Anytime Safe PAC Efficient Reasoning

Chengyao Yu

Southern University of Science and Technology

Outline

Background: Overthinking of Large Reasoning Models

Large reasoning models (LRMs) have exhibited remarkable capabilities by generating long of chains (CoTs).

Overthinking

Generate excessively long reasoning chains even for simple questions.

- Substantial computational overhead; high latency.

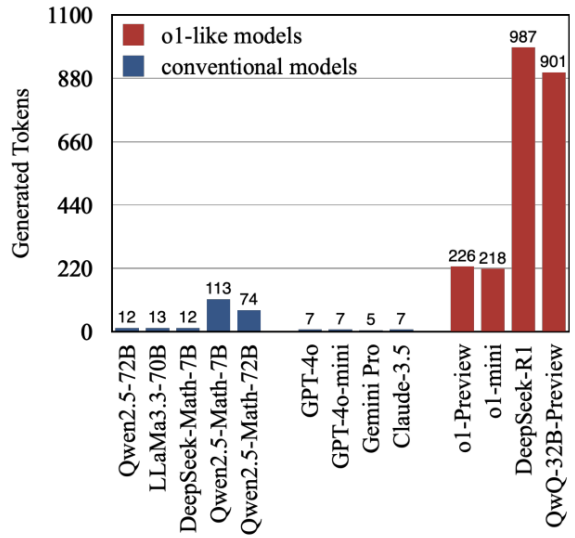


Figure 1.1: Generated tokens on question "what is the answer of 2 plus 3?"

Selective Thinking

When to use a LRM and when not? Cost-accuracy tradeoff.

Selective Thinking

Switching LRMs between non-thinking and thinking modes **based on the difficulty of queries.**

Unfortunately, existing approaches are

- Heuristic-driven: route complex queries to the non-thinking model, resulting in significant performance degradation.
- Targeted for i.i.d. offline settings and need calibration dataset.

Goal of This Paper

Developing an efficient reasoning method with anytime-valid safety guarantees on performance loss relative to the thinking model.

Challenges for Anytime Safe Efficient Reasoning

Challenge 1: No Calibration Data

For an online setting where queries arrive sequentially, the access to a calibration dataset can be impractical.

Challenge 2: Partial Feedback

Performance loss is only observed when the thinking and non-thinking model are both invoked.

Challenge 3: Non-stationary Data

The difficulty of query X_t changes over time t.

Workflow of B-PAC: A Brief Look

We propose *Betting Probably Approximately Correct* (B-PAC) *reasoning*, a method for efficient **online reasoning under partial feedback**, without any access to the offline calibration dataset. The first safe, model-agnostic, efficient reasoning method in online and non-stationary settings.

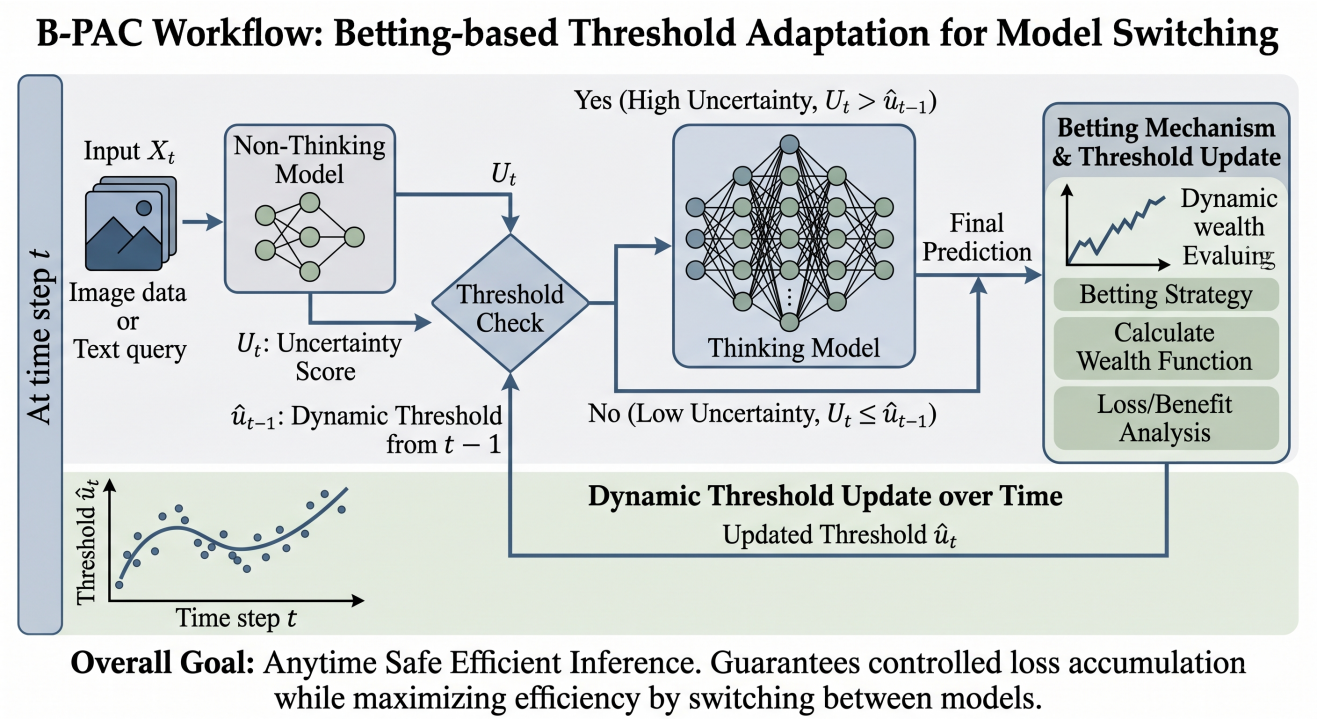


Figure 1.2: Workflow of B-PAC reasoning

More Applications of B-PAC

Theoretical properties established in this paper, such as the **anytime safety**, are **model-agnostic**. Therefore,

B-PAC is highly general and directly applies to any routing system with a cost-accuracy trade-off.

Multi-Agent Expert Routing

Standard queries are handled by generalist agents, while high-risk or complex requests are dynamically routed to specialized expert agents or human operators to ensure system reliability.

Edge-Cloud Collaboration in Autonomous Systems

Autonomous systems use low-latency edge models for routine tasks, dynamically routing to high-precision cloud models when environmental uncertainty exceeds safety thresholds.

Outline

Problem Setup

- Query space: $\mathcal{X}$; Response space: $\mathcal{Y}$. Test samples (X_t, Y_t) arrives sequentially for $t = 1, 2, \ldots$, where $(Y_t)_{t\geq 1}$ are **unobservable.**
- Thinking model $f : \mathcal{X} \to \mathcal{Y}$; Non-thinking model $\tilde{f} : \mathcal{X} \to \mathcal{Y}$.
- Error tolerance level: $\epsilon > 0$; confidence level $1 - \alpha \in (0, 1)$.

When X_t arrives, we construct $\hat{f}_t(X_t) \in \{f(X_t), \tilde{f}(X_t)\}$, which determines whether the non-thinking model output $\tilde{f}(X_t)$ can be taken as the final answer of X_t to reduce inference costs.

Goal: Composite models $(\hat{f}_t)_{t\geq 1}$ should control the performance loss (w.r.t. $(f_t)_{t\geq 1}$) below the level ϵ at any time t with probability at least $1 - \alpha$, while improving efficiency as much as possible.

Anytime (ϵ, α)-PAC efficiency

Definition: Anytime (ϵ, α)-PAC efficiency

A sequence of models $(\hat{f}_t)_{t\geq 0}$ is said to be anytime (ϵ, α)-PAC efficient with respect to a loss $l \in [0, 1]$ if, for any given $\epsilon > 0$ and $\alpha \in (0, 1)$,

$$\mathbb{P}(\forall t \geq 1 : R_t(\hat{f}_t) \leq \epsilon) \geq 1 - \alpha,$$

where $R_t(\hat{f}_t) = \mathbb{E}_{X\sim P_X}[l(\hat{f}_t(X), f(X))]$ is the risk function at time t, measuring the performance loss with respect to the thinking model, $l(\cdot, \cdot)$ is a loss function, and $X \sim P_X$. The probability is taken over the randomness of the streaming data and internal randomization of the procedure that generates $(\hat{f}_t)_{t\geq 1}$.

Outline

1. Uncertainty-based Routing Mechanism

We first produce $\tilde{f}(X_t)$ and compute its uncertainty score $U_t \in [0, 1]$. In line with intuition, U_t should be positively correlated with the likelihood of inconsistency with f.

- If U_t is small, B-PAC reasoning tends to take $\tilde{f}(X_t)$ as a proxy of $f(X_t)$.
- For $\tilde{f}(X_t)$ with large U_t, B-PAC reasoning invokes the thinking LRM to produce $f(X_t)$.

Formally, B-PAC calls thinking model with probability of π_t, where

$$\pi_t = \pi(U_t; \hat{u}_{t-1}, \rho_t) = \mathbb{I}\{U_t \geq \hat{u}_{t-1}\} + \rho_t \mathbb{I}\{U_t < \hat{u}_{t-1}\},$$

where $\rho_t \in (0, 1)$ is a minimum exploration probability at time t.
Therefore, the composite model is written by

$$\hat{f}_t(X_t) = (1 - \xi_t)\tilde{f}(X_t) + \xi_t f(X_t), \tag{3.1}$$

where $\xi_t \sim \text{Bernoulli}(\pi_t)$.

1. Uncertainty-based Routing Mechanism

Remark (Uncertainty Score)

Practitioners can utilize any existing uncertainty scores since the **safety** of B-PAC **holds for any types of scores**. But to gain reasoning acceleration, scores U_t correlated with the likelihood of disagreement with the $f(X_t)$ are preferred.

- **(Open source model)** Logit-based scores such as Perplexity (PPL) and Entropy.
- **(Closed source model)** Verbalized score.
- Score based on another cheap model (not discussed but allowed).

2. Threshold Selection as a Betting Process

Step 1: IPS estimator. Denote the loss of the non-thinking model by $l_t = l(\tilde{f}(X_t), f(X_t))$, which is observable only if the thinking model is utilized (i.e., $\xi_t = 1$). Note that l_t is **not** the realized loss $l(\hat{f}_t(X_t), f(X_t))$ of B-PAC reasoning.

To estimate the risk given partial feedback, we construct the inverse propensity score (IPS) by

$$Z_t(u) = (1 - \rho_{\min})\frac{l_t}{\pi_t}\xi_t \mathbb{I}\{U_t < u\},$$

where $\rho_{\min} = \inf_{t \geq 1} \rho_t$. Here, the term ξ_t/π_t corrects the selection bias inherent in the partial feedback while the coefficient $(1 - \rho_{\min})$ accounts for the time-varying ρ_t, jointly ensuring that $Z_t(u)$ serves as **a tight upper bound of the true risk under threshold** u.

2. Threshold Selection as a Betting Process

Step 2: Supermartingale. Denote the filtration $\mathcal{F}_t$ by the σ-algebra generated by observations up to time t, i.e.,

$$\mathcal{F}_t = \sigma(\{(X_i, \tilde{f}(X_i), l_i\xi_i, U_i)\}_{i=1}^t). \tag{3.2}$$

Let $D_t(u) = \epsilon - Z_t(u)$ and $K_0 = 1$. We construct the process by

$$K_t(u) = K_{t-1}(u)(1 + \lambda_t(u)D_t(u)), \tag{3.3}$$

where $\lambda_t(u) \in \mathcal{F}_{t-1}$ is a non-negative random variable satisfying $1 + \lambda_t(u)D_t(u) \geq 0$ for any $u \in (0, 1)$. It can be proved that $(K_t(u))_{t\geq 0}$ is a **nonnegative supermartingale**.

2. Threshold Selection as a Betting Process

Step3: Threshold selection. Denote the search space of threshold by $\mathcal{U} = \{u^{(1)}, \ldots, u^{(N)}\}$, where $0 = u^{(1)} < u^{(2)} < \cdots < u^{(N)} = 1$. B-PAC reasoning determines the threshold $\hat{u}_t$ by

$$\hat{u}_t = \max \left\{ u^{(i)} \in \mathcal{U} : \forall j \leq i, K_t(u^{(j)}) \geq \frac{1}{\alpha} \right\}, \tag{3.4}$$

where the maximum is to obtain a most efficient model. If $\{u^{(i)} \in \mathcal{U} : \forall j \leq i, K_t(u^{(j)}) \geq 1/\alpha\} = \emptyset$, we set $\hat{u}_t = 0$.

Betting Explanation

- $K_0 = 1$: **initial capital**.
- $K_t(u)$: the **accumulated capital** of a gambler testing the null hypothesis that threshold u is unsafe.
- $\lambda_t \in \mathcal{F}_{t-1}$: the **wager** (bet ratio) determined prior to round t.
- $D_t(u)$: the **payoff of each unit**, comprising the risk tolerance ϵ against the estimated loss $Z_t(u)$.

Therefore,

$$K_t(u) = K_{t-1}(u) + (\lambda_t(u)K_{t-1})D_t(u) = K_{t-1}(1 + \lambda_t(u)D_t(u)).$$

Play a game:

Bet against "threshold u is unsafe".

Betting Explanation

- Under the null hypothesis that threshold u is unsafe,

  $\{K_t(u)\}_{t\geq 0}$ **forms a nonnegative supermartingale.**

Therefore, if u is indeed unsafe, then

the expected wealth decreases over time: $\mathbb{E}[K_t(u)|\mathcal{F}_{t-1}] \leq K_{t-1}(u)$.

Conversely, if u is truly safe, the capital is expected to grow.

So a **large value** of $K_t(u)$ serves as **strong evidence of safety**.

3. Hyperparameter Selection: λ_t

Advance notice: The choices of $(\rho_t)_{t\geq 0}$ and $(\lambda_t)_{t\geq 1}$ do not influence the anytime (ϵ, α)-PAC efficient guarantee, but can have an impact on the reasoning efficiency.

Goal: Maximum efficiency!

Maximum efficiency $\rightarrow$ maximize (log-)wealth $\log K_t(u)$. By $\lambda_t \geq 0$ and $1 + \lambda_t(u) D_t(u) \geq 0$,

$$\lambda_t \in [0, 1/((1 - \rho_{\mathsf{min}})/\rho_t - \epsilon)].$$

To avoid the worst-case scenario $K_t = 0$, we consider $\lambda_t \in [0, c/((1 - \rho_{\mathsf{min}})/\rho_t - \epsilon)]$, where $c \in (0, 1)$ (usually c is close to 1).

3. Hyperparameter Selection: λ_t

Note that

$$\frac{d}{d\lambda}\log K_t(u) \approx \sum_{i=1}^{t-1}(D_i(u) - D_i^2(u)\lambda_i(t)).$$

By optimization theorem, we choose

$$\lambda_t(u) = \min\left\{\max\left\{\frac{\sum_{i=0}^{t-1} D_i(u)}{\sum_{i=0}^{t-1} D_i^2(u) + 1}, 0\right\}, \frac{c}{M_t}\right\}, \tag{3.5}$$

where $M_t = \max\{\epsilon, (1-\rho_{\mathsf{min}})/\rho_t - \epsilon\}$.

3. Hyperparameter Selection: ρ_t

- **A small** ρ_t leads to a large value of $Z_t(u)$ if $\xi_t \mathbb{I}\{U_t < u\} = 1$, which forces a **conservative betting strategy** $(\lambda_t \leq c/(1/\rho_t - \epsilon) \approx 0)$ to maintain non-negative wealth. **This limits the wealth growth rate even when the threshold is safe**.
- **A large** ρ_t permits an **aggressive betting strategy**, which accelerates the identification of optimal thresholds during the initial phase.
- When $\hat{u}_t$ stabilizes, **a large** ρ_t will lead to a **low efficiency** since we call the thinking model with at least ρ_t probability.

Two-stage strategy: $\rho_t = \rho_{\text{warm}} \mathbb{I}\{t \leq T_{\text{warm}}\} + \rho_{\text{deploy}} \mathbb{I}\{t > T_{\text{warm}}\}$,

where ρ_{warm} and ρ_{deploy} take suitable large and small values, respectively.

Engineering Considerations

(1) Negligible Latency Overhead. 0.046s/1000 requests (total systematic computational cost).

(2) Asynchronous State Management. In high-concurrency scenarios, strict synchronization of the wealth process K_t across all incoming requests may introduce lock contention. To address this, B-PAC reasoning can be implemented in an asynchronous manner. The routing decision can read the current threshold snapshot from a shared cache, while the wealth update runs in a background thread or a separate microservice upon receiving feedback.

(3) Scalability via Distributed Betting. For large-scale services distributed across multiple clusters, maintaining a single global wealth process might be challenging. A practical engineering solution is to maintain sharded wealth processes, where different traffic segments (e.g., grouped by user tiers or domain topics) maintain their independent betting games.

Outline

Theory 1: Safety

For brevity, we re-parameterize the risk $R_t(\hat{f}_t)$ as $R_t(\hat{u}_{t-1})$ with a slight abuse of notation. We first focus on the i.i.d. setting.

Theorem 1: Safety

Let $(K_t(u))_{t\geq 0}$, $\hat{u}_t$, and $(\rho_t)_{t\geq 0}$ be defined as before. For any $\alpha \in (0,1)$, $\epsilon \in (0,1)$, and any nonnegative $\lambda_t \in \mathcal{F}_{t-1}$ with $1+\lambda_t D_t(u) \geq 0$, B-PAC reasoning satisfies that

$$\mathbb{P}(\forall t \in \mathbb{N} : R_t(\hat{u}_t) \leq \epsilon) \geq 1-\alpha.$$

Theory 2: Efficiency

Objective: maximizing $\log K_T(u) = \sum_{t=1}^{T} \log(1 + \lambda D_t(u))$.
To achieve a computationally efficient strategy with *closed-form updates*, we maximize *a quadratic surrogate*. Define

$$\lambda_T^*(u) := \arg \max_{0 \le \lambda \le c/((1-\rho_{\min})/\rho_T - \epsilon)} \sum_{t=1}^{T} g_t(\lambda; u),$$

where $g_t(\lambda; u) := \lambda D_t(u) - \frac{1}{2}\lambda^2 D_t^2(u)$ (since $\log(1+x) \approx x - x^2/2$).

Theorem 2: Efficiency

Let $\lambda_t(u)$, ρ_t, and λ_T^* be defined as before. Define the regret of $(\lambda_t(u))_{t \ge 0}$ by $\mathcal{R}_T^{\text{quad}} = \sum_{t=1}^{T} (g_t(\lambda_T^*(u)) - g_t(\lambda_t(u)))$. For $T > T_{\text{warm}}$, we have

$$\mathcal{R}_T^{\text{quad}} \le \frac{c^2}{2M^2} + \frac{(1+c)^2 M^2}{2\beta \log(1+M^2)} \log\left(TM^2 + 1\right),$$

where $M = \max\{\epsilon, 1/\rho_{\text{deploy}} - (1+\epsilon)\}$.

Theory 3: Non-stationary Data

Let $\nu(u)$ be the prior probability mass for threshold u with $\sum_{u\in\mathcal{U}}\nu(u)=1$. The data-dependent threshold $\hat{u}_t$ at time t is determined by

$$\hat{u}_t = \max\left\{u\in\mathcal{U} : K_t(u) \geq \frac{1}{\alpha\nu(u)}\right\}.$$

For $u\in\mathcal{U}$, define the weighted cumulative risk at time t by

$$L_t(u) = \sum_{j=1}^{t}\frac{\lambda_j(u)}{\sum_{i=1}^{t}\lambda_i(u)}\mathbb{E}[l_j\mathbb{I}\{\xi_j(u)=0\}|\mathcal{F}_{j-1}].$$

Theorem 3: Safety for Non-Stationary Data

Consider that $(X_t)_{t\geq 0}$ is an arbitrary data stream. For any prior probability mass ν with domain $\mathcal{U}$, and any $\rho_t\in(0,1)$, $\lambda_t\in\mathcal{F}_{t-1}$ with $1+\lambda_t D_t(u)\geq 0$, we have

$$\mathbb{P}(\forall t\in\mathbb{N} : L_t(\hat{u}_t)\leq\epsilon)\geq 1-\alpha.$$

Outline

Experiments: Basic Setup

Datasets. MATH, MMLU-Pro, BIG-Bench Hard (BBH), and Magpie.

LLMs. *Thinking model*: `Qwen3-4B-Thinking-2507`;
Non-thinking model: `Qwen3-4B-Instruct-2507`.

Loss function. *Verifiable tasks*: 0-1 loss;
Open-ended tasks: LLM-as-a-judge loss.

Evaluation: We evaluate reasoning efficiency through two metrics: *Expert Call Percentage* (ECP) and *Token Percentage* (TP). For $t \geq 1$, we define

$$\mathrm{ECP}_t = \frac{1}{t}\sum_{i=1}^{t} \mathbb{I}\{\xi_i = 1\} \times 100\%,$$

$$\mathrm{TP}_t = \frac{\sum_{i=1}^{t}(\tilde{h}(X_i) + h(X_i)\mathbb{I}\{\xi_i = 1\})}{\sum_{i=1}^{t} h(X_i)} \times 100\%.$$

where $\tilde{h}(x_i)$ and $h(x_i)$ represent the number of tokens of $\tilde{f}(x_i)$ and $f(x_i)$, respectively.

Experiments: Basic Setup

The safety is evaluated by the *Empirical Risk* (ER):

$$\mathrm{ER}_t = \frac{1}{t}\sum_{i=1}^{t} l(\hat{f}_i(X_i), f(X_i)).$$

Baselines.

- *PAC reasoning* (designed for offline setting).
- *O-naive and IPS+Hoeff* (O-naive has no safety guarantee and IPS+Hoeff has safety guarantee by leveraging the Hoeffding's inequality).
- *Chain of Draft* (CoD) and *No-Thinking*.

Experiments: Efficiency Outperforms Offline Method

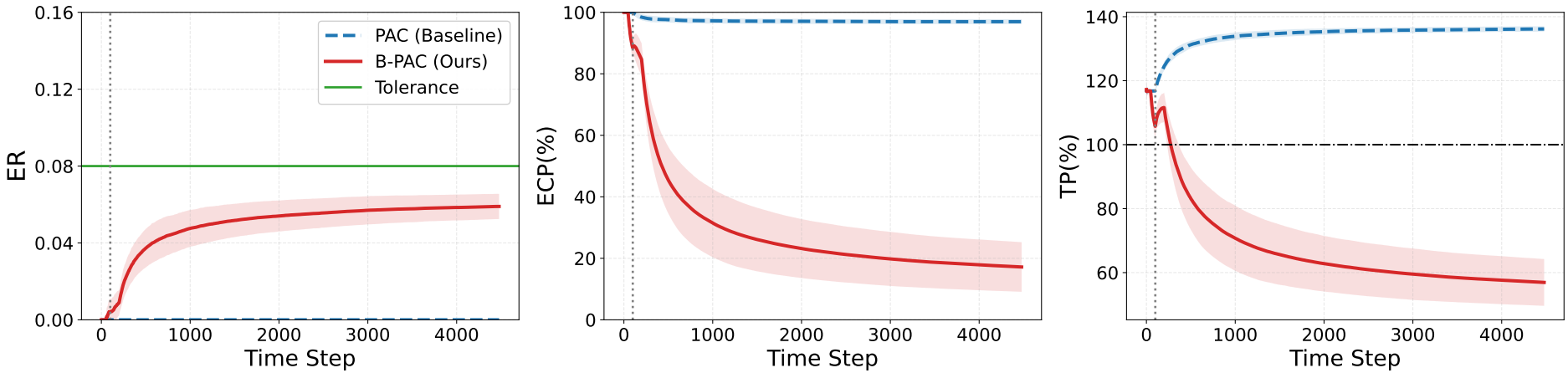


Figure 5.1: **Efficiency outperforms offline PAC reasoning**. ER, ECP, and TP are reported on a combined dataset of Magpie and BBH, with $\epsilon = 0.08$ and $\alpha = 0.1$. The vertical dotted line indicates the size of the calibration set used for the offline PAC baseline. Experiments are repeated 100 times, and the shaded areas represent standard deviations.

Achieving ECP $= 18.99\%$ and TP $= 58.63\%$!.

Experiments: Risk control under Non-stationary Settings

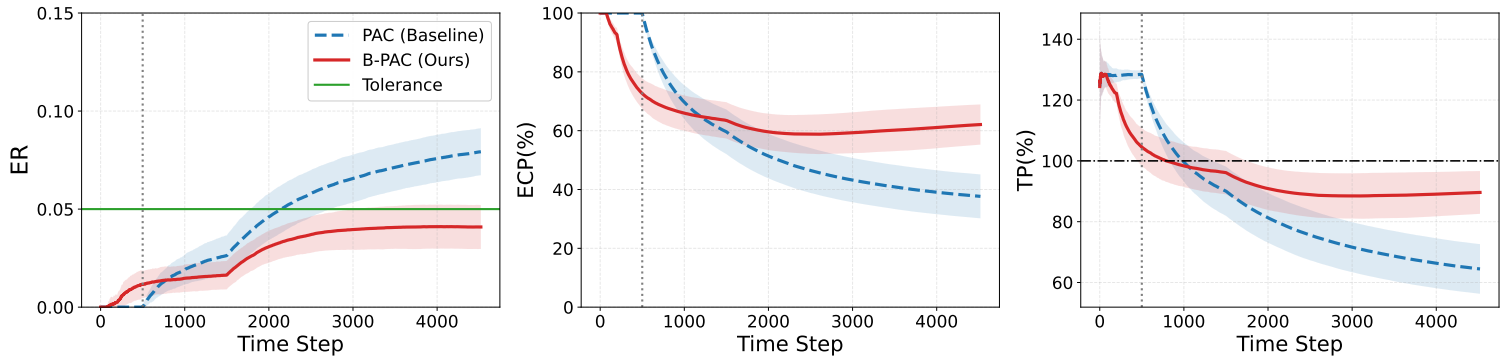
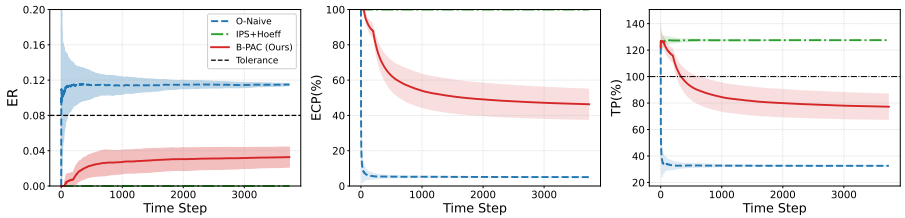


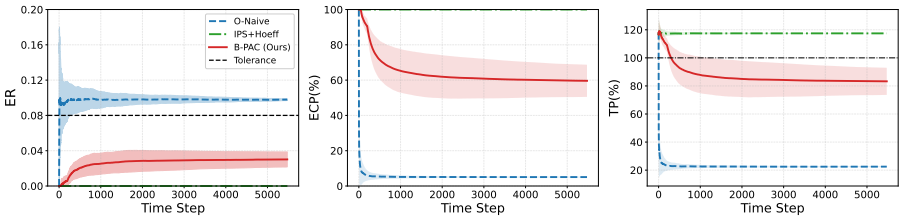
Figure 5.2: **Anytime safety for non-stationary data**. Results are reported on a combined dataset of MMLU-Pro and BBH, with $\epsilon = 0.05$ and $\alpha = 0.1$.

Always safe even under non-stationary settings by dynamically adjusting threshold.

Experiments: Comparison with Online Methods



(a) Results on MMLU-Pro.



(b) Results on BBH.

Figure 5.3: **Safety and efficiency of B-PAC reasoning compared with online methods, including IPS+Hoeff and O-Naive.** Results are reported on MMLU-Pro and BBH benchmarks, with $\epsilon = 0.08$ and $\alpha = 0.1$.

Always safe and efficient!

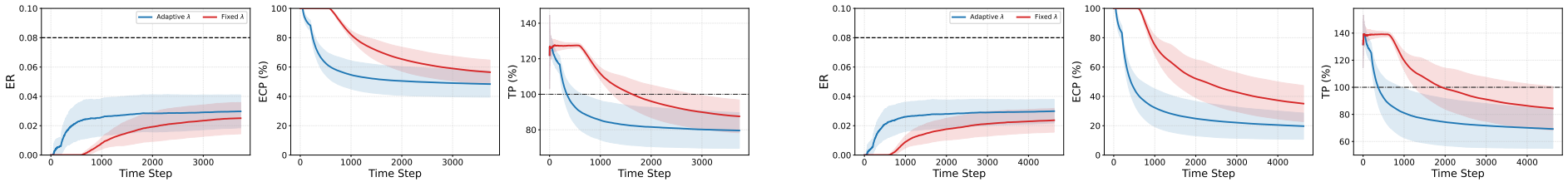
Experiments: Comparison with CoD and NoThinking

Table 5.1: Results of B-PAC reasoning, CoD, and NoThinking on MATH and MMLU-Pro, with $\epsilon = 0.08$ and $\alpha = 0.1$.

	MATH			MMLU-Pro		
Metric	**B-PAC**	**CoD**	**NoThinking**	**B-PAC**	**CoD**	**NoThinking**
ER $\downarrow$	$\mathbf{0.03 \pm 0.01}$	0.1204	0.1577	$\mathbf{0.03 \pm 0.01}$	0.10	0.12
TP (%) $\downarrow$	$\mathbf{68.16 \pm 14.56}$	95.63	77.26	77.24 ± 9.94	$\mathbf{73.63}$	76.52

Always safe and efficient!

Ablation Study on Impact of the Adaptive Betting Strategy



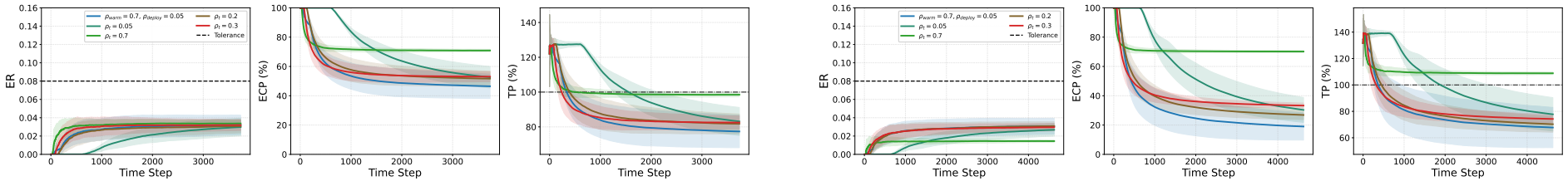
(a) MMLU-Pro

(b) MATH

Figure 5.4: **Ablation study on the betting strategy λ_t on MMLU-Pro and MATH.** The blue curve represents the adaptive betting strategy given by (3.5). The red curve represents the fixed betting strategy with $\lambda_t \equiv 0.05$. Results demonstrate that the adaptive betting strategy is more efficient than the fixed betting strategy.

The adaptive approach rapidly reduces expert invocations in the early stages, whereas the fixed λ_t approach remains overly conservative, yielding a much slower decline in computational cost.

Ablation Study on Impact of the Exploration Strategy



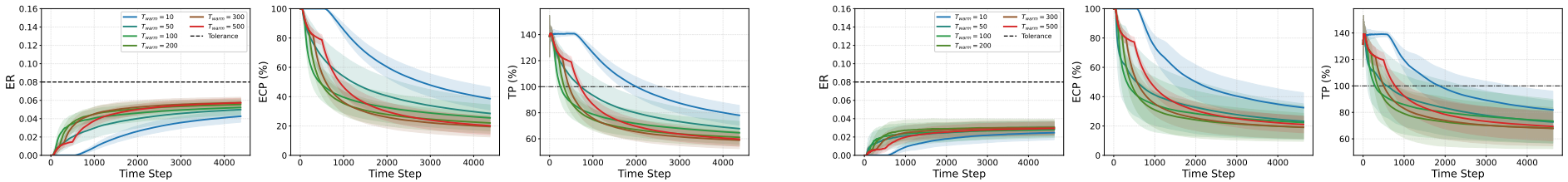
(a) MMLU-Pro

(b) MATH

Figure 5.5: **Ablation study on the exploration probability ρ_t.** We compare the proposed two-stage exploration strategy with fixed exploration probabilities $\rho_t \in \{0.05, 0.2, 0.3, 0.7\}$ on MMLU-Pro and MATH. The results show that the two-stage strategy is more efficient.

Two-stage strategy effectively searches for the optimal threshold in the early stage and reduces exploration cost in the later stage.

Sensitivity to warm-up duration T_{warm}



(a) Magpie

(b) MATH

Figure 5.6: **Sensitivity to Warm-up Duration T_{warm}.** We compare the performance of B-PAC reasoning for $T_{\text{warm}} \in \{10, 50, 100, 200, 300, 500\}$ on Magpie and MATH. The results demonstrate the robustness of medium-sized T_{warm}.

A medium-sized T is preferred. B-PAC reasoning demonstrates **strong robustness** across a wide range of intermediate values, suggesting that precise hyperparameter tuning is not required for deployment.

Outline

Conclusions

- We propose B-PAC reasoning, a method for efficient online reasoning under partial feedback, without any access to the offline calibration dataset. To the best of our knowledge, B-PAC reasoning is the first safe, model-agnostic, efficient reasoning method in online and non-stationary settings.
- Theories: anytime-valid performance loss control for both i.i.d. and non-stationary data, as well as the efficiency of the threshold-updating strategy.
- We provide comprehensive experiments on diverse reasoning benchmarks, demonstrating that B-PAC reasoning is anytime safe and efficient.

Also suitable for other scenarios with cost-accuracy tradeoff, such as multi-agent routing.

Reference

[1] Yu, C., et al. "Anytime Safe PAC Efficient Reasoning." In Proceedings of the International Conference on Machine Learning (ICML), 2026.
[2] Yu, C., et al. "A Generalized E-value Feature Detection Method with FDR Control at Multiple Resolutions." arXiv preprint arXiv:2409.17039, 2024.
[3] Zeng, H., et al. "PAC Reasoning: Controlling the Performance Loss for Efficient Reasoning." arXiv preprint arXiv:2510.09133, 2025.
[4] Angelopoulos, A. N. and Bates, S. Conformal Prediction: A Gentle Introduction. Foundations and Trends in Machine Learning, 16(4):494–591, 2023.
[5] Angelopoulos, A. N., et al. Conformal Risk Control. In The Twelfth International Conference on Learning Representations (ICLR), 2024.

Thank you!