A Theoretical Study on Bridging Internal Probability and Self-Consistency for LLM Reasoning

Zhi Zhou [1], Yuhao Tan [1], Zenan Li [2], Yuan Yao [1], Lan-Zhe Guo [1], Yu-Feng Li [1,✉], Xiaoxing Ma [1,✉]

[1]State Key Laboratory of Novel Software Technology, Nanjing University, China

[2]Department of Computer Science, ETH Zurich, Switzerland ✉ Corresponding Author

TL; DR We introduce the **first theoretical framework for LLM reasoning**, and bridge two test-time scaling methods to achieve both **low error and fast convergence**.

Theoretical Framework and Insights

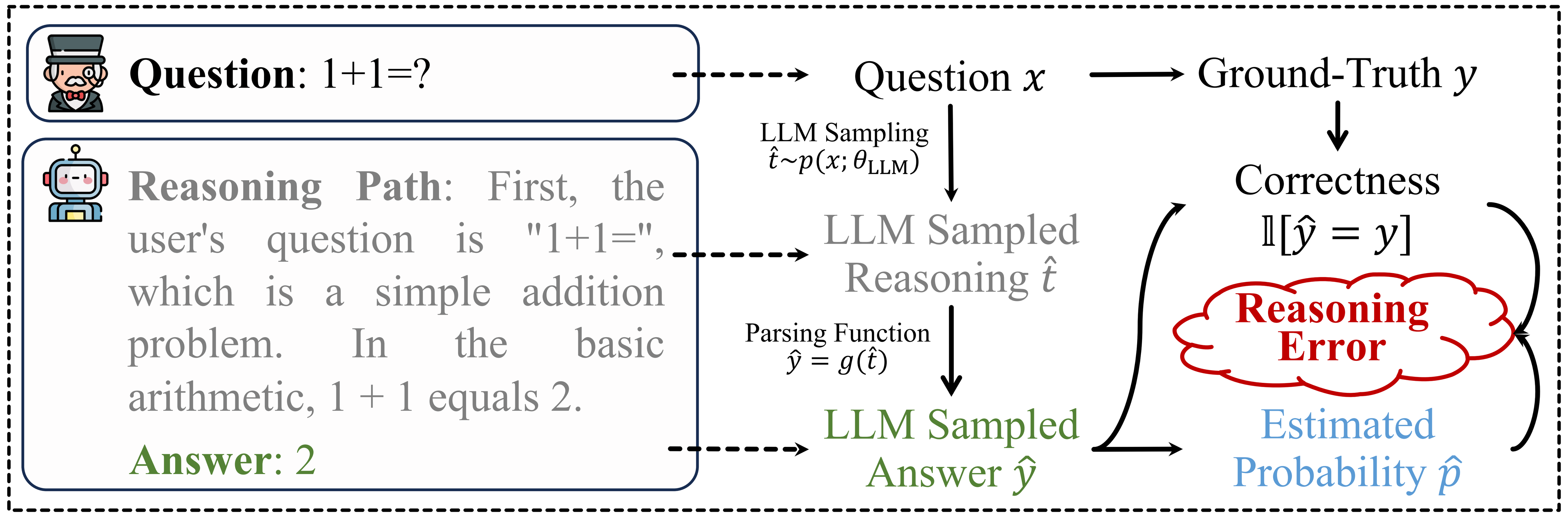

Figure 1: Formulation of LLM Reasoning and Reasoning Error

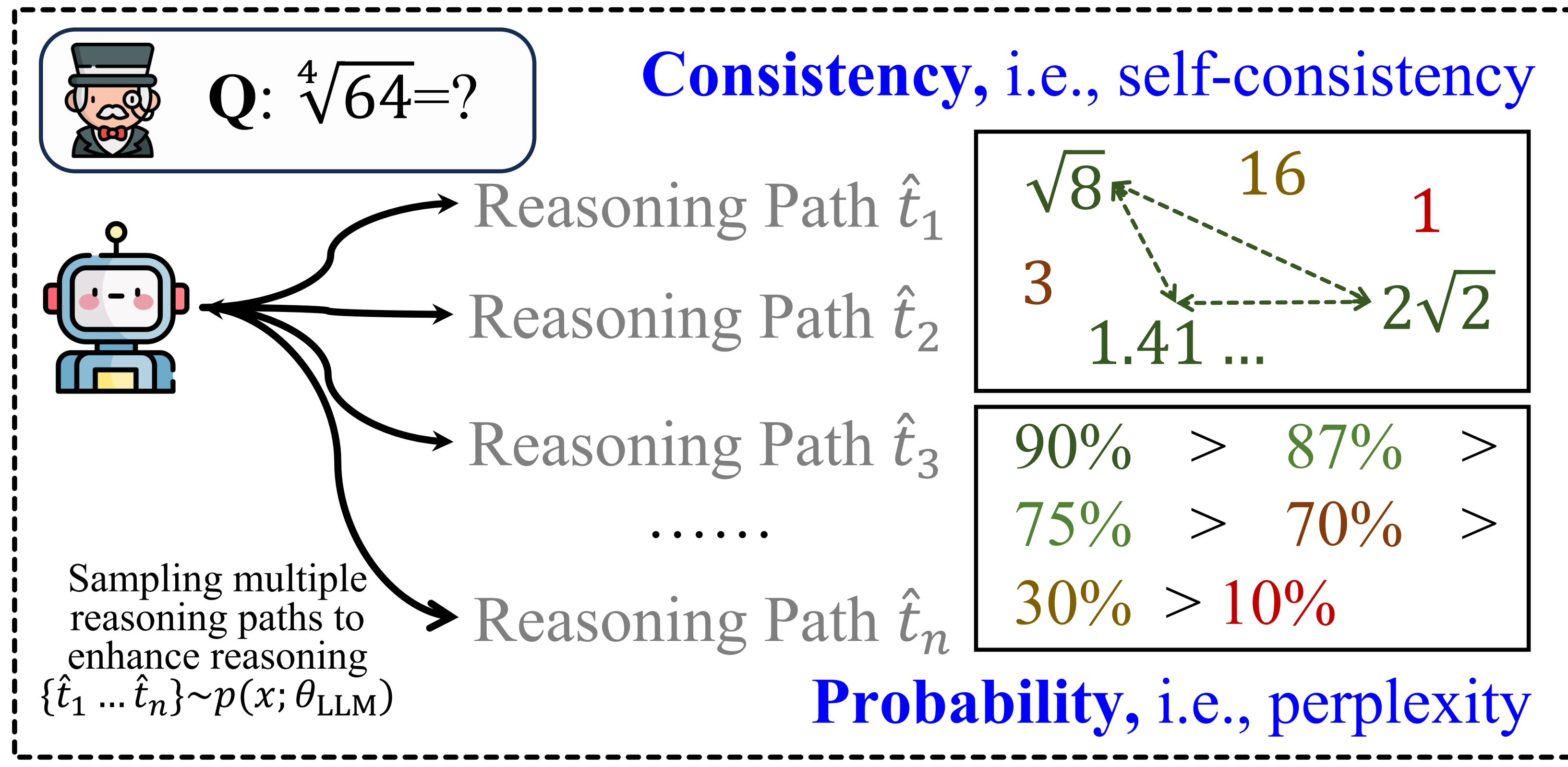

Figure 2: Sampling-Based Test-Time Scaling Methods

Contribution #1: We introduce the first theoretical framework for LLM reasoning in the context of confidence estimation, which evaluates the provided $\hat{p}(\hat{y} \mid x)$ for each candidate answer $\hat{y}$.

Definition of Reasoning Error

First, the reasoning error for each candidate answer $\hat{y}$ is defined based on its confidence estimation and correctness as follows:

$$\mathcal{E}_{\hat{p}}(\hat{y}) = \mathbb{E}\left[\left(\hat{p}(\hat{y} \mid x) - \mathbb{I}[\hat{y} = y]\right)^2\right]$$

where the expectation is taken over all possible LLM samplings used to compute the confidence estimation.

Decomposition of Reasoning Error

Then, the reasoning errors are decomposed into two components: the estimation error and the model error:

$$\mathcal{E}_{\hat{p}}(\hat{y}) = \underbrace{\mathbb{E}\left[\left(\hat{p}(\hat{y} \mid x) - p(\hat{y} \mid x)\right)^2\right]}_{\text{Estimation Error}} + \underbrace{\left(p(\hat{y} \mid x) - \mathbb{I}[\hat{y} = y]\right)^2}_{\text{Model Error}}$$

where the estimation error is related only to the estimation algorithm, while the model error is related only to the LLM itself.

Analysis of Self-Consistency

Self-consistency (SC) adopts Monte Carlo estimation:

$$\mathcal{E}_{\hat{p}^{(\mathrm{SC})}}(\hat{y}) = \underbrace{\frac{1}{n} p(\hat{y} \mid x)(1 - p(\hat{y} \mid x))}_{\text{Estimation Error}} + \underbrace{\left(p(\hat{y} \mid x) - \mathbb{I}[\hat{y} = y]\right)^2}_{\text{Model Error}}$$

SC only achieves a linear convergence rate of the estimation error corresponding to the sampling size, which results in substantial reasoning error when sampling is limited.

Analysis of Perplexity

Perplexity (PPL) uses internal probability as confidence:

$$\mathcal{E}_{\hat{p}^{(\mathrm{PPL})}}(\hat{t}) = \underbrace{(1 - p(\hat{t} \mid x))^n p(\hat{t} \mid x)(2\mathbb{I}[\hat{y}_i = y] - p(\hat{t} \mid x))}_{\text{Estimation Error}} + \underbrace{\left(p(\hat{t} \mid x) - \mathbb{I}[g(\hat{t}) = y]\right)^2}_{\text{Model Error}}$$

The estimation error of PPL decreases exponentially, but the rate depends on the value of the ground-truth confidence; the model error of PPL is not satisfactory due to ignoring parsing function $g(\cdot)$.

RPC Method

Contribution #2: By combining the strengths of both SC and PPL, we introduce the RPC method.

- **Perplexity Consistency (PC)** bridges the SC and PPL methods to achieve both low model error and fast estimation error convergence, but its convergence may degrade as $\alpha \to 1$.

Theorem 4 (PC Reasoning Error Decomposition). *Assume that $k = |\{\tilde{t} \mid g(\tilde{t}) = \hat{y}\}|$ and define $\alpha := 1 - \frac{1}{k} p(\hat{y} \mid x)$. Then, the reasoning error $\mathcal{E}(\hat{p}^{(\mathrm{PC})})$ of PC can be divided into two components:*

$$\mathcal{E}_{\hat{p}^{(\mathrm{PC})}}(\hat{y}) = \underbrace{\alpha^n p(\hat{y} \mid x)\left(2\mathbb{I}[\hat{y} = y] - (1 + \alpha^n) p(\hat{y} \mid x)\right)}_{\text{Estimation Error}} + \underbrace{\left(p(\hat{y} \mid x) - \mathbb{I}[\hat{y} = y]\right)^2}_{\text{Model Error}}.$$

- **Reasoning Pruning (RP)** eliminates degradation cases by automatically pruning reasoning paths that are not useful, thereby ensuring the theoretical guarantees.

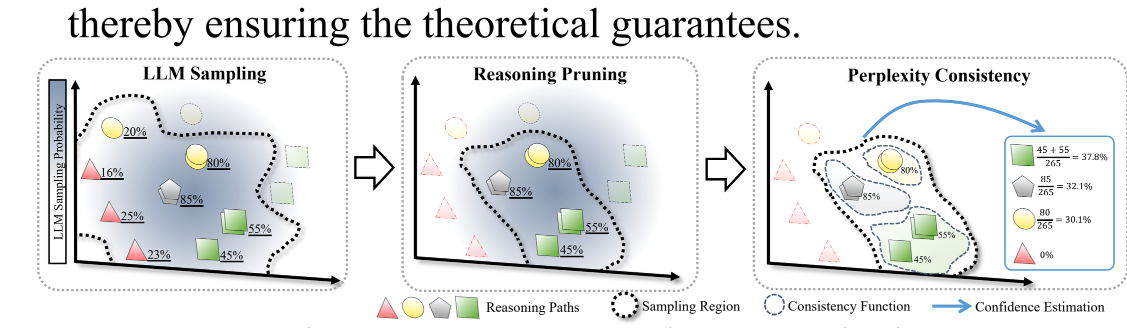

Figure 3: Our proposed RPC method

Experiments

Contribution #3: Our theoretical results align with practice, e.g., RPC can enhance both efficiency and reliability with broad applications.

Method	MATH		MathOdyssey		OlympiadBench		AIME	
	Accuracy	#Samplings	Accuracy	#Samplings	Accuracy	#Samplings	Accuracy	#Samplings
Best of SC	50.57	64	28.32	112	11.07	128	9.40	128
PC	50.63	32	28.51	112	11.07	128	9.00	64
Δ	+0.06	**-50.0%**	+0.19	-0.0%	0.00	-0.0%	0.00	**-50.0%**
RPC	51.16	32	29.31	32	11.07	64	9.50	48
Δ	+0.59	**-50.0%**	+0.99	**-71.4%**	0.00	**-50.0%**	+0.10	**-62.5%**

Table 1: **Efficiency.** RPC achieves equal or better performance as SC while using 50% fewer samples.

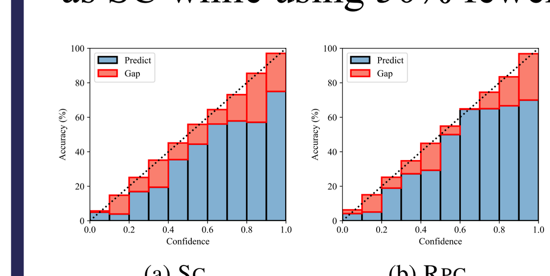

Figure 4: **Reliability.** RPC improves reliability diagrams compared to SC.

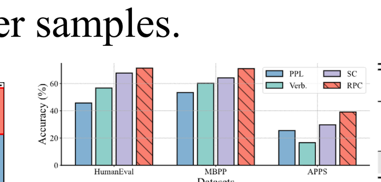

Figure 5: Code Generation

Method	GPQA	LogiQA
PPL	41.46	54.36
SC	43.00	56.71
RPC	**44.09**	**58.42**

Table 2: Commonsense

Method	MathOdyssey	AIME	MATH	OlympiadBench
PPL	60.04	72.92	81.81	21.65
SC	57.22	70.40	82.03	21.93
RPC	**61.11**	**76.47**	**82.78**	**22.81**

Table 3: Results on R1 LLMs

Generality. Effectiveness on code, commonsense and R1 LLMs.

Project Homepage https://wnjxyk.github.io/RPC

Project Demo https://wnjxyk-rpc.hf.space

A Theoretical Study on Bridging Internal Probability and Self-Consistency for LLM Reasoning
Published on Oct 17 · Submitted by Zhi Zhou on Oct 20 #1 Paper of the day · NJU-IRP
Authors: Zhi Zhou, Yuhao Tan, Zenan Li, Yuan Yao, Lan-Zhe Guo, Yu-Feng Li, Xiaoxing Ma

Hugging Face #1 Paper of The Day https://huggingface.co/papers/2510.15444