NeurIPS 2025 Poster

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

Zhi Zhou, Yuhao Tan, Zenan Li, Yuan Yao, Lan-Zhe Guo, Yu-Feng Li, Xiaoxing Ma

Nanjing University, China | ETH Zurich, Switzerland

Presented by Yejin Yoon

Pre-Requisite

The Core Problem of LLM Reasoning

Accuracy vs. Consistency

Pre-requisites — The Core Problem of LLM Reasoning

🤔 Confidence Estimation Problem

LLMs generate text probabilistically — same input can produce different outputs each run.

Q. What is $\sqrt{144}$?

- ✓ Try 1: "12"
- ✓ Try 2: "The answer is 12"
- ✗ Try 3: "14"
- ✓ Try 4: "12"
- ✗ Try 5: "16"

→ **"12" *appeared 3 times, but is that enough?***

Why is this challenging?

⛳ Goal : Pick the best answer from N

- **Best-of-N** Strategy :
  - Generate N reasoning paths
  - Score each with a confidence estimate
  - Select the highest-scored answer
- But how do we measure **confidence** ?

(1) LLM sampling is *stochastic* by design → (2) how many samples do we need to be sure?
→ (3) we can't check against ground truth — ***No labels at test time*** 🤔

Pre-requisites — Accuracy vs. Consistency

We want accuracy — but we can't use it in practice 😕

In real deployment we don't know the ground-truth answer — so how do we judge quality?

Accuracy – *vs. ground truth*

✓ "12"
✓ "The answer is 12"
✗ "14"
✓ "12"
✗ "16"

→ **answer: 12** ✓

(*in deployment*)
No one knows the correct answer !
– until it's verified

*Not usable in **real deployment** :*

- Requires knowing the correct answer
- Only for **offline** evaluation *(can measure after the fact)*

Consistency – *vs. each other*

✓ "12"
✓ "The answer is 12"
✗ "14"
✓ "12"
✗ "16"

→ **"12" wins (3/5)**

Answer	**12**	14	16
Count	**3**	1	1

*Practical **confidence** proxy :*

- No ground truth needed; works in real deployment
- Enables *Best-of-N* selection

Consistency may be the only practical proxy for **confidence** in real deployment.

Contents

1 Background

- Why does confidence estimation matter for LLM reasoning?

2 Suggestion #1 Theoretical Framework

- Error decomposition – analyzing SC and PPL rigorously

3 Suggestion #2 Method: RPC

- Perplexity Consistency + Reasoning Pruning

4 Experimental Result

- Efficiency, Efficacy, Reliability – 7 benchmarks

Background

How Have We Been Measuring Consistency?

Why Does *Consistency* Matter?

Background

How Have We Been Measuring Consistency?

Three main approaches for estimating confidence from sampled reasoning paths:

1 Self-Consistency (SC)

Wang, Xuezhi, et al. "Self-Consistency Improves Chain of Thought Reasoning in Language Models." (ICLR 2023).

Generate n paths → majority vote. Confidence = fraction of paths giving the same answer.

Log-prob. : Not needed ✗
→ Open & Closed source

2 Perplexity (PPL)

Stanley F Chen, Douglas Beeferman, and Roni Rosenfeld. “Evaluation metrics for language models.” (1998).

Use LLM's internal token probabilities (mean log-prob per token). *"How confidently did the model write this path?"*

Log-prob. : Required ✓
→ Open-source only

3 Verbalized Confidence

Kadavath, Saurav, et al. "Language models (mostly) know what they know." arXiv preprint arXiv:2207.05221 (2022).

Ask LLM: *"Is the proposed answer correct? How confident are you? (0%–100%)"*

Log-prob. : Not needed ✗
→ Open & Closed source

🤔 Each method works in practice — but none has a ***theoretical foundation***.

Background

How Have We Been Measuring Consistency?

Three main approaches for estimating confidence from sampled reasoning paths:

1 Self-Consistency (SC)

Wang, Xuezhi, et al. "Self-Consistency Improves Chain of Thought Reasoning in Language Models." (ICLR 2023).

$\log prob.$: Not needed ✗
→ Open & Closed source

Generate n paths → majority vote. Confidence = fraction of paths giving the same answer.

Q. What is $\sqrt{144}$?

- ✓ "12"
- ✓ "The answer is 12"
- ✗ "14"
- ✓ "12"
- ✗ "16"

Answer	**12**	14	16
Count	**3**	1	1

(out of 5 samples)

$$\hat{p}^{(\mathrm{SC})}(\hat{y} \mid x) = \frac{1}{n}\sum_{i=1}^{n} \mathbb{I}[\tilde{y}_i = \hat{y}]$$

✓ Low model error — consistency function aggregates equivalent paths
✗ Slow convergence — error decreases only linearly $O(1/n)$
✗ Needs many samples to be reliable

Correct answers *converge* naturally; incorrect ones *scatter* — majority vote does the rest.

Background

How Have We Been Measuring Consistency?

Three main approaches for estimating confidence from sampled reasoning paths:

2 Perplexity (PPL)

Stanley F Chen, Douglas Beeferman, and Roni Rosenfeld. “Evaluation metrics for language models.” (1998).

Use LLM's internal token probabilities (mean log-prob per token). *"How confidently did the model write this path?"*

Log-prob. : Required ✓
→ Open-source only

Q. What is $\sqrt{144}$?

- ✓ "12” → 0.85
- ✓ "The answer is 12” → 0.72
- ✗ "14” → 0.45
- ✓ "12” → 0.80
- ✗ "16” → 0.3

Answer	**12**	14	16
PPL Score	**0.85**	0.45	0.3

$$\hat{p}^{(PPL)}(\hat{t} \mid x) = \sum_{\tilde{t} \in \mathcal{R}} \mathbb{I}[\tilde{t}_i = \hat{t}]\, p(\tilde{t}_i|x)$$

✓ Fast convergence — exponential $O((1-p)^n)$ when p is large
✗ High model error — evaluates paths individually, no aggregation
✗ Degrades when $p \to 0$ — advantage collapses to linear on hard problems

Each path is judged ***alone*** — same answer, different wording, different score → model error 🔺

Background

How Have We Been Measuring Consistency?

Three main approaches for estimating confidence from sampled reasoning paths:

3 Verbalized Confidence

Kadavath, Saurav, et al. "Language models (mostly) know what they know." arXiv preprint arXiv:2207.05221 (2022).

Ask LLM: *"Is the proposed answer correct? How confident are you? (0%–100%)"*

Log-prob. : Not needed ✗
→ Open & Closed source

Q. What is $\sqrt{144}$? 🤔 ***How confident are you?***

✓	"12"	→ 90% confident this is correct !
✓	"The answer is 12"	→ 60%
✗	"14"	→ 15%
✓	"12"	→ 65%
✗	"16"	→ 60%

Answer	**12**	14	16
Self-conf.	**0.9**	0.15	0.6

✓ Intuitive — no special access required

✗ Very poor calibration — ECE* avg. 72+ across all benchmarks

* ECE: *average gap between **predicted confidence** and **actual accuracy** (lower = better)*

✗ Requires careful prompt engineering — output format is inconsistent across models

LLMs often don't know ***what they don't know*** — stated confidence rarely reflects true accuracy.

Motivation — Why Does ***Consistency*** Matter?

👀 We need a method that is *fast*, *accurate*, AND *well-calibrated*

Poor confidence estimation has direct, real-world consequences:

Why Confidence Estimation Matters

1. **Must work without ground truth**
   : No labels at inference time
   → Consistency is the only viable proxy
2. **Drives Best-of-N selection**
   : Poor confidence → wrong answer gets picked, regardless of how many samples
3. **Directly controls sampling cost**
   : Better confidence
   = fewer samples needed = lower inference cost

Why Existing Methods Fall Short

SC Good consistency, but slow convergence
Needs many samples to be reliable

PPL Fast convergence, but no aggregation
Confidence varies by phrasing, not correctness

Both work empirically
— but no one could explain why, or how to improve them.

[***Research question***] Can we bridge the fast convergence of PPL and the low model error of SC?

Theoretical Framework

Proposition 1 Error Decomposition

Proposition 2 SC Error Decomposition

Proposition 3 PPL Error Decomposition

Theoretical Framework — Error Decomposition

Proposition 1 LLM reasoning error can be formally split into two independent parts.

Reasoning Error = Estimation Error + Model Error

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

Estimation Error *Model Error*

Estimation Error (we can control)

How far is estimated confidence from the true confidence?

- Depends on sample size n and estimation strategy
- Shrinks as n ▲ — but **how fast** depends on the *method*

→ This is what differentiates *SC* and *PPL*

Model Error (fixed ; cannot control)

How far is true confidence from the correct answer?

- Reflects the LLM's inherent reasoning capability
- Independent of n — fixed for a given model

→ Can only improve by using a **better LLM**

How fast does each method shrink Estimation Error?

Theoretical Framework — Analysis of Self-Consistency

Proposition 2 SC Reasoning Error Decomposition

Reasoning Error = Estimation Error + Model Error

$$\mathcal{E}_{\hat{p}(sc)}(\hat{y}) = \mathbb{E}\left[\left(\frac{1}{n}\sum_{i=1}^{n}\mathbb{I}[\tilde{y}_i = \hat{y}] - \mathbb{I}[\hat{y} = y]\right)^2\right] + (p(\hat{y}|x) - \mathbb{I}[\hat{y} = y])^2$$

Estimation Error ▲ *Model Error* ▼

Estimation Error : $\frac{1}{n}\, p(\hat{y}|x)(1 - p(\hat{y}|x))$ → ***Linear convergence*** $\mathbf{O}\left(\frac{1}{n}\right)$

- ✓ "12"
- ✓ "The answer is 12"
- ✗ "14"
- ✓ "12"
- ✗ "16"

→ Did the LLM answer $\hat{y} = 12$ this time? ✓ or ✗

SC $\hat{p}^{(\mathrm{SC})}(\hat{y} = 12) = \frac{3}{5} = 0.6$ → Var(sample mean of n Bernoulli trials)

🐢To halve the error: must double the number of samples

Theoretical Framework — Analysis of Self-Consistency

Proposition 2 SC Reasoning Error Decomposition

Reasoning Error = Estimation Error + Model Error

$$\mathcal{E}_{\hat{p}(sc)}(\hat{y}) = \mathbb{E}\left[\left(\frac{1}{n}\sum_{i=1}^{n}\mathbb{I}[\tilde{y}_i = \hat{y}] - \mathbb{I}[\hat{y} = y]\right)^2\right] + (p(\hat{y}|x) - \mathbb{I}[\hat{y} = y])^2$$

Estimation Error ▲ *Model Error* ▼

Estimation Error : $\frac{1}{n}\, p(\hat{y}|x)(1 - p(\hat{y}|x))$ → ***Linear convergence*** $\mathbf{O}\left(\frac{\mathbf{1}}{\boldsymbol{n}}\right)$

Model Error : SC operates at **answer level**, not path level
→ Different reasoning paths leading to the same answer are merged

SC is reliable but slow. We need faster Estimation Error convergence.

Theoretical Framework — Analysis of Perplexity

Proposition 3 *PPL* Reasoning Error Decomposition

Reasoning Error = Estimation Error + Model Error

$$\mathcal{E}_{\hat{p}^{(\mathrm{PPL})}}(\hat{t}) = \mathbb{E}\left[\left(\sum_{\tilde{t}\in\mathcal{R}} \mathbb{I}[\hat{t}=\tilde{t}]\,p(\tilde{t}|x) - \mathbb{I}[g(\hat{t})=y]\right)^2\right] + (p(\hat{y}|x) - \mathbb{I}[\hat{y}=y])^2$$

Estimation Error ▼ *Model Error* ▲

Estimation Error : $\left(1-p(\hat{t}|x)\right)^n p(\hat{t}|x)(2\mathbb{I}[\hat{y}_i=y] - p(\hat{t}|x))$ → ***Exponential convergence*** $\mathbf{O}((\mathbf{1}-\boldsymbol{p})^{\boldsymbol{n}})$

Model Error :

- High Model Error — no consistency function, evaluates each path in isolation
- Degrades when $p \to 0$: $(1-p)^n \approx 1/(1+np)$ → collapses to ***linear*** !

PPL is fast but unreliable. High model error + degrades on hard problems.

Theoretical Framework — Error Decomposition

Proposition 1 LLM reasoning error can be formally split into two independent parts.

Reasoning Error = Estimation Error + Model Error

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

Estimation Error *Model Error*

Method	Estimation Error Convergence	Model Error	Key Problem
SC	Linear O(1/n) ✗	Low ✓	Too slow with limited sampling
PPL	Exponential $O(\alpha^n)$ ✓	High ✗	Degrades at low probability
RPC (ours)	Exponential ✓	Low ✓	Best of both worlds

- Exponential **Estimation Error** (from *PPL*) + Low **Model Error** (from *SC*)
- No degradation at low probability (neither method has this)

Method: RPC

Component 1 PC

Component 2 RP

Method : RPC

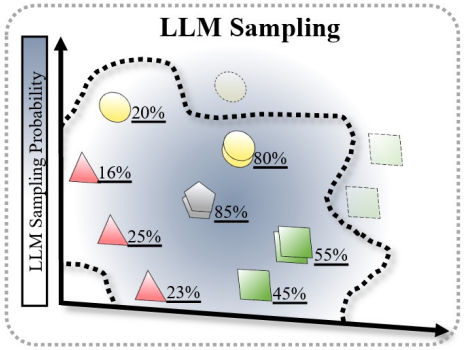
Reasoning Paths Sampling Region Consistency Function Confidence Estimation

RPC: Reasoning-Pruning Perplexity Consistency

RPC is a post-hoc confidence estimation method with two sequential components:

Input
LLM Sampling

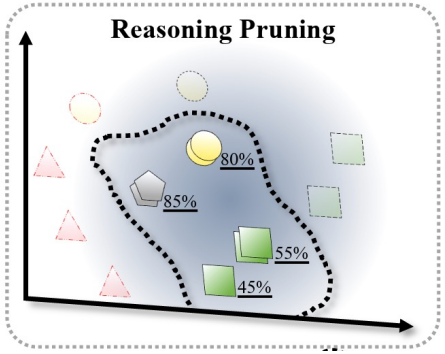
- Sample n reasoning paths $\tilde{t}_1, \ldots, \tilde{t}_n$
- Obtain $mean_{log-prob.}$ for each path (geometric mean of token log-probs)
- Get final answer via extraction $g(\cdot)$



Phase 1 — Component RP
Reasoning Pruning

- Fit 2-component Weibull mixture to the probability distribution
- Remove paths where $P_{High} < 0.5$ & prob $< overall\ mean$

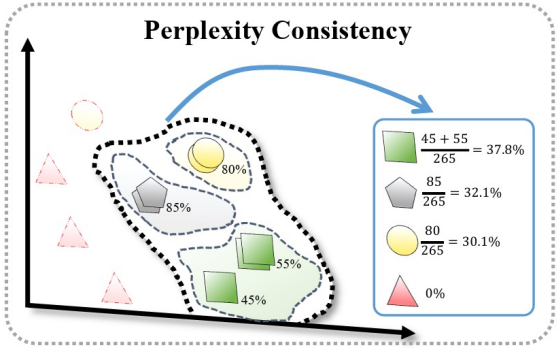
→ Removes low-confidence paths



Phase 2 — Component PC
Perplexity Consistency

- For each candidate answer $\hat{y}$:
- Confidence$(\hat{y}) = \sum p(\tilde{t}|x)$ for all retained paths where $g(\tilde{t}) = \hat{y}$

→ Select highest-confidence answer



Method : RPC

Component 1 Perplexity Consistency (PC)

PC fuses SC's consistency aggregation with PPL's internal probabilities (Theorem 4):

* Perplexity Consistency (PC)

Combine SC's answer-level consistency + PPL's internal probability

Q. What is $\sqrt{144}$?

- ✓ "12" → 0.30
- ✓ "The answer is 12" → 0.25
- ✗ "14" → 0.35
- ✓ "12" → 0.06
- ✗ "16" → 0.04

Answer	**12**	14	16
PC Score	**0.61**	0.35	0.04

(out of 5 samples)

$$\hat{p}^{(\mathrm{PC})}(\hat{y} \mid x) = \sum \mathbb{I}[g(\tilde{t}) = \hat{y}]p(\tilde{t}|x)$$

✓ Exponential convergence + Low Model Error

✗ Like SC → checks answer-level consistency → model error stays low

✗ Like PPL → uses probability information → estimation error converges faster

🐇vs. SC: Same winner ("12"), but PC converges faster with fewer samples

Method : RPC

Component 1 Perplexity Consistency (PC)

PC fuses SC's consistency aggregation with PPL's internal probabilities (Theorem 4):

* Perplexity Consistency (PC)

Combine SC's answer-level consistency + PPL's internal probability

Q. What is $\sqrt{144}$?

- ✓ "12" → 0.30
- ✓ "The answer is 12" → 0.25
- ✗ "14" → 0.35
- ✓ "12" → 0.06
- ✗ "16" → 0.04

Answer	12	14	16
PC Score	**0.61**	0.35	0.04

(out of 5 samples)

$$\hat{p}^{(\mathrm{PC})}(\hat{y} \mid x) = \sum \mathbb{I}[g(\tilde{t}) = \hat{y}]\, p(\tilde{t}|x)$$

✓ Exponential convergence + Low Model Error

✗ Like SC → chooses answer with the highest score

✗ Like PPL → uses probability information → estimation error converges faster

Degradation problem:

when $p \to 0$: $(1-p)^n \approx 1/(1+np)$ → collapses to ***linear*** !

🐇 vs. SC: Same winner ("12"), but PC converges faster with fewer samples + RP

Method : RPC

Component 2 Reasoning Pruning (RP)

If the model itself assigns near-zero probability to a path — just throw it out (Theorem 7).

* Perplexity Consistency (PC)

Combine SC's answer-level consistency + PPL's internal probability

Q. What is $\sqrt{144}$?

- ✓ "12" → 0.30
- ✓ "The answer is 12" → 0.25
- ✗ "14" → 0.35
- ~~✓ "12" → 0.06~~ **Pruned**
- ~~✗ "16" → 0.04~~ **Pruned**

Answer	**12**	14	~~16~~
PC Score	**0.55**	0.35	~~0.04~~

(out of 5 samples)

* Weibull Distribution

$$P_{\text{High}(x)} = \frac{w_1 f_{\text{W}(x;\, k_1,\lambda_1)}}{w_1 f_{\text{W}(x;\, k_1,\lambda_1)} + w_2 f_{\text{W}(x;\, k_2,\lambda_2)}}$$

- ***Before*** pruning: **PC** gets "contaminated" by noise paths
- ***After*** pruning: Cleaner distribution → faster & more reliable convergence

RP filters noise before PC runs — giving RPC cleaner, faster, and more reliable estimation.

Method : RPC

RPC = Reasoning Pruning + Perplexity Consistency

Phase 1 filters out noise paths — Phase 2 aggregates remaining probabilities by answer

```
Algorithm: RPC
──────────────────────────
Input: paths t̃₁..t̃ₙ, probs p₁..pₙ

── Phase 1: Reasoning Pruning
  Fit 2-Weibull mixture to probs
  pmean = mean(p₁..pₙ)
  For each path i:
    if P_High(pᵢ)<0.5 and pᵢ<pmean:
      proba_i = 0   ← pruned

── Phase 2: Perplexity Consistency
  For each candidate answer ŷ:
    conf(ŷ) = Σ pᵢ
      where ans_i=ŷ and pᵢ>0

Return: argmax conf(ŷ)
```

Q. What is $\sqrt{144}$?

- ✓ "12" → 0.30
- ✓ "The answer is 12" → 0.25
- ✗ "14" → 0.35
- ✓ "12" → 0.06
- ✗ "16" → 0.04

Method	Score for ans 12	Score for ans 14	Score for ans 16	Selected
SC	**3 votes**	1 vote	1 vote	12 ✅
PPL	0.30 (best single path)	**0.35**	0.04	14 ❌
PC	0.30+0.25+0.06 = **0.61**	0.35	0.04	12 ✅
RPC	0.30+0.25 = **0.55** (0.06 pruned)	0.35	~~0.04 (pruned)~~	12 ✅

Aggregation + pruning beats single-path selection.

Experimental Result

RQ1 Efficiency

RQ2 Accuracy

RQ3 Reliability

Experimental Setup

Models · Datasets · Baselines · Metrics

1. **Models:** InternLM2-Math-Plus 1.8B/7B, DeepSeekMath-RL 7B/-Coder 33B/R1-Distill 7B
2. **Datasets:** MATH/MathOdyssey, OlympiadBench/AIME, HumanEval/MBPP/APPS, GPQA/LogiQA
3. **Baselines:** PPL, SC, VERB(verbalized), ESC(early-stop SC), BoN + Reward Model
4. **Metrics:** Accuracy ▲, ECE (calibration) ▼, sampling budget n

1	**RQ1 Efficiency**	Can RPC achieve comparable performance with fewer samples?	**sampling budget n**
2	**RQ2 Efficacy**	Does RPC improve reasoning accuracy over existing methods?	**Accuracy ▲**
3	**RQ3 Reliability**	Does RPC produce more trustworthy confidence scores?	**ECE ▼**

RPC evaluated across 7 benchmarks · 5 models · 10 runs each

Effects

RQ1 **Efficiency** : Can **RPC** achieve comparable performance with fewer samples?

RPC achieves **equal or better accuracy** with 50 - 71% fewer samples across **all 4 math benchmarks**.

- Faster convergence (PC component) explains sample reduction on MATH and AIME
- **Reasoning Pruning** additionally resolves the degeneration issue — biggest win on MathOdyssey (−71%!)

Table 1: Efficiency comparison of *Perplexity Consistency* module (PC) and RPC. The table shows the minimum number of samples needed to exceed the best performance of SC, with reduction rates in bold when sampling is reduced.

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%**

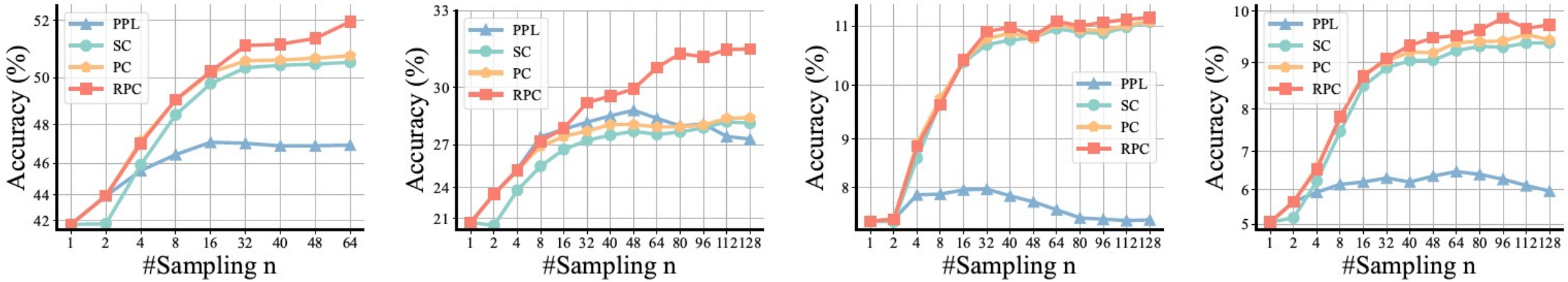
RPC reduces sampling cost more consistently and significantly than PC alone.

Effects

RQ2 **Efficacy** : Does **RPC** improve reasoning accuracy over existing methods?

RPC outperforms all baselines across all 4 datasets, at every sample size.

- RPC converges faster than SC from the very first samples
- PC already beats SC in speed — RPC pushes accuracy even higher
- PPL plateaus early due to high model error; RPC avoids this entirely



(a) MATH (b) MathOdyssey (c) OlympiadBench (d) AIME

Adding Reasoning Pruning on top of PC consistently raises the accuracy ceiling.

Effects

RQ3 **Reliability**： Does **RPC** produce more trustworthy confidence scores?

RPC achieves the highest accuracy and lowest ECE across almost all benchmarks.

- PPL: high accuracy but ECE ~73–89 → completely unreliable confidence
- VERB: lowest accuracy across the board — verbalization doesn't work well
- SC: decent accuracy, reasonable ECE — but RPC beats both
- RPC: best accuracy + best calibration — the only method that wins on both

* ECE: *average gap between **predicted confidence** and **actual accuracy** (lower = better)*

Table 2: Performance Comparison using InternLM-2-MATH-Plus 7B model measured by accuracy and expected calibration error metrics. The best performance is highlighted in **bold**. The results show that our RPC outperforms existing methods in majority of cases.

Method	MATH		MathOdyssey		OlympiadBench		AIME		Average	
	Accuracy(↑)	ECE(↓)	Accuracy(↑)	ECE(↓)	Accuracy(↑)	ECE(↓)	Accuracy(↑)	ECE(↓)	Acc.(↑)	ECE(↓)
PPL	46.99 ± 0.20	48.99 ± 0.19	27.35 ± 1.22	67.70 ± 1.22	7.27 ± 0.36	86.90 ± 0.35	5.96 ± 0.48	88.98 ± 0.49	21.90	73.14
VERB	26.14 ± 0.25	47.46 ± 0.07	10.06 ± 0.61	69.92 ± 0.88	3.68 ± 0.16	84.68 ± 0.25	3.17 ± 0.17	86.29 ± 0.20	10.76	72.09
SC	50.57 ± 0.17	6.71 ± 0.18	28.25 ± 0.60	12.23 ± 0.54	11.07 ± 0.15	20.20 ± 0.16	9.40 ± 0.21	14.35 ± 0.23	24.82	13.37
RPC	**51.95 ± 0.15**	**6.41 ± 0.18**	**31.62 ± 0.75**	**9.87 ± 0.73**	**11.14 ± 0.15**	**18.86 ± 0.18**	**9.74 ± 0.23**	**14.32 ± 0.21**	**26.11**	**12.37**

RPC is not just more accurate — it's more trustworthy.

Contribution

1 First theoretical framework for sampling-based test-time scaling

- Reasoning error = Estimation Error + Model Error — formally and rigorously proven

2 SC is reliable but slow. PPL is fast but unreliable.

- Now we know WHY — and can design something better based on theory

3 RPC achieves the best of both worlds

- Exponential convergence + low model error + no degradation on hard problems

4 50 - 71% fewer samples, better accuracy, better calibration

- Validated across 7 benchmarks, multiple model sizes, multiple tasks

Thank You

Yejin Yoon

HYU NLP Lab.

Dept. of Computer Science
Hanyang University, South Korea

stillwithyou@hanyang.ac.kr