty level for which the UCB remains below the tolerance  $\varepsilon$ :

$$\widehat{u}_j = \max \left\{ u \in [0, 1] : \widehat{L}_{j,u}(\alpha) \leq \varepsilon \right\}. \quad (2)$$

Following PAC reasoning (Zeng et al., 2025), we construct an estimator of the group-conditional performance loss via an importance sampling procedure. Specifically, we sample indices from the calibration set uniformly with replacement and perform Bernoulli trials to decide whether to query expert (thinking-model) outputs. This procedure yields i.i.d. random variables whose expectation equals the target group-conditional performance loss. Based on these samples, we can construct a UCB using different statistical tools, including the central limit theorem (CLT), Hoeffding's inequality (Hoeffding, 1963), and Bernstein's inequality (Bentkus, 2004). In this work, we adopt a CLT-based approach to form a confidence interval for the sample mean, which is suitable for sufficiently large sample sizes. Algorithm 1 summarizes the resulting CLT-based UCB construction. Detailed



**Figure 1. Trade-off between feasibility and efficiency.** Left: PAC reasoning uses a single threshold for all inputs, which is feasible but not the most efficient. Right: conditional PAC reasoning requires per-input guarantees, which is fully efficient but impossible. Middle: G-PAC reasoning balances this trade-off by grouping similar inputs and calibrating dynamic thresholds  $\hat{u}_j$  for each group, making it both feasible and group-conditional efficient.

derivations and alternative concentration-based bounds are provided in Appendix A.

---

**Algorithm 1** UCB( $\mathcal{D}$ ): UCB construction via CLT

**Input:** Data  $\mathcal{D} = \{x_i\}_{i=1}^n$  with model outputs  $y_i = f(x_i)$  and  $\tilde{y}_i = \tilde{f}(x_i)$ , uncertainty scores  $\{U_i\}_{i=1}^n$ , sampling weights  $\{\pi_i\}_{i=1}^n$ , sampling size  $m$ , confidence level  $\alpha$

**Output:** UCB function  $\hat{L}_u(\alpha)$

```

1: Initialize  $\mathcal{Z} \leftarrow []$ 
2: for  $t = 1$  to  $m$  do
3:   Sample index  $i_t \sim \text{Unif}(\{1, \dots, n\})$ 
4:   Sample  $\xi_{i_t} \sim \text{Bern}(\pi_{i_t})$ 
5:    $Z_t \leftarrow \xi_{i_t} \cdot \ell(y_{i_t}, \tilde{y}_{i_t}) / \pi_{i_t}$ 
6:   Append  $Z_t$  to  $\mathcal{Z}$ 
7: end for
8: for candidate  $u \in \mathcal{U}$  do
9:   Compute  $Z_t(u) \leftarrow Z_t \cdot \mathbf{1}\{U_{i_t} \leq u\}$  for all  $t \in [m]$ 
10:   $\hat{\mu}_Z(u) \leftarrow \frac{1}{m} \sum_{t=1}^m Z_t(u)$ 
11:   $\hat{\sigma}_Z(u) \leftarrow \sqrt{\frac{1}{m-1} \sum_{t=1}^m (Z_t(u) - \hat{\mu}_Z(u))^2}$ 
12:   $\hat{L}_u(\alpha) \leftarrow \hat{\mu}_Z(u) + z_{1-\alpha} \cdot \hat{\sigma}_Z(u) / \sqrt{m}$ 
13: end for
14: Return  $\hat{L}_u(\alpha)$ 

```

---

The G-PAC reasoning algorithm operates in two phases: calibration and deployment. In the calibration phase, for each group  $G_j$ , we use its calibration subset  $\mathcal{D}_j$  to estimate group-conditional performance loss and learn a group-specific threshold  $\hat{u}_j$  by constructing an upper confidence bound  $\hat{L}_{j,u}(\alpha)$  such that  $\hat{L}_{j,\hat{u}_j}(\alpha) \leq \varepsilon$ . In the deployment phase, each test input  $x$  is assigned to its corresponding group  $G_j$  and routed according to the learned threshold: it is processed by the non-thinking model if  $U(x) \leq \hat{u}_j$ , and otherwise by the thinking model. Algorithm 2 summarizes

the complete procedure of calibration, while the theoretical analysis is provided in Section 4. Algorithm 2 assumes that the group partition  $\mathcal{G}$  is known *a priori*. When this assumption does not hold, we introduce Clustered PAC (C-PAC) reasoning to infer the partition from calibration data.

---

**Algorithm 2** G-PAC reasoning

**Input:** Calibration set  $\mathcal{D}_{\text{cal}}$ , group partition  $\mathcal{G} = \{G_1, \dots, G_k\}$ , error tolerance  $\varepsilon$ , confidence  $\alpha$

**Output:** Group-specific thresholds  $\{\hat{u}_1, \dots, \hat{u}_k\}$  and composite model  $\hat{f}$

```

1: for each group  $G_j \in \mathcal{G}$  do
2:    $\mathcal{D}_j \leftarrow \{x_i \in \mathcal{D}_{\text{cal}} : x_i \in G_j\}$ 
3:    $\hat{L}_{j,u}(\alpha) \leftarrow \text{UCB}(\mathcal{D}_j)$ 
4:    $\hat{u}_j \leftarrow \max\{u : \hat{L}_{j,u}(\alpha) \leq \varepsilon\}$ 
5:    $\hat{f}_j(x) \leftarrow \begin{cases} \tilde{f}(x) & \text{if } U(x) \leq \hat{u}_j \\ f(x) & \text{otherwise} \end{cases}$ 
6: end for
7:  $\hat{f}(x) \leftarrow \sum_{j=1}^k \hat{f}_j(x) \cdot \mathbf{1}\{x \in G_j\}$ 
8: Return  $\{\hat{u}_1, \dots, \hat{u}_k\}$ , and  $\hat{f}$ 

```

---

The theoretical analysis for G-PAC reasoning is presented in Section 4.1. Specifically, Theorem 4.2 proves that each group achieves the PAC guarantee, and Theorem 4.3 provides a finite-sample PAC guarantee.

### 3.2. Clustered PAC (C-PAC) reasoning

When the group partition is not known in advance, we learn it from calibration data. We use uncertainty-based clustering to discover groups from the uncertainty score  $U(x)$ .

**Clustering in  $U$  space.** We map each calibration point  $x_i$  to a scalar score  $U_i = U(x_i)$ , and cluster the values  $\{U_i\}$ . This is a one-dimensional clustering problem, which is simpler than clustering in the high-dimensional input space  $\mathcal{X}$  (or its embedding space). In practice, we can use standard methods such as 1D k-means or hierarchical clustering to obtain groups with similar uncertainty levels.

We consider two ways to combine clustering and calibration. In the split approach, we split the calibration set into two disjoint subsets,  $\mathcal{D}_{\text{cluster}}$  and  $\mathcal{D}_{\text{cal}}$ : we learn the partition from  $\mathcal{D}_{\text{cluster}}$  and calibrate thresholds on  $\mathcal{D}_{\text{cal}}$ , so the learned partition is independent of the threshold calibration. This yields exact coverage per learned group. In the joint approach, we use the same set for both steps,  $\mathcal{D}_{\text{cluster}} = \mathcal{D}_{\text{cal}}$ , which is more sample-efficient but makes the learned partition and the thresholds dependent; we account for this by adding an extra gap term  $\delta(n, k)$  in the UCB construction. The theoretical analysis is given in Section 4.3, and Theorem 4.6 provides PAC guarantees for both approaches: the split approach achieves exact coverage for each learned group, while the joint approach has an extra gap term  $c \cdot \delta(n, k)$ .

## 4. Theoretical analysis

This section establishes the theoretical guarantees for G-PAC reasoning. We prove group-wise validity for known partitions (Section 4.1), show that oracle grouping improves efficiency under heterogeneity (Section 4.2), and analyze the theoretical results of learned partitions when group partitions are not available *a priori* (Section 4.3).

### 4.1. Validity of G-PAC reasoning

**Assumption 4.1** (UCB validity). For each group  $G_j$ , threshold  $u$ , and a confidence level  $\alpha \in (0, 1)$ , the upper confidence bound  $\widehat{L}_{j,u}(\alpha)$  computed on the group set  $\mathcal{D}_j$  satisfies

$$\mathbb{P}_{\mathcal{D}_j}(R(\widehat{f} | G_j) \leq \widehat{L}_{j,u}(\alpha)) \geq 1 - \alpha,$$

where  $R(\widehat{f} | G_j)$  is the group-conditional performance loss of the composite model  $\widehat{f}_u$  with threshold  $u$ , and the probability is taken over the randomness of the group calibration set  $\mathcal{D}_j = \{x_i \in \mathcal{D}_{\text{cal}} : x_i \in G_j\}$ .

In this work, we use bootstrap-based methods combined with the central limit theorem to satisfy the assumptions above, as described in Algorithm 1. Alternatively, the same assumption can be instantiated using standard concentration inequalities, such as Hoeffding's inequality (Hoeffding, 1963) or empirical Bernstein bounds (Howard et al., 2021).

**Theorem 4.2** (PAC guarantee for known grouping). *Let  $\mathcal{G} = \{G_1, \dots, G_k\}$  be a known partition of the input space. Suppose the loss function  $\ell : \mathcal{Y} \times \mathcal{Y} \rightarrow [0, B]$  is bounded for some constant  $B > 0$ , and Assumption 4.1 holds. If Algo-*

rithm 2 is applied with calibration set  $\mathcal{D}_{\text{cal}}$ , error tolerance  $\varepsilon > 0$ , and confidence level  $\alpha \in (0, 1)$ , for each group  $G_j$ , we have:

$$\mathbb{P}_{\mathcal{D}_j}(R(\widehat{f} | G_j) \leq \varepsilon) \geq 1 - \alpha. \quad (3)$$

Eq. (3) states the group-wise guarantee. See Appendix B for the proof. We also provide an empirical version of the guarantee that bounds the test risk on a finite test set.

**Theorem 4.3** (PAC finite-sample guarantee for known grouping). *Suppose the loss function  $\ell : \mathcal{Y} \times \mathcal{Y} \rightarrow [0, B]$  is bounded for some constant  $B > 0$ , Assumption 4.1 holds, and the test set  $\mathcal{D}_{\text{test}}$  is independent of the calibration set  $\mathcal{D}_{\text{cal}}$ . Let  $\mathcal{G} = \{G_1, \dots, G_k\}$  be a known partition, and let  $\widehat{u}_j$  be the threshold selected by Algorithm 2 for group  $G_j$ . Given  $\varepsilon, \alpha \in (0, 1)$ , for each group  $G_j$  with  $N_j$  test samples, and any  $t > 0$ ,*

$$\mathbb{P}_{\mathcal{D}_j}\left(\widehat{R}(\widehat{f} | G_j) \leq \varepsilon + t\right) \geq 1 - \alpha - \exp\left(-\frac{2N_j t^2}{B^2}\right),$$

where  $\widehat{R}(\widehat{f} | G_j) = \frac{1}{N_j} \sum_{i: x_i \in G_j} \ell(\widehat{f}(x_i), f(x_i))$  is the empirical risk on the test set for group  $G_j$ .

See Appendix C for the proof.

**Remark 3.** A common case is bounded loss  $\ell \in [0, 1]$ , e.g., 0-1 loss for verifiable answers. Then, the bound simplifies to  $\mathbb{P}(\widehat{R}(\widehat{f} | G_j) \leq \varepsilon + t) \geq 1 - \alpha - e^{-2N_j t^2}$ . This provides exact risk control for each group with probability at least  $1 - \alpha$  up to a slack  $t$  that decreases with test set size.

### 4.2. Oracle partition

We characterize the oracle optimal partition and prove that grouping strictly improves efficiency under heterogeneity. We formalize the notion of an oracle group partition in the PAC framework. For an error tolerance  $\varepsilon > 0$ , and a confidence parameter  $\alpha \in (0, 1)$ , we say a threshold  $u_j$  is  $(\varepsilon, \alpha)$ -PAC feasible for group  $G_j$  if:

$$\mathbb{P}(R(u_j | G_j) \leq \varepsilon) \geq 1 - \alpha,$$

where  $R(u | G_j) = \mathbb{E}_{x \sim \mathcal{P}}[\ell(\tilde{f}(x), f(x)) \cdot \mathbf{1}\{U(x) \leq u\} | x \in G_j]$  is the performance loss conditional on  $G_j$ , and the probability is over the randomness of the calibration data  $\mathcal{D}_j$  for group  $G_j$ . The PAC-optimal threshold for group  $G_j$  is the largest feasible threshold:

$$\widehat{u}_j^*(\varepsilon, \alpha) = \sup\{u : u \text{ is } (\varepsilon, \alpha)\text{-PAC feasible for } G_j\}. \quad (4)$$

Then we define the oracle optimal partition that maximizes the use of the non-thinking model while maintaining PAC guarantees:

**Definition 4.4** (Oracle optimal partition). *Given an error tolerance  $\varepsilon > 0$ , a confidence parameter  $\alpha \in (0, 1)$ , and*

number of groups  $k$ , the oracle optimal partition  $\mathcal{G}^* = \{G_1^*, \dots, G_k^*\}$  is defined as:

$$\mathcal{G}^* = \arg \max_{\mathcal{G} \in \Pi_k(\mathcal{X})} \sum_{j=1}^k p_j \cdot \mathbb{P}(U(x) \leq \hat{u}_j^*(\varepsilon, \alpha) \mid x \in G_j),$$

where  $\Pi_k(\mathcal{X})$  denotes the set of all measurable  $k$ -partitions of the input space  $\mathcal{X}$ ,  $p_j = \mathbb{P}(x \in G_j)$  is the probability mass of group  $G_j$ , and  $\hat{u}_j^*(\varepsilon, \alpha)$  is the PAC-optimal threshold for group  $G_j$  defined in Eq. (4).

**Remark 4** (Efficiency as non-thinking model usage). The PAC efficiency  $\text{Eff}_{\text{PAC}}(\mathcal{G}; \varepsilon, \alpha)$  equals the probability of routing an input to the non-thinking model under the PAC-optimal thresholds. Maximizing efficiency is equivalent to maximizing the use of the non-thinking model while satisfying the PAC guarantee. Since the non-thinking model is faster and cheaper than the thinking model, higher efficiency directly translates to lower computational cost.

**Benefits of grouping.** Intuitively, the oracle optimal partition maximizes the usage of the non-thinking model while ensuring that the performance loss of each group satisfies the  $(\varepsilon, \alpha)$ -PAC guarantee. A natural question is whether grouping provides any benefit over the marginal approach that uses a single threshold for all inputs. The following proposition shows that grouping always weakly improves efficiency, with strict improvement when groups exhibit heterogeneous risk profiles:

**Proposition 4.5** (Group-conditional benefit). *For the optimal partition  $\mathcal{G}$ , the PAC efficiency satisfies*

$$\text{Eff}_{\text{PAC}}(\mathcal{G}; \varepsilon, \alpha) \geq \text{Eff}_{\text{PAC}}(\{\mathcal{X}\}; \varepsilon, \alpha),$$

where  $\{\mathcal{X}\}$  denotes the trivial input space (no grouping). Equality holds if and only if the PAC-optimal thresholds are identical across all groups:  $\hat{u}_1^*(\varepsilon, \alpha) = \dots = \hat{u}_k^*(\varepsilon, \alpha) = \hat{u}^*(\varepsilon, \alpha)$ , where  $\hat{u}^*(\varepsilon, \alpha)$  is the PAC-optimal threshold.

See Appendix D for the proof. Heterogeneity in the conditional risk functions  $R(u \mid G_j)$  arises from different data distributions or non-uniform model capability across groups.

### 4.3. Validity of C-PAC reasoning

Let  $\widehat{\mathcal{G}} = \{\widehat{G}_1, \dots, \widehat{G}_k\}$  be a learned partition and let  $\mathcal{G}^* = \{G_1^*, \dots, G_k^*\}$  be the oracle one. Let  $\widehat{g}, g^* : \mathcal{X} \rightarrow [k]$  be their group assignment functions. We define the *partition gap* as

$$\delta := \min_{\sigma \in S_k} \mathbb{P}_{x \sim \mathcal{P}} (\widehat{g}(x) \neq \sigma(g^*(x))), \quad (5)$$

where  $S_k$  is the set of permutations on  $[k]$ . When  $\delta = 0$ , the learned partition matches the oracle exactly (up to relabeling). We establish coverage guarantees under two approaches: sample splitting uses disjoint sets  $\mathcal{D}_{\text{cluster}}$  and

$\mathcal{D}_{\text{cal}}$  for clustering and threshold calibration, while the joint approach uses the same set  $\mathcal{D}_{\text{cal}}$  for both tasks.

**Theorem 4.6** (PAC guarantee for unknown grouping). *Let  $\widehat{\mathcal{G}} = \{\widehat{G}_1, \dots, \widehat{G}_k\}$  be a learned partition and let  $\delta$  be the partition gap in Eq. (5). Suppose the loss function  $\ell$  is bounded for some constant  $B > 0$ , and Assumption 4.1 holds. If Algorithm 2 is applied with error tolerance  $\varepsilon > 0$  and confidence parameter  $\alpha \in (0, 1)$ , then:*

1. **Sample splitting:** If  $\widehat{\mathcal{G}}$  is learned from  $\mathcal{D}_{\text{cluster}}$  and calibration uses  $\mathcal{D}_{\text{cal}}$  (disjoint from  $\mathcal{D}_{\text{cluster}}$ ), then for each learned group  $\widehat{G}_j$ ,

$$\mathbb{P}_{\mathcal{D}_{\text{cal}}} (R(\widehat{f} \mid \widehat{G}_j) \leq \varepsilon) \geq 1 - \alpha. \quad (6)$$

And, the efficiency satisfies  $\text{Eff}_{\text{PAC}}(\mathcal{G}^*; \varepsilon, \alpha) - \text{Eff}_{\text{PAC}}(\widehat{\mathcal{G}}; \varepsilon, \alpha) \leq 2\delta$ .

2. **Joint approach:** If  $\widehat{\mathcal{G}}$  is learned from the calibration set  $\mathcal{D}_{\text{cal}}$  of size  $n$  (same set used for both clustering and calibration), then with probability at least  $1 - \alpha$ ,

$$R(\widehat{f} \mid \widehat{G}_j) \leq \varepsilon + c \cdot \delta \quad \text{for all } j = 1, \dots, k, \quad (7)$$

for some positive constant  $c$ .

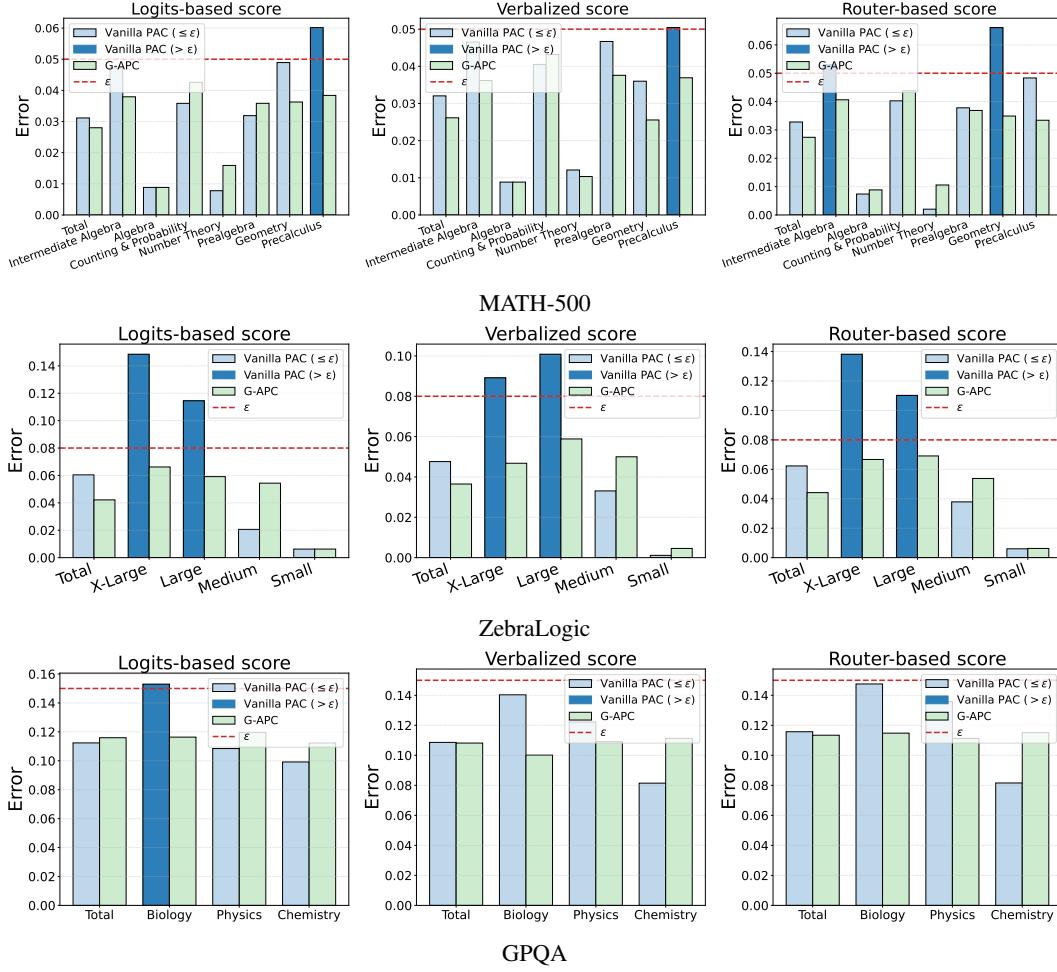
Eq. (6) and (7) analyze the two settings: Sample splitting uses disjoint data for partition learning and calibration, giving exact coverage regardless of the partition gap  $\delta$ . Joint learning reuses the same data, improving efficiency but incurring a coverage gap of order  $c\delta$ . In both cases, a larger  $\delta$  reduces efficiency, but under standard conditions  $\delta = O(n^{-\beta})$  (Lu & Zhou, 2016; von Luxburg et al., 2008), so the joint gap vanishes as  $n$  grows.

## 5. Experiments

In this section, we evaluate G-PAC (for known partitions) and C-PAC (for learned partitions) across multiple LLM benchmarks and deployment settings. Specifically, we investigate two key questions: (i) **Group-conditional risk control**: Can G-PAC and C-PAC effectively control the performance loss for each group, especially when vanilla PAC reasoning fails? (ii) **Trade-off and cost**: What is the cost of achieving group-level risk control in terms of efficiency?

### 5.1. Setup

**Large language models and Datasets.** We evaluate the PAC reasoning based on 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 LLM and “Qwen3-4B-Instruct-2507” as the non-thinking LLM. Moreover, we use “DeepSeek-R1-Distill-Llama-8B” (DeepSeek-AI et al., 2025) as the thinking



**Figure 2. G-PAC controls the group-conditional performance loss below the target while vanilla PAC fails (dark blue) across three different uncertainty scores.** All results are obtained using the Qwen model. The figure reports the overall and per-category performance losses on three reasoning benchmarks.

model and ‘‘Llama-3.1-8B-Instruct’’ (Grattafiori et al., 2024) as the non-thinking model. Further details are provided in Appendix F.1. We conduct experiments on a range of real-world benchmarks, including MATH-500 (Lightman et al., 2023), ZebraLogic (Lin et al., 2025), and GPQA (Rein et al., 2024). Additional details can be found in Appendix F.1.

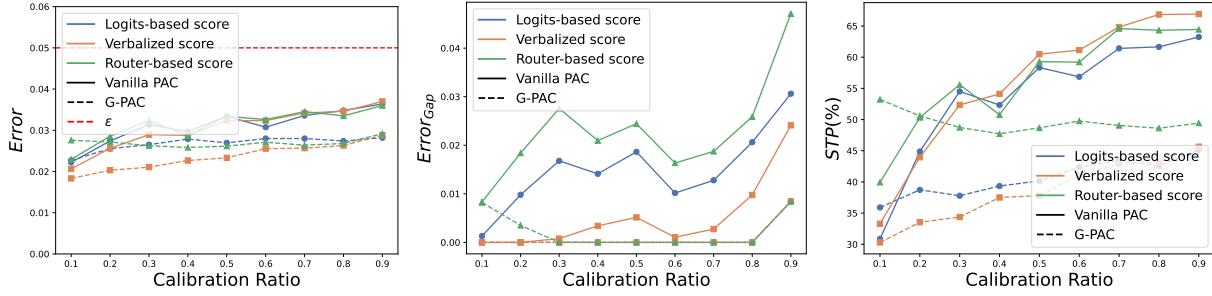
**Uncertainty scores.** In our experiments, we consider three uncertainty scores to support different deployment settings: a logits-based score, a verbalized score, and a router-based score, with definitions provided in Appendix F.2.

**Baselines and evaluation metrics.** We compare the proposed G-PAC reasoning with the naive PAC reasoning (Zeng et al., 2025). To evaluate risk control under both marginal and group-conditional settings, we adopt two evaluation metrics, i.e., Error and Error<sub>Gap</sub>, to measure the empirical risk and its gap on the test dataset. In addition, we use *Saved Token Percentage* (STP) to quantify computational cost savings. Details are described in detail in Appendix F.3.

Across all experiments, we fix the confidence level at  $\alpha = 0.05$  and the sampling weight at  $\pi = 0.5$ , while varying  $\epsilon$ . Each experiment is repeated 100 times, and we report the mean error and cost savings, using the binary loss to quantify performance loss (see Appendix F.4).

## 5.2. Results

**G-PAC and C-PAC successfully control the group-conditional performance loss below the target while vanilla PAC fails.** Table 1 and Table 5 report the results of G-PAC reasoning under known and unknown partitions. In the known case, we use the predefined partitions from each dataset. For C-PAC, groups are made by clustering uncertainty scores into 3 bins. Across all datasets, vanilla PAC reasoning fails to control the group-wise error, leading to non-zero Error<sub>Gap</sub>. Specifically, as shown in Figure 2, vanilla PAC reasoning has high error rates in some groups, such as the *precalculus* category in MATH-500, the *X-Large*



**Figure 3. More calibration samples enhance efficiency of reasoning.** Experimental results of G-PAC reasoning for different calibration ratios on MATH-500. The red dashed line  $\epsilon$  means the target risk level.

**Table 1. G-PAC performs a smaller error gap than vanilla PAC.** Experimental results of the binary loss function on verifiable datasets ( $\alpha = 0.05$ ). We set  $\epsilon = 0.05$  for MATH-500,  $\epsilon = 0.08$  for ZebraLogic, and  $\epsilon = 0.15$  for GPQA.

Dataset	Metric	Logits-based score		Verbalized score		Router-based score	
		PAC	G-PAC	PAC	G-PAC	PAC	G-PAC
MATH-500	Error (%)	3.08	2.78	3.24	2.57	3.26	2.63
	Error <sub>Gap</sub> (%)	1.02	0.00	0.11	0.00	1.63	0.00
	STP (%) ↑	56.86	43.44	61.14	41.63	59.20	49.30
ZebraLogic	Error (%)	6.04	4.21	4.76	4.01	6.23	4.35
	Error <sub>Gap</sub> (%)	10.30	0.00	3.00	0.00	8.84	0.00
	STP (%) ↑	27.65	10.90	4.97	8.53	34.54	34.32
GPQA	Error (%)	11.24	11.59	10.86	10.82	11.57	11.34
	Error <sub>Gap</sub> (%)	0.30	0.00	0.00	0.00	0.00	0.00
	STP (%) ↑	8.73	13.78	17.07	19.60	34.21	30.54

category in Zebra Logic, and the *biology* category in GPQA. In contrast, both G-PAC and C-PAC achieve zero group-conditional error gap, showing they can control group-wise risk even when group labels are not known beforehand. Although G-PAC and C-PAC may slightly reduce STP because of stricter rules, they still maintain good efficiency in most cases and offer a practical way to balance validity and speed.

**Trade-off between validity and efficiency.** Although G-PAC and C-PAC ensure group-wise error control, partitioning the input space reduces the calibration samples per group, leading to a loss in efficiency (STP). As shown in Table 1, G-PAC often has a lower STP than vanilla PAC while maintaining valid risk control. The results in Figure 3 further reflect this sample loss cost, as a larger calibration set helps recover some of the efficiency.

### 5.3. Additional discussion

**Open-domain task.** We further evaluate our method on an open-domain benchmark, Arena-Hard (Li et al., 2025). Since answers in open-domain settings cannot be automatically verified and Arena-Hard does not provide pre-defined group labels, we adopt the semantic loss in Eq. (15) to quantify performance loss and apply C-PAC reasoning. Table 6 shows that C-PAC reasoning (with 3 groups) consistently

achieves zero Error<sub>Gap</sub>, while the vanilla PAC reasoning fails to control group-conditional risk. Although C-PAC may slightly reduce STP, it still offers significant token savings and remains effective in open-domain settings.

**Ablation experiment.** We conduct ablation experiments to investigate the stability of risk control under different calibration set sizes. Specifically, we vary the calibration ratio and evaluate both PAC and G-PAC reasoning under a binary loss. The results on MATH-500 are presented in Figure 3. Both PAC reasoning and G-PAC reasoning maintain stable marginal error control across different calibration ratios. Meanwhile, G-PAC reasoning consistently achieves a lower group-conditional error gap than standard PAC reasoning, indicating improved group-wise risk control. G-PAC preserves stable efficiency gains as the calibration ratio varies. Results of ZebraLogic and GPQA are shown in Figure 6.

## 6. Conclusion

We proposed G-PAC reasoning, a statistical framework that enables group-conditional risk control for efficient LLM reasoning. By partitioning the input space into groups, G-PAC strengthens marginal PAC guarantees with finer-grained validity at the group level. We establish theoretical guarantees for G-PAC, showing that the group-conditional error gap vanishes for both known and learned groupings. Experiments across reasoning benchmarks demonstrate that G-PAC achieves group-conditional control while maintaining substantial inference-time savings.

**Limitations** First, the efficiency gains of PAC reasoning depend on the quality of the uncertainty score: poorly informative uncertainty estimates may reduce the computational savings. Second, when extending PAC reasoning to unknown groupings, the method requires either sample splitting, which reduces statistical efficiency, or joint clustering, which may introduce coverage gaps.

## Impact Statement

This paper presents work whose goal is to advance the field of Machine Learning. There are many potential societal consequences of our work, none which we feel must be specifically highlighted here.

## References

- Aggarwal, Kim, Lanchantin, Welleck, Weston, Kulikov, and et al. OptimalThinkingBench: Evaluating over and underthinking in LLMs. *Preprint at arXiv:2508.13141*, 2025.
- Angelopoulos, A. N., Bates, S., Candès, E. J., Jordan, M. I., and Lei, L. 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.
- Angelopoulos, A. N., Bates, S., Fisch, A., Lei, L., and Schuster, T. Conformal risk control. *Preprint at arXiv:2208.02814*, 2025b.
- Bates, S., Angelopoulos, A., Lei, L., Malik, J., and Jordan, M. Distribution-free, risk-controlling prediction sets. *Journal of the ACM*, 68(6):1–34, 2021. ISSN 0004-5411, 1557-735X.
- Bentkus, V. On hoeffding’s inequalities. *Annals of Probability*, 32(2):0–26, 2004. ISSN 0091-1798.
- 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.
- Cheng, X., Li, J., Zhao, W. X., and Wen, J.-R. Think more, hallucinate less: Mitigating hallucinations via dual process of fast and slow thinking. *Preprint at arXiv:2501.01306*, 2025.
- Chung, S., Du, W., and Fu, J. 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.
- Dekoninck, J., Baader, M., and Vechev, M. A unified approach to routing and cascading for LLMs. In *Forty-Second International Conference on Machine Learning*, 2025.
- Fang, G., Ma, X., and Wang, X. Thinkless: LLM learns when to think. *Preprint at arXiv:2505.13379*, 2025.
- Feng, T., Zhang, H., Lei, Z., Yue, H., Lin, C., Liu, G., and et al. LLMRouter: An Open-Source Library for LLM Routing, 2025.
- Grattafiori, Dubey, Jauhri, Pandey, Kadian, Al-Dahle, and et al. The Llama 3 herd of models. *Preprint at arXiv:2407.21783*, 2024.
- Hao, Gu, Ma, Hong, Wang, Wang, and et al. Reasoning with language model is planning with world model. *Preprint at arXiv:2305.14992*, 2023.
- Hao, B., Abbasi Yadkori, Y., Wen, Z., and Cheng, G. 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.
- Hoeffding, W. Probability inequalities for sums of bounded random variables. *Journal of the American Statistical Association*, 58(301):13–30, 1963. ISSN 0162-1459.
- Howard, S. R., Ramdas, A., McAuliffe, J., and Sekhon, J. Time-uniform, nonparametric, nonasymptotic confidence sequences. *Annals of Statistics*, 49(2):1055–1080, 2021. ISSN 0090-5364, 2168-8966.
- 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.
- Jiang, G., Quan, G., Ding, Z., Luo, Z., Wang, D., and Hu, Z. FlashThink: An early exit method for efficient reasoning. *Preprint at arXiv:2505.13949*, 2025.

- Joshi, M., Choi, E., Weld, D., and Zettlemoyer, L. TriviaQA: A large scale distantly supervised challenge dataset for reading comprehension. In Barzilay, R. and Kan, M.-Y. (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.
- Li, Chiang, Frick, Dunlap, Wu, Zhu, and et al. From crowd-sourced data to high-quality benchmarks: Arena-hard and benchbuilder pipeline. In *Forty-Second International Conference on Machine Learning*, ICML 2025, Vancouver, BC, Canada, 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.
- Lu, Y. and Zhou, H. H. Statistical and computational guarantees of lloyd’s algorithm and its variants. *Preprint at arXiv:1612.02099*, 2016.
- Ma, W., He, J., Snell, C., Griggs, T., Min, S., and Zaharia, M. Reasoning models can be effective without thinking. *Preprint at arXiv:2504.09858*, 2025.
- 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.
- 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.
- Talmor, A., Herzig, J., Lourie, N., and Berant, J. 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 Bouamor, H., Pino, J., and Bali, K. (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.
- Valiant, L. G. 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.
- von Luxburg, U., Belkin, M., and Bousquet, O. Consistency of spectral clustering. *Annals of Statistics*, 36(2):555–586, 2008. ISSN 0090-5364, 2168-8966.
- Xiao, W. and Gan, L. 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.
- 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.
- Yang, D., Tsai, Y.-H. H., and Yamada, M. On verbalized confidence scores for LLMs. *Preprint at arXiv:2412.14737*, 2024.
- Yong, X., Zhou, X., Zhang, Y., Li, J., Zheng, Y., and Wu, X. 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.
- Zeng, H. A note on the impossibility of conditional PAC-efficient reasoning in large language models. *Preprint at arXiv:2512.03057*, 2025.

Zeng, H., Huang, J., Jing, B., Wei, H., and An, B. PAC reasoning: Controlling the performance loss for efficient reasoning. *Preprint at arXiv:2510.09133*, 2025.

Zhang, Li, Long, Zhang, Lin, Yang, and et al. Qwen3 embedding: Advancing text embedding and reranking through foundation models. *Preprint at arXiv:2506.05176*, 2025a.

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

Zhang, T., Mehradfar, A., Dimitriadis, D., and Avestimehr, S. Leveraging uncertainty estimation for efficient LLM routing. *Preprint at arXiv:2502.11021*, 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. Detailed UCB construction

We provide detailed derivations for constructing the upper confidence bound (UCB) for the performance loss using importance sampling.

**Importance sampling procedure** Let  $\mathcal{I}_{\text{cal},j}$  denote the index set of samples belonging to group  $G_j$  in the calibration dataset. Given a sampling size  $m$ , we uniformly draw  $m$  indices  $\{i_1, \dots, i_m\}$  with replacement from  $\mathcal{I}_{\text{cal},j}$ . For each selected index  $i_t$ , we perform a Bernoulli trial  $\xi_{i_t} \sim \text{Bern}(\pi_{i_t})$  to decide whether to query the expert answer  $y_{i_t}$ , where  $\{\pi_{i_t}\}_{t=1}^m$  are sampling weights. This yields  $m$  i.i.d. random variables:

$$Z_t(u) = \ell(y_{i_t}, \tilde{y}_{i_t}) \frac{\xi_{i_t}}{\pi_{i_t}} \mathbf{1}\{U_{i_t} \leq u\}.$$

The expectation of  $Z_t(u)$  equals the target quantity  $L(u, G_j)$ , since  $\mathbb{E}_{\xi_{i_t}}[\xi_{i_t}/\pi_{i_t} | i_t] = 1$ . Then, we can estimate an upper bound for  $L(u, G_j)$  by computing a confidence interval for the mean of  $\{Z_t(u)\}_{t=1}^m$ .

**CLT-based UCB** For large  $m$ , the sample mean  $\hat{\mu}_Z(u) = \frac{1}{m} \sum_{j=1}^m Z_j(u)$  is approximately normal by the central limit theorem, and the UCB is given by:

$$\hat{L}_{j,u}(\alpha) = \hat{\mu}_Z(u) + z_{1-\alpha} \frac{\hat{\sigma}_Z(u)}{\sqrt{m}},$$

where  $\hat{\sigma}_Z(u)$  is the sample standard deviation and  $z_{1-\alpha}$  is the  $(1 - \alpha)$ -quantile of the standard normal distribution. Then, we construct a valid upper confidence bound on  $R(\hat{f}|G_j)$  at level  $1 - \alpha$ :  $\mathbb{P}(R(\hat{f}|G_j) \leq \hat{L}_{j,u}(\alpha)) \geq 1 - \alpha$ .

**Hoeffding-based UCB** When the loss function is bounded in  $[0, B]$ , we can replace the CLT-based UCB with a finite-sample bound using Hoeffding's inequality. In line 16 of Algorithm 1, instead of computing the CLT-based bound, we first define  $R = B/\pi_{\min}$  where  $\pi_{\min} = \min_i \pi_i$ , then compute:

$$\delta_{\text{HB}}(\alpha) = \sqrt{\frac{R^2 \log(2/\alpha)}{2m}}, \quad \hat{L}_u(\alpha) = \hat{\mu}_Z(u) + \delta_{\text{HB}}(\alpha).$$

This bound provides strict finite-sample guarantees without relying on asymptotic normality, which is useful when the sample size  $m$  is small. The trade-off is that Hoeffding's bound is typically more conservative than the CLT-based bound for large  $m$ .

The CLT-based and Hoeffding-based approaches are two examples of constructing valid UCBs. Alternatively, one could use other concentration inequalities such as Bernstein's inequality (Bentkus, 2004; Hao et al., 2019) to construct a valid confidence bound, which may provide better guarantees under different conditions. Other approaches include betting-based confidence intervals and empirical Bernstein bounds. The choice of UCB construction depends on the sample size, loss boundedness, and desired tightness.

## B. Proof of Theorem 4.2

*Proof.* Fix a group  $G_j \in \mathcal{G}$  and let  $\mathcal{D}_j = \{x_i \in \mathcal{D}_{\text{cal}} : x_i \in G_j\}$  be the calibration data for this group. By Assumption 4.1, the UCB  $\hat{L}_{j,u}(\alpha)$  computed on  $\mathcal{D}_j$  satisfies

$$\mathbb{P}_{\mathcal{D}_j}(R(u | G_j) \leq \hat{L}_{j,u}(\alpha)) \geq 1 - \alpha. \tag{8}$$

The algorithm selects the threshold  $\hat{u}_j = \max\{u : \hat{L}_{j,u}(\alpha) \leq \varepsilon\}$ . By definition,  $\hat{L}_{j,\hat{u}_j}(\alpha) \leq \varepsilon$ . Combining with the UCB validity, with probability at least  $1 - \alpha$ :

$$R(\hat{f} | G_j) = R(\hat{u}_j | G_j) \leq \hat{L}_{j,\hat{u}_j}(\alpha) \leq \varepsilon. \tag{9}$$

If no valid threshold exists (i.e.,  $\hat{L}_{j,u}(\alpha) > \varepsilon$  for all  $u$ ), we set  $\hat{u}_j = -\infty$ , which means  $\hat{f}(x) = f(x)$  for all  $x \in G_j$ . In this case,  $R(\hat{f} | G_j) = 0 \leq \varepsilon$  trivially. Since each group  $G_j$  uses disjoint calibration data  $\mathcal{D}_j$ , the calibration procedures are independent. The guarantee  $\mathbb{P}_{\mathcal{D}_j}(R(\hat{f} | G_j) \leq \varepsilon) \geq 1 - \alpha$  holds for each group separately. No Bonferroni correction is needed because we do not require simultaneous validity across all groups.  $\square$

### C. Proof of Theorem 4.3

*Proof.* The proof combines the population risk guarantee from Theorem 4.2 with Hoeffding's inequality for the test set. Let  $\hat{u}_j$  be the threshold selected by Algorithm 2 for group  $G_j$ , which is determined by the calibration set  $\mathcal{D}_j$ . Conditioned on  $\hat{u}_j$ , the test losses  $\ell(\hat{f}(x_i), f(x_i))$  for  $x_i \in G_j \cap \mathcal{D}_{\text{test}}$  are i.i.d. and bounded in  $[0, B]$ . By Hoeffding's inequality, for any  $t > 0$ :

$$\mathbb{P}\left(\hat{R}(\hat{f} | G_j) - R(\hat{f} | G_j) > t | \hat{u}_j\right) \leq \exp\left(-\frac{2N_j t^2}{B^2}\right). \quad (10)$$

Using the inclusion:

$$\{\hat{R}(\hat{f} | G_j) > \varepsilon + t\} \subseteq \{R(\hat{f} | G_j) > \varepsilon\} \cup \{\hat{R}(\hat{f} | G_j) - R(\hat{f} | G_j) > t\}. \quad (11)$$

Taking probabilities and applying the union bound:

$$\mathbb{P}(\hat{R}(\hat{f} | G_j) > \varepsilon + t) \leq \mathbb{P}(R(\hat{f} | G_j) > \varepsilon) + \mathbb{P}(\hat{R}(\hat{f} | G_j) - R(\hat{f} | G_j) > t) \quad (12)$$

$$\leq \alpha + \exp\left(-\frac{2N_j t^2}{B^2}\right). \quad (13)$$

The first term is bounded by  $\alpha$  from Theorem 4.2. The second term follows from the law of total probability and the conditional Hoeffding bound. Therefore:

$$\mathbb{P}\left(\hat{R}(\hat{f} | G_j) \leq \varepsilon + t\right) \geq 1 - \alpha - \exp\left(-\frac{2N_j t^2}{B^2}\right). \quad (14)$$

□

### D. Proof of Proposition 4.5

We prove that grouping always improves PAC efficiency, with strict improvement when groups have heterogeneous risk functions.

*Proof.* Let  $\mathcal{G} = \{G_1, \dots, G_k\}$  be a partition of  $\mathcal{X}$  with group probabilities  $p_j = \mathbb{P}(x \in G_j)$ . Let  $\hat{u}^*$  denote the marginal PAC-optimal threshold (no grouping), and let  $\hat{u}_j^*$  denote the PAC-optimal threshold for group  $G_j$ . By definition, the marginal threshold  $\hat{u}^*$  satisfies the PAC constraint on the entire population:

$$\mathbb{P}_{\mathcal{D}_{\text{cal}}}(R(\hat{u}^*) \leq \varepsilon) \geq 1 - \alpha.$$

The marginal risk decomposes as  $R(u) = \sum_{j=1}^k p_j \cdot R(u | G_j)$ . Since  $R(\hat{u}^*) \leq \varepsilon$  implies  $\sum_{j=1}^k p_j \cdot R(\hat{u}^* | G_j) \leq \varepsilon$ , the threshold  $\hat{u}^*$  is feasible for the marginal problem. For each group  $G_j$ , the group-conditional calibrated threshold is defined as

$$\hat{u}_j^* = \sup\{u : \mathbb{P}_{\mathcal{D}_j}(R(u | G_j) \leq \varepsilon) \geq 1 - \alpha\}.$$

We claim that  $\hat{u}_j^* \geq \hat{u}^*$  for all  $j \in [k]$ . Note that  $R(\hat{u}^*) \leq \varepsilon$  implies  $\sum_{j=1}^k p_j \cdot R(\hat{u}^* | G_j) \leq \varepsilon$ , but not necessarily  $R(\hat{u}^* | G_j) \leq \varepsilon$  for each  $j$ . Since  $\hat{u}_j^*$  is optimized for  $G_j$  alone while  $\hat{u}^*$  satisfies  $\sum_{j=1}^k p_j \cdot R(\hat{u}^* | G_j) \leq \varepsilon$ , we have:

$$\begin{aligned} R(\hat{u}^* | G_j) < R(\hat{u}^*) &\Rightarrow \hat{u}_j^* > \hat{u}^*, \\ R(\hat{u}^* | G_j) > R(\hat{u}^*) &\Rightarrow \hat{u}_j^* < \hat{u}^*. \end{aligned}$$

The PAC efficiency under grouping is

$$\text{Eff}_{\text{PAC}}(\mathcal{G}) = \sum_{j=1}^k p_j \cdot \mathbb{P}(U(x) \leq \hat{u}_j^* | x \in G_j).$$

The marginal PAC efficiency is

$$\text{Eff}_{\text{PAC}}(\{\mathcal{X}\}) = \mathbb{P}(U(x) \leq \hat{u}^*) = \sum_{j=1}^k p_j \cdot \mathbb{P}(U(x) \leq \hat{u}^* \mid x \in G_j).$$

To show  $\text{Eff}_{\text{PAC}}(\mathcal{G}) \geq \text{Eff}_{\text{PAC}}(\{\mathcal{X}\})$ , note that  $\hat{u}^*$  is feasible for the group-conditional problem. For each  $G_j$ ,  $\hat{u}_j^*$  maximizes  $\mathbb{P}(U(x) \leq u \mid x \in G_j)$  subject to  $\mathbb{P}_{\mathcal{D}_j}(R(u \mid G_j) \leq \varepsilon) \geq 1 - \alpha$ . Thus

$$\mathbb{P}(U(x) \leq \hat{u}_j^* \mid x \in G_j) \geq \mathbb{P}(U(x) \leq \hat{u}^* \mid x \in G_j).$$

Summing over all groups:

$$\text{Eff}_{\text{PAC}}(\mathcal{G}) = \sum_{j=1}^k p_j \cdot \mathbb{P}(U(x) \leq \hat{u}_j^* \mid x \in G_j) \geq \sum_{j=1}^k p_j \cdot \mathbb{P}(U(x) \leq \hat{u}^* \mid x \in G_j) = \text{Eff}_{\text{PAC}}(\{\mathcal{X}\}).$$

Equality holds iff  $\hat{u}_j^* = \hat{u}^*$  for all  $j \in [k]$ , which occurs when  $R(u \mid G_j) = R(u \mid G_{j'})$  for all  $j, j'$  (no heterogeneity). When  $\exists j \neq j' : R(u \mid G_j) \neq R(u \mid G_{j'})$ , we have  $\exists j : \hat{u}_j^* \neq \hat{u}^*$ , yielding strict improvement.  $\square$

## E. Proof of Theorem 4.6

We prove the PAC guarantee for unknown grouping under both sample splitting and joint approaches.

*Proof.* We prove the two parts of the theorem separately.

### 1. Sample splitting approach

The proof has two components: (1) the coverage guarantee, and (2) the efficiency gap bound.

**Coverage guarantee.** Let  $\widehat{\mathcal{G}} = \{\widehat{G}_1, \dots, \widehat{G}_k\}$  be the partition learned from  $\mathcal{D}_{\text{cluster}}$ . Since  $\mathcal{D}_{\text{cluster}}$  and  $\mathcal{D}_{\text{cal}}$  are disjoint, the partition  $\widehat{\mathcal{G}}$  is independent of the calibration data  $\mathcal{D}_{\text{cal}}$ . Conditioned on  $\widehat{\mathcal{G}}$ , the calibration procedure is identical to the known grouping case. For each group  $\widehat{G}_j$ , let  $\mathcal{D}_j = \{x_i \in \mathcal{D}_{\text{cal}} : x_i \in \widehat{G}_j\}$  be the calibration data for this group. By Assumption 4.1, the UCB  $\widehat{L}_{j,u}(\alpha)$  computed on  $\mathcal{D}_j$  satisfies:

$$\mathbb{P}_{\mathcal{D}_j}(R(u \mid \widehat{G}_j) \leq \widehat{L}_{j,u}(\alpha) \mid \widehat{\mathcal{G}}) \geq 1 - \alpha.$$

The algorithm selects  $\hat{u}_j = \max\{u : \widehat{L}_{j,u}(\alpha) \leq \varepsilon\}$ , so  $\widehat{L}_{j,\hat{u}_j}(\alpha) \leq \varepsilon$ . Combining with the UCB validity:

$$\mathbb{P}_{\mathcal{D}_j}(R(\hat{f} \mid \widehat{G}_j) \leq \varepsilon \mid \widehat{\mathcal{G}}) \geq 1 - \alpha.$$

Since this holds for any fixed  $\widehat{\mathcal{G}}$ , by the law of total probability:

$$\mathbb{P}_{\mathcal{D}_{\text{cal}}}(R(\hat{f} \mid \widehat{G}_j) \leq \varepsilon) \geq 1 - \alpha.$$

**Efficiency gap bound.** Let  $\delta = d(\widehat{\mathcal{G}}, \mathcal{G}^*)$  be the partition gap. For any input  $x$ , let  $g(x)$  and  $g^*(x)$  denote the group assignments under  $\widehat{\mathcal{G}}$  and  $\mathcal{G}^*$  respectively. The efficiency difference can be bounded as:

$$\begin{aligned} & \text{Eff}_{\text{PAC}}(\mathcal{G}^*; \varepsilon, \alpha) - \text{Eff}_{\text{PAC}}(\widehat{\mathcal{G}}; \varepsilon, \alpha) \\ &= \mathbb{E}[\mathbf{1}\{U(x) \leq \hat{u}_{g^*(x)}^*\}] - \mathbb{E}[\mathbf{1}\{U(x) \leq \hat{u}_{g(x)}^*\}] \\ &\leq \mathbb{P}(g(x) \neq g^*(x)) + \mathbb{P}(g(x) \neq g^*(x)) \\ &= 2\delta. \end{aligned}$$

The inequality follows because the efficiency difference is at most 1 for misclassified inputs and 0 for correctly classified inputs.

## 2. Joint approach

Let  $\widehat{\mathcal{G}} = \{\widehat{G}_1, \dots, \widehat{G}_k\}$  be the learned partition from calibration set  $\mathcal{D}_{\text{cal}} = \{(x_i, U_i, L_i)\}_{i=1}^n$ . Let  $\mathcal{G}^* = \{G_1^*, \dots, G_k^*\}$  be the oracle optimal partition, and let  $\delta = d(\widehat{\mathcal{G}}, \mathcal{G}^*)$  be the partition gap. We prove that with probability at least  $1 - \alpha$ , all learned groups satisfy  $R(\widehat{f} | \widehat{G}_j) \leq \varepsilon + c \cdot \delta$ .

For each learned group  $\widehat{G}_j$ , the population risk can be decomposed as:

$$R(\widehat{f} | \widehat{G}_j) = \mathbb{E}_{x \sim \mathcal{P}}[\ell(\widehat{f}(x), f(x)) \mid x \in \widehat{G}_j].$$

Let  $\widehat{g} : \mathcal{X} \rightarrow [k]$  and  $g^* : \mathcal{X} \rightarrow [k]$  denote the group assignment functions for  $\widehat{\mathcal{G}}$  and  $\mathcal{G}^*$  respectively. We decompose the population risk based on whether the learned assignment matches the oracle:

$$\begin{aligned} R(\widehat{f} | \widehat{G}_j) &= \mathbb{E}[\ell(\widehat{f}(x), f(x)) \mid \widehat{g}(x) = j] \\ &= \mathbb{E}[\ell(\widehat{f}(x), f(x)) \cdot \mathbf{1}\{\widehat{g}(x) = g^*(x)\} \mid \widehat{g}(x) = j] \\ &\quad + \mathbb{E}[\ell(\widehat{f}(x), f(x)) \cdot \mathbf{1}\{\widehat{g}(x) \neq g^*(x)\} \mid \widehat{g}(x) = j]. \end{aligned}$$

For the misclassification term, since the loss function is bounded by  $B$ , we have:

$$\begin{aligned} \mathbb{E}[\ell(\widehat{f}(x), f(x)) \cdot \mathbf{1}\{\widehat{g}(x) \neq g^*(x)\} \mid \widehat{g}(x) = j] &\leq B \cdot \mathbb{P}(\widehat{g}(x) \neq g^*(x) \mid \widehat{g}(x) = j) \\ &\leq B \cdot \frac{\mathbb{P}(\widehat{g}(x) \neq g^*(x))}{\mathbb{P}(\widehat{g}(x) = j)} = B \cdot \frac{\delta}{p_j}, \end{aligned}$$

where  $p_j = \mathbb{P}(\widehat{g}(x) = j)$  is the probability mass of learned group  $\widehat{G}_j$ .

For inputs with  $\widehat{g}(x) = g^*(x)$ , the learned group assignment matches the oracle. Conditioned on the learned partition  $\widehat{\mathcal{G}}$ , define the empirical risk on calibration data:

$$\widehat{R}_j = \frac{1}{|\mathcal{D}_j|} \sum_{i: x_i \in \widehat{G}_j} L_i \cdot \mathbf{1}\{U_i \leq \widehat{u}_j\},$$

where  $\mathcal{D}_j = \{x_i \in \mathcal{D}_{\text{cal}} : x_i \in \widehat{G}_j\}$ . By Assumption 4.1, the UCB  $\widehat{L}_{j,u}(\alpha)$  satisfies:

$$\mathbb{P}_{\mathcal{D}_j}(R(\widehat{u}_j | \widehat{G}_j, \widehat{g} = g^*) \leq \widehat{L}_{j,\widehat{u}_j}(\alpha) | \widehat{\mathcal{G}}) \geq 1 - \alpha,$$

where  $R(\widehat{u}_j | \widehat{G}_j, \widehat{g} = g^*)$  denotes the risk restricted to correctly classified inputs. Since the algorithm selects  $\widehat{u}_j = \max\{u : \widehat{L}_{j,u}(\alpha) \leq \varepsilon\}$ , we have  $\widehat{L}_{j,\widehat{u}_j}(\alpha) \leq \varepsilon$ . Thus, with probability at least  $1 - \alpha$ :

$$\mathbb{E}[\ell(\widehat{f}(x), f(x)) \cdot \mathbf{1}\{\widehat{g}(x) = g^*(x)\} \mid \widehat{g}(x) = j] \leq \varepsilon.$$

Combining the bounds for both terms, for each group  $\widehat{G}_j$ , with probability at least  $1 - \alpha$ :

$$R(\widehat{f} | \widehat{G}_j) \leq \varepsilon + B \cdot \frac{\delta}{p_j}.$$

To simplify the bound, we note that  $p_j \geq p_{\min}$  where  $p_{\min} = \min_j \mathbb{P}(\widehat{g}(x) = j)$ . For balanced partitions with  $k$  groups,  $p_{\min} \geq 1/(2k)$  with high probability. Setting  $c = 2Bk$ , we obtain for each group  $\widehat{G}_j$ :

$$\mathbb{P}(R(\widehat{f} | \widehat{G}_j) \leq \varepsilon + c \cdot \delta) \geq 1 - \alpha.$$

This guarantee holds for each group independently with its own confidence  $1 - \alpha$ .  $\square$

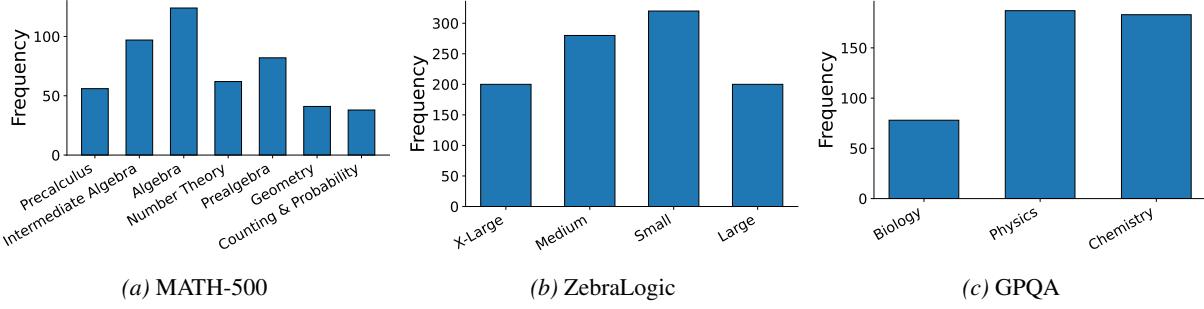


Figure 4. The category distribution for three verifiable benchmarks.

## F. Experimental setup

### F.1. LLMs and datasets

**Large language models** In this study, we evaluate the G-PAC reasoning based on the Qwen3 series models (Yang et al., 2025a) and the Llama-3.1-8B-based LLMs. Specifically, we employ the “Qwen3-4B-Thinking-2507” as the thinking model and “Qwen3-4B-Instruct-2507” as the lower-performance non-thinking model. We also conduct a complementary experiment on Llama-3.1-8B-based models where the “DeepSeek-R1-Distill-Llama-8B” as the thinking model and “Llama-3.1-8B-Instruct” as the lower-performance non-thinking model. The sampling temperature and other hyperparameters for both LLMs are configured following the settings in the original paper. Experiments were run on four NVIDIA RTX A6000 Graphics Cards.

**Datasets** We evaluate PAC reasoning on a range of real-world datasets spanning different reasoning paradigms. Specifically, our evaluation includes a mathematical reasoning benchmark, MATH-500 (Lightman et al., 2023), a text-based logical reasoning dataset, ZebraLogic (Lin et al., 2025), and a commonsense question answering benchmark, GPQA (Rein et al., 2024). For each dataset, we randomly split the original test set into a PAC calibration subset and a held-out PAC test subset. Table 2 summarizes the datasets used in our experiments, along with their corresponding splitting strategies, including dataset type, total size, and the sizes of the calibration and test partitions. We also report the distribution of categories for each dataset in Figure 4.

Table 2. 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 PAC Test	300 200
ZebraLogic	Text reasoning	1000	PAC Calibration PAC Test	500 500
GPQA	QA Task	448	PAC Calibration PAC Test	224 224

**The performance of tested LLMs on different benchmarks** We report the accuracy of the tested LLMs on different benchmarks in Table 3. As shown, the performance gaps between models vary substantially across datasets, reflecting differences in task difficulty. Since such gaps directly influence the tightness of PAC-style guarantees, using a uniform tolerance would be either overly conservative for certain benchmarks. Therefore, we adopt dataset-specific values of  $\varepsilon$  to appropriately account for these varying performance gaps and to ensure meaningful guarantees.

### F.2. Uncertainty score function

In this part, we introduce the details of three uncertainty score functions used to quantify the uncertainty of an answer from the non-thinking model, including a white-box score derived from model logits and two black-box scores obtained from verbalized self-reports and an external router model.

Table 3. Accuracy (%) of different models on three reasoning benchmarks.

Dataset	Qwen LLMs		Llama-based LLMs	
	Qwen3-4B-Instruct-2507	Qwen3-4B-Thinking-2507	Llama-3.1-8B-Instruct	DeepSeek-R1-Distill-Llama-8B
MATH-500	92.80	96.20	45.20	85.80
ZebraLogic	80.40	89.20	12.50	38.40
GPQA	46.43	48.88	27.23	39.73

**Logits-based score** For the logits-based score, we use token-level probabilities computed from the prediction logits (Kwon et al., 2023; Zheng et al., 2024; Zhou et al., 2025). Formally, let  $y_i = (y_{i,1}, \dots, y_{i,l})$  be an answer with  $l$  tokens and  $y_{i,j}$  be the  $j$ -th token of the answer. Furthermore, we define the uncertainty score of  $y_i$  as its average token probability (Hao et al., 2023; Huang et al., 2025):

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

where  $\mathbb{P}(y_{i,j} | y_{i,1}, \dots, y_{i,j-1}, x_i)$  is the conditional probability of token  $y_{i,j}$ .

**Verbalized score** We also consider 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. The corresponding prompts are listed in Table 4. In this study, we compute the average confidence over 10 trials and then define the verbalized uncertainty score as one minus this average confidence.

Table 4. 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.

**Router-based score** We trained a router for each mode pair based on an open-source library, named LLMRouter (Feng et al., 2025). Specifically, we sample 2,000 examples from eight common 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 trained as a lightweight classifier that takes the input prompt as features and predicts a routing score representing the probability that using the thinking model is necessary. At inference time for our PAC reasoning, we define the uncertainty score of the non-thinking model as one minus the routing probability predicted for the non-thinking model. Compared to other uncertainty scores, the router-based score relies on supervised signals collected from paired model outputs and can capture more complex input interactions between different models beyond simple confidence estimation. In this work, we use the “Qwen3-Embedding-8B” (Zhang et al., 2025a) as the embedding model to encode the input prompts for training and inference of the router.

**Remark 5.** Logits-based and verbalized uncertainty scores follow a cascade routing setup, where a lightweight model first generates a candidate answer, and the corresponding uncertainty score is computed based on this output to decide whether to use a stronger model. In contrast, router-based scores determine the routing decision directly from the router’s output, without requiring the non-thinking model to perform explicit answer generation. Both setups are widely used in LLM routing (Zeng et al., 2025; Dekoninck et al., 2025; Zhang et al., 2025c). Notably, router-based routing is often more resource-efficient, as it avoids the cost of generating candidate answers with the non-thinking LLM. By supporting both answer-dependent and direct routing scores, our framework demonstrates its generality and compatibility with common routing strategies in practice.

**Alternative scores** The choice of the uncertainty score is flexible. Beyond the uncertainty scores considered in this work, a reward model (Zhang et al., 2025b) can also be used to quantify the quality of the generated answer, and its output can naturally serve as an alternative uncertainty score.

**Expected Calibration Error of different uncertainty scores** Figure 5 reports the Expected Calibration Error (ECE) of different uncertainty scores across three verifiable benchmarks. We observe substantial variation in calibration quality across datasets and scoring methods: logits-based scores are generally better calibrated than verbalized scores, while router-based scores often exhibit the largest ECE, especially on more challenging benchmarks. Despite these differences, PAC reasoning consistently maintains valid control of performance loss across all settings (See Section 5). This result highlights a key property of our framework: PAC reasoning does not rely on well-calibrated uncertainty estimates. **These findings demonstrate that PAC reasoning is model-agnostic, dataset-agnostic, and score-agnostic.**

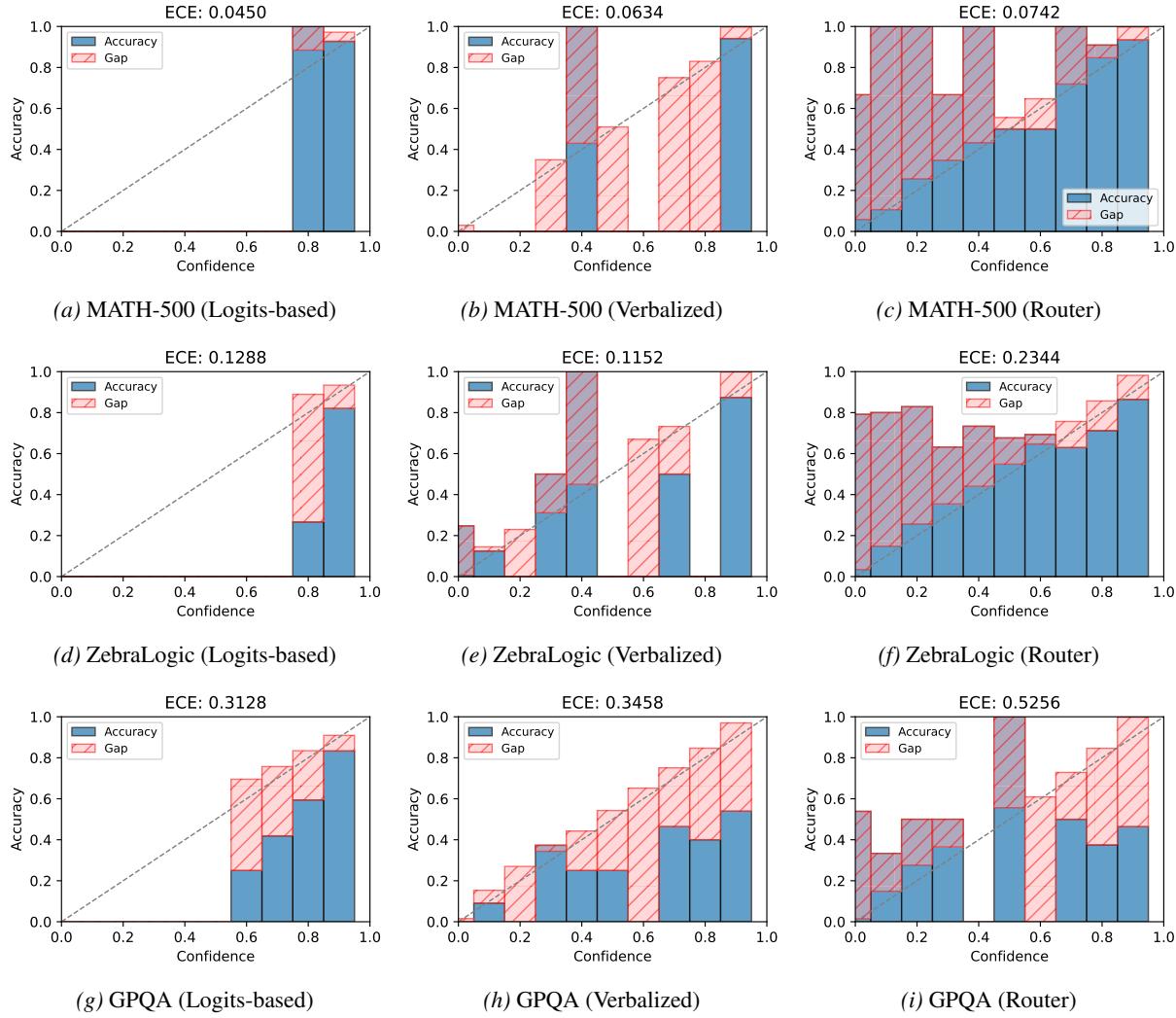


Figure 5. Expected Calibration Error across three verifiable benchmarks using Qwen LLMs.

### F.3. Evaluation metrics

Let  $T$  denote the number of independent trials, indexed by  $t = 1, \dots, T$ . To evaluate the effectiveness of our method, we consider the following metrics.

**Risk evaluation** For the  $t$ -th trial, the empirical error is defined as

$$\text{Error}^{(t)} = \frac{1}{N} \sum_{j=1}^k \sum_{i \in \mathcal{I}_{\text{test},j}^{(t)}} \ell(y_i, \tilde{y}_i) \mathbf{1}\{U(x_i) \leq u\}.$$

We first average the group-wise empirical risk across trials. For each group  $j$ , define

$$\bar{E}_j := \frac{1}{T} \sum_{t=1}^T E_j^{(t)}, \quad E_j^{(t)} = \frac{1}{N_j^{(t)}} \sum_{i \in \mathcal{I}_{\text{test},j}^{(t)}} \ell(y_i, \tilde{y}_i) \mathbf{1}\{U(x_i) \leq u\}.$$

We report the final metrics by averaging over all trials:

$$\text{Error} = \frac{1}{T} \sum_{t=1}^T \text{Error}^{(t)}, \quad \text{Error}_{\text{Gap}} = \sum_{j=1}^k (\bar{E}_j - \varepsilon) \mathbf{1}\{\bar{E}_j > \varepsilon\}.$$

When  $\text{Error}_{\text{Gap}} = 0$ , the group-conditional empirical risk does not exceed the tolerance level  $\varepsilon$  for any group in any trial.

**Efficiency evaluation** To evaluate the effectiveness of PAC reasoning in reducing inference cost, we introduce an efficiency metric termed *Saved Token Percentage* (STP). Let  $l_{\tilde{y}_i}$  denote the number of tokens generated by the non-thinking model for its candidate answer  $\tilde{y}_i$ , and let  $l_{y_i}$  denote the number of tokens generated by the thinking model for its reference answer  $y_i$ . The STP is formally defined as

$$\text{STP} := \frac{1}{N} \sum_{i \in \mathcal{I}_{\text{test}}} \left( 1 - \frac{l_{\tilde{y}_i} + \mathbf{1}\{U_i > u\} l_{y_i}}{l_{y_i}} \right) \times 100\%.$$

Here,  $\mathbf{1}\{\cdot\}$  is the indicator function,  $U_i$  denotes the uncertainty score, and  $u$  is the calibrated threshold. This metric measures the relative token savings by comparing the tokens consumed by PAC reasoning against always invoking the thinking model, while accounting for cases in which an expert (thinking-model) call is triggered.

In the router-based setting, however, the uncertainty score  $U_i$  is produced by an external router rather than by the non-thinking model itself. As a result, the non-thinking model is only executed when the router decides not to use the thinking model. Accordingly, we slightly revise the STP definition as follows:

$$\text{STP} := \frac{1}{N} \sum_{i \in \mathcal{I}_{\text{test}}} \left( 1 - \frac{\mathbf{1}\{U_i \leq u\} l_{\tilde{y}_i} + \mathbf{1}\{U_i > u\} l_{y_i}}{l_{y_i}} \right) \times 100\%.$$

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  $\varepsilon$ , and set the sampling weight  $\pi = \pi_i = 0.5$  for each  $i \in \mathcal{I}_{\text{cal}}$  and the sample size  $m = n \times \frac{1}{\pi}$ .

#### F.4. Loss function

Since the motivation is that PAC reasoning is designed to control the additional error introduced by switching from a reliable but expensive thinking model to a cheaper non-thinking model, this loss is intentionally defined relative to the thinking model rather than directly with respect to the ground-truth accuracy. In this work, we consider two types of loss functions for evaluating the PAC guarantee of our method: the semantic cosine distance and the binary 0-1 loss.

**Semantic loss function** The semantic cosine distance measures similarity between outputs in the embedding space. Formally, given reference output  $y_i = f(x_i)$  and PAC reasoning output  $\hat{y}_i = \hat{f}(x_i)$ , we compute their embeddings  $v_{y_i}$  and  $v_{\hat{y}_i}$ , and define the loss as:

$$\ell(y_i, \hat{y}_i) = 1 - \frac{v_{y_i} \cdot v_{\hat{y}_i}}{\|v_{y_i}\| \|v_{\hat{y}_i}\|}. \quad (15)$$

For the semantic embedding model, we adopt “Qwen3-Embedding-4B” (Zhang et al., 2025a).

**Binary loss function** We also employ a binary 0–1 loss, defined in Eq. (16), which captures the actual loss in answer accuracy when comparing the PAC reasoning output  $\hat{y}$  with the reference output  $y$ :

$$\ell(y_i, \hat{y}_i) = \ell(y_i, \hat{y}_i | y_{i,gold}) = \mathbf{1}\{\hat{y}_i \neq y_i^{gold}\} \mathbf{1}\{y_i = y_i^{gold}\} \quad (16)$$

where  $y_{i,gold}$  is the ground-truth answer for the problem  $x_i$ .

**Remark 6.** By counting an error only when the thinking model produces a correct answer but the PAC reasoning output does not, this loss isolates the incremental performance degradation caused by efficiency-oriented routing. This formulation is particularly suitable for switching-based deployment scenarios, where the thinking model serves as a reliability anchor, and the primary objective is to ensure that efficiency gains do not incur excessive additional errors. In this sense, PAC reasoning provides a statistically valid guarantee on relative performance loss with respect to a fixed thinking model, rather than a guarantee of absolute correctness with respect to ground truth.

## G. Main experimental results

### G.1. Main results of Qwen series models

In this part, we present the results of the Qwen series LLMs.

**Error analysis across categories for vanilla PAC reasoning and G-PAC reasoning.** To examine the validity of group-conditional guarantees, we report error bars for both the overall population and each category under the Qwen series models, as shown in Figure 2. The corresponding results have been summarized in Table 1. While vanilla PAC reasoning successfully controls the overall error below the target tolerance  $\varepsilon$ , it fails to consistently satisfy the error constraint at the category level, with several categories exhibiting errors that exceed  $\varepsilon$ . In contrast, G-PAC reasoning maintains valid error control not only for the overall dataset but also uniformly across all categories, demonstrating its effectiveness in enforcing group-conditional guarantees.

**Experiment of C-PAC under known group partitions.** Table 5 presents the experimental results comparing vanilla PAC and C-PAC under the setting where the group partition is known. We evaluate both methods on three verifiable reasoning benchmarks (MATH-500, ZebraLogic, and GPQA) with  $\alpha = 0.05$ . Across all scoring strategies and under both disjoint and joint clustering, C-PAC consistently achieves lower empirical error than PAC. More importantly, C-PAC effectively drives the group-wise error gap  $\text{Error}_{\text{Gap}}$  to zero in all cases. These results demonstrate that incorporating group-aware calibration enables C-PAC to provide stronger and more reliable guarantees across heterogeneous groups.

**Ablation analysis on the calibration ratios.** Figure 6 presents additional results on ZebraLogic and GPQA. Across different calibration ratios, both PAC and G-PAC maintain stable marginal error control. However, the group-conditional error gap of vanilla PAC varies noticeably with the calibration ratio, while G-PAC consistently achieves near-zero error gaps across all uncertainty scores. Moreover, the efficiency of G-PAC changes smoothly with the calibration ratio and remains comparable to that of PAC. These results further demonstrate the robustness of G-PAC to the choice of calibration set size.

### G.2. Main results of Llama-based LLMs

In this part, we report results for the Llama model. Table 7 shows that, when group partitions are known, G-PAC consistently eliminates the group-conditional error gap across all datasets and uncertainty scores on Llama-based models, while retaining comparable computational savings to PAC. Moreover, Figure 7 shows that the average risk for each category when using Llama-based LLMS. Table 8 demonstrates that under unknown group partitions, C-PAC maintains zero empirical  $\text{Error}_{\text{Gap}}$  under both independent and joint clustering, whereas vanilla PAC exhibits substantial conditional risk violations. Table 9 further confirms that on the open-domain Arena-Hard benchmark, C-PAC achieves strict conditional loss control with meaningful efficiency gains, even in the absence of verifiable answers and predefined group labels.

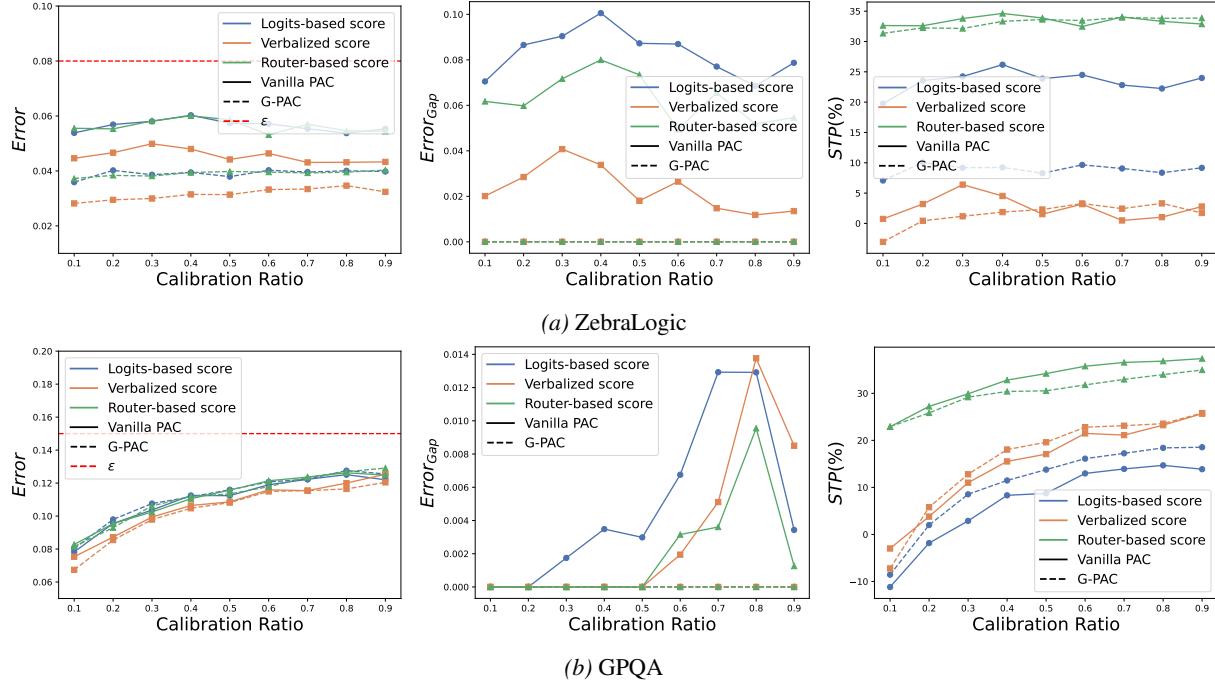


Figure 6. Error control and STP of PAC reasoning for binary loss for different calibration ratios at a confidence level of  $\alpha = 0.05$ . All experiments are conducted on Qwen LLMs. The red dashed line  $\varepsilon$  means the target risk level.

Table 5. Performance comparison between vanilla PAC and G-PAC under known group partitions. Experimental results of the binary loss function on verifiable datasets using the Qwen models with  $\alpha = 0.05$ . We set  $\varepsilon = 0.05$  for MATH-500,  $\varepsilon = 0.1$  for ZebraLogic, and  $\varepsilon = 0.15$  for GPQA.

Dataset	Metric	Disjoint clustering						Joint clustering					
		Logits-based score		Verbalized score		Router-based score		Logits-based score		Verbalized score		Router-based score	
		PAC	C-PAC	PAC	C-PAC	PAC	C-PAC	PAC	C-PAC	PAC	C-PAC	PAC	C-PAC
MATH-500	Error (%)	3.03	2.56	2.70	2.25	2.94	2.88	3.34	2.45	3.24	2.44	3.34	2.92
	Error <sub>Gap</sub> (%)	0.17	0.00	0.00	0.00	0.00	0.00	1.02	0.00	0.33	0.00	0.14	0.00
	STP (%) ↑	50.95	43.19	47.47	35.01	54.10	53.04	58.34	45.79	60.48	42.73	59.29	54.51
ZebraLogic	Error (%)	7.69	5.41	6.67	5.99	7.36	6.77	7.51	5.45	6.67	5.97	7.77	6.74
	Error <sub>Gap</sub> (%)	1.98	0.00	0.00	0.00	1.49	0.00	1.44	0.00	0.00	0.00	2.20	0.00
	STP (%) ↑	36.07	18.99	20.68	10.87	39.01	35.38	35.72	18.95	21.06	10.55	40.34	35.34
GPQA	Error (%)	11.57	11.21	11.24	9.46	11.79	11.36	11.27	11.57	11.00	8.87	11.91	11.60
	Error <sub>Gap</sub> (%)	0.00	0.00	0.06	0.00	1.68	0.00	0.38	0.00	0.00	0.00	1.36	0.00
	STP (%) ↑	11.28	10.32	18.86	5.98	34.64	30.39	9.05	10.87	18.21	0.54	34.91	29.75

**Table 6. C-PAC successfully controls the group-conditional risk on an open-domain task.** Experimental results on Arena-Hard with  $\varepsilon = 0.1$ , using Qwen LLMs.

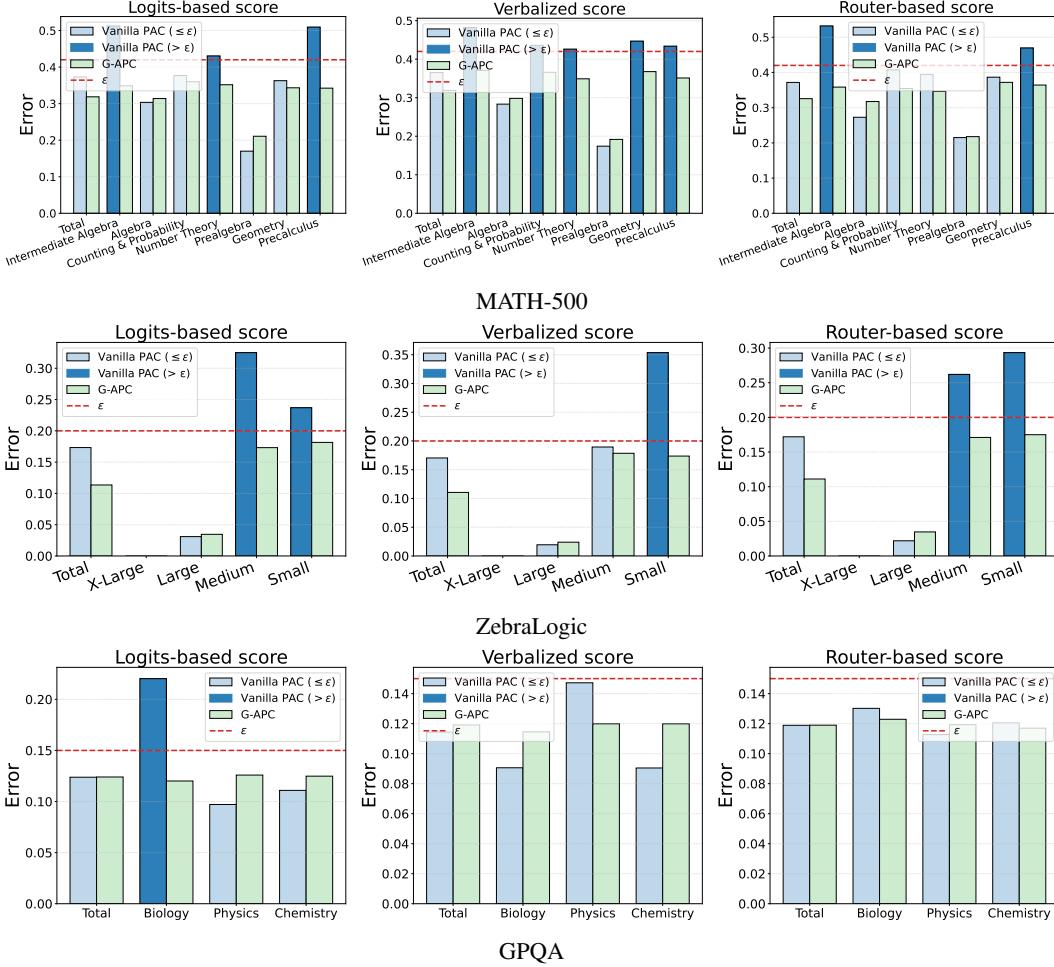
Uncertainty	Method	Error (%)	Error <sub>Gap</sub> (%)	STP (%) ↑
Disjoint clustering				
Logits	PAC	8.84	0.00	51.77
	C-PAC	8.54	0.00	47.01
Verbalized	PAC	8.63	3.68	30.07
	C-PAC	5.27	0.00	10.91
Router	PAC	8.90	4.44	33.07
	C-PAC	8.84	0.00	42.34
Joint clustering				
Logits	PAC	8.94	0.00	52.25
	C-PAC	8.69	0.00	49.32
Verbalized	PAC	8.67	4.04	30.15
	C-PAC	5.17	0.00	10.30
Router	PAC	8.94	4.69	33.39
	C-PAC	8.77	0.00	42.02

**Table 7. G-PAC performs a smaller error gap than vanilla PAC.** Experimental results on verifiable datasets, using Llama-based LLMs ( $\alpha = 0.05$ ). We set  $\varepsilon = 0.42$  for MATH-500,  $\varepsilon = 0.2$  for ZebraLogic, and  $\varepsilon = 0.2$  for GPQA.

Dataset	Metric	Logits-based score		Verbalized score		Router-based score	
		PAC	G-PAC	PAC	G-PAC	PAC	G-PAC
MATH-500	Error (%)	37.28	32.16	36.52	32.18	37.17	32.99
	Error <sub>Gap</sub> (%)	19.27	0.00	12.47	0.00	16.20	0.00
	STP (%) ↑	12.55	2.91	11.32	-2.21	26.01	23.04
ZebraLogic	Error (%)	17.33	11.43	17.05	11.13	17.20	11.13
	Error <sub>Gap</sub> (%)	16.20	0.00	15.37	0.00	15.55	0.00
	STP (%) ↑	23.72	13.23	-30.68	-9.52	25.22	25.98
GPQA	Error (%)	12.38	12.40	11.44	11.91	11.89	11.90
	Error <sub>Gap</sub> (%)	7.03	0.00	0.00	0.00	0.00	0.00
	STP (%) ↑	18.13	23.47	42.52	36.87	39.67	39.88

**Table 8. Performance comparison between vanilla PAC and G-PAC under known group partitions.** Experimental results of the binary loss function on verifiable datasets using the Qwen models with  $\alpha = 0.05$ . We set  $\varepsilon = 0.4$  for MATH-500,  $\varepsilon = 0.2$  for ZebraLogic, and  $\varepsilon = 0.15$  for GPQA.

Dataset	Metric	Disjoint clustering						Joint clustering					
		Logits-based score		Verbalized score		Router-based score		Logits-based score		Verbalized score		Router-based score	
		PAC	C-PAC	PAC	C-PAC	PAC	C-PAC	PAC	C-PAC	PAC	C-PAC	PAC	C-PAC
MATH-500	Error (%)	33.22	33.78	33.38	31.64	33.54	34.41	35.08	34.06	34.86	32.37	35.34	34.83
	Error <sub>Gap</sub> (%)	8.22	0.00	7.68	0.00	2.16	0.00	8.36	0.00	8.90	0.00	4.07	0.00
	STP (%) ↑	3.29	8.40	3.51	0.40	24.28	22.63	7.99	8.96	7.91	0.96	26.47	23.59
ZebraLogic	Error (%)	16.53	15.21	16.26	12.32	16.05	16.38	16.82	15.88	16.15	12.27	16.24	16.57
	Error <sub>Gap</sub> (%)	3.10	0.00	13.62	0.00	3.74	0.00	3.70	0.00	13.43	0.00	4.20	0.00
	STP (%) ↑	21.81	13.31	-30.62	-31.89	23.33	22.65	22.75	17.03	-32.76	-32.38	23.75	23.23
GPQA	Error (%)	11.96	11.98	12.14	12.11	12.29	11.84	12.65	11.92	12.02	11.77	12.34	11.92
	Error <sub>Gap</sub> (%)	10.75	0.00	5.66	0.00	10.47	0.00	12.32	0.00	5.53	0.00	11.05	0.00
	STP (%) ↑	16.75	32.73	38.86	34.85	40.77	37.83	18.89	31.53	38.51	31.36	41.27	37.19



**Figure 7. G-PAC controls the group-conditional performance loss below the target while vanilla PAC fails (dark blue) across three different uncertainty scores.** Experiments are conducted with the Llama-based model. The figure reports the overall and per-category performance losses on three reasoning benchmarks.

**Table 9. C-PAC successfully controls the group-conditional risk on an open-domain task.** Experimental results on Arena-Hard with  $\epsilon = 0.15$ , using Llama-based LLMs.

Uncertainty	Method	Error (%)	Error <sub>Gap</sub> (%)	STP (%) ↑
Disjoint clustering				
Logits	PAC	13.51	3.88	42.70
	C-PAC	13.08	0.00	36.63
Verbalized	PAC	13.34	6.35	29.49
	C-PAC	12.61	0.00	22.50
Router	PAC	13.16	8.67	13.00
	C-PAC	13.18	0.00	36.32
Joint clustering				
Logits	PAC	13.46	3.91	42.62
	C-PAC	13.32	0.00	38.05
Verbalized	PAC	13.23	7.10	28.95
	C-PAC	12.70	0.00	23.06
Router	PAC	13.30	9.38	13.49
	C-PAC	13.20	0.00	35.96