ICML 2026 · EFFICIENT REASONING

Conformal Thinking

Risk Control for Reasoning on a Compute Budget

Xi Wang*, Anushri Suresh*, Alvin Zhang*, Rishi More*, William Jurayj,
Benjamin Van Durme, Mehrdad Farajtabar, Daniel Khashabi, Eric Nalisnick

Johns Hopkins University · Apple — *equal contribution

arXiv:2602.03814

Adaptive stopping turns “set a token budget” into “set a threshold.” We turn it into “set a risk” — an interpretable target a validation set calibrates for you.

01 The problem

Reasoning LLMs *overthink* : more tokens buy accuracy, but much of that compute is wasted. Adaptive early-stopping promises a fix — yet two problems block it in practice.

PROBLEM 1

Thresholds are hard to set

“Stop when confident” just swaps a token budget for a confidence threshold — a trickier knob. Its value is uninterpretable and shifts with every signal, model, and task.

PROBLEM 2

Budget wasted on the unsolvable

On impossible problems, confidence never rises — so an upper-threshold-only rule never fires. The model burns the *entire* budget getting nowhere.

02 Risk control

Every early stop risks an error, so we control that risk directly. The user names a **risk tolerance ε** — a target error rate — and a validation set does the rest.

ε is interpretable — it interfaces with downstream decisions; a raw threshold is not, so we calibrate it away.

Calibration ε → thresholds Alg. 1

1. Enumerate signal–threshold candidates on a held-out validation set.
2. Keep those with risk ≤ ε under a finite-sample **(UCB)** correction, so the guarantee transfers to unseen data.
3. Among survivors, pick the lowest efficiency loss — the most tokens saved.

→ A stopping rule with a high-probability risk guarantee.

One risk target ε in → calibrated thresholds out, with a guarantee. No opaque knobs to hand-tune.

03 Dual thresholds

Two risks, two thresholds. One catches confident success; a novel lower threshold catches confident failure.

signal s_t = {confidence , entropy , probe}

UPPER λ_+
Stop when confident
false positive (confidently wrong)

LOWER $\lambda_-(t)$
Stop when stalled
false negative (premature give-up)

exit τ = min(τ_+, τ_-, T)

The lower threshold rises with token use — halting runs whose confidence fails to keep pace, saving the budget on the unsolvable.

04 Results

risk ≤ ε
Test risk stays under target with high probability; Naïve calibration overfits and breaks it (Fig 4).

↓ tokens
Same accuracy, fewer tokens — gains grow as unsolvable problems dominate (Fig 6).

→ **Division of labor:** solvable instances exit via the upper threshold, unsolvable via the lower.

→ **Ensembling signals** per ε automatically picks the most efficient stop.

→ **Robust** across validation-set size and distribution shift.

MODELS Qwen3-8B · Qwen3-30B-A3B · R1-Distill-32B · Qwen3-VL-8B

DATA AIME · DeepScaleR · GPQA-Diamond · MathVision

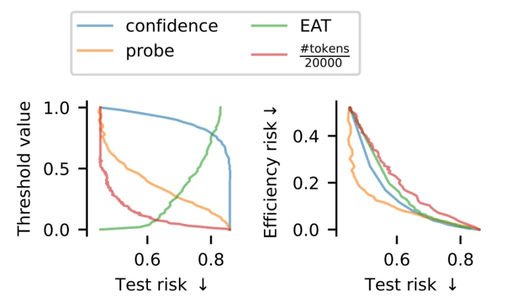

Fig 1 · Threshold for a target risk is signal-dependent — no single value transfers.

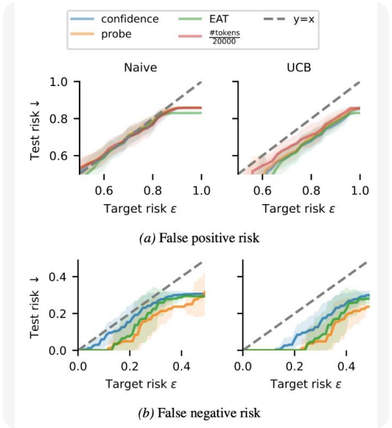

Fig 4 · UCB keeps realized risk ≤ target; Naïve violates it.

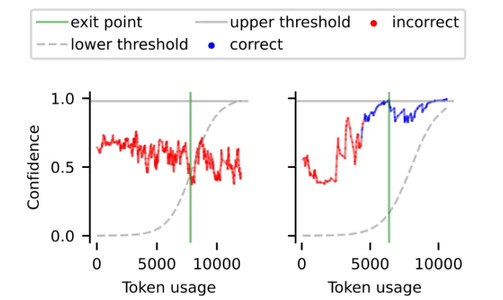

Fig 2 · Unsolvable runs exit via the lower threshold; solvable runs cross the upper.

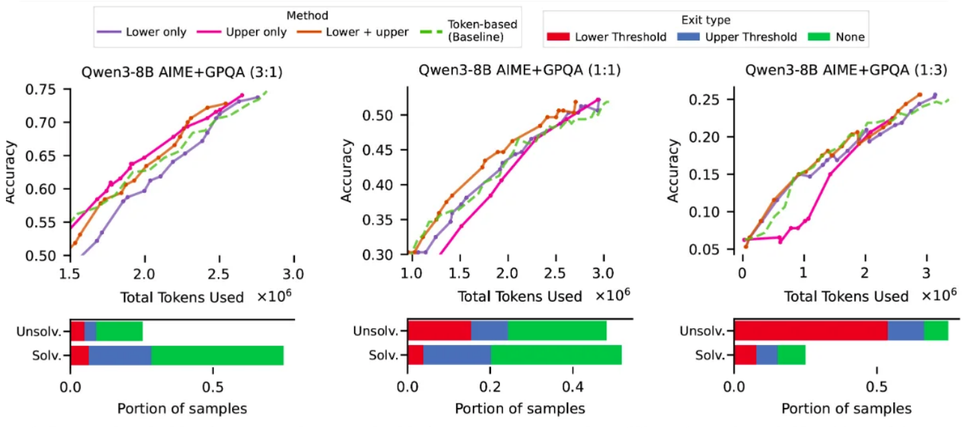

Fig 6 · The lower threshold shifts the accuracy-token curve left when unsolvable instances are common (1:1, 1:3).