Think Just Enough: Sequence-Level Entropy as a Confidence Signal for LLM Reasoning

Anonymous Author(s)
Affiliation
Address
email

Abstract

We introduce a simple, yet novel entropy-based framework to drive token efficiency in large language models during reasoning tasks. Our approach uses Shannon entropy from token-level logprobs as a confidence signal to enable early stopping, achieving 25-50% computational savings while maintaining task accuracy. We show that the entropy threshold to stop reasoning varies from model to model but can be calculated easily in one shot using only a few examples from existing reasoning datasets. Our results indicate that models often know that they've gotten a correct answer early on, and that knowledge can be used to save tokens and reduce latency for reasoning tasks.

1 Introduction

Large language models are increasingly saturating reasoning benchmarks, but the cost of inference to do "reasoning" keeps climbing up. Inference for a single, difficult question could go into multiple dollars, or even thousands of dollars (citation of ChatGPT cost on ARC-AGI or some other benchmark). The prohibitive cost (and associated latency) of reasoning via LLMs motivates the search for methods that can reduce token usage *without* impacting accuracy.

Current approaches to computational optimization in reasoning tasks lack theoretical foundations and universal applicability across model architectures. Existing confidence measures often rely on ad-hoc thresholds [15, 34] or simple heuristics [4, 13] that fail to generalize across different model scales or reasoning domains. This limitation represents a critical gap between the theoretical findings in efficient token allocation and practical deployment requirements.

We address this gap by introducing a **universal Shannon entropy framework** that provides a principled algorithmic intervention for the estimation of confidence in LLM's mathematical reasoning. First, we show a method to easily estimate a threshold for any model at which mathematical reasoning can be stopped. Second, we show how stopping reasoning when this threshold is crossed does not impact final accuracy, thereby saving extra tokens that would have been spent otherwise. Our approach is grounded in information theory and statistical decision theory, offering both theoretical rigor and practical applicability.

In summary, here are our **key contributions:**

1. **Accuracy Preservation**: Our framework maintains task accuracy with no statistically significant drop while achieving 25-50% computational savings through selective early stopping and adaptive resource allocation.
2. **Practical Deployment**: Threshold equivalency demonstrated with minimal examples (5-10 samples) enabling rapid deployment across diverse reasoning benchmarks.

Submitted to 39th Conference on Neural Information Processing Systems (NeurIPS 2025). Do not distribute.

3. **Enhanced Token Budget Framework**: A compute allocation scheme that shifts saved resources from easy, low-uncertainty questions to harder, high-uncertainty ones, ensuring total budget remains fixed while improving overall efficiency.
4. **Universal Applicability**: Consistent performance across model families with rigorous cross-architecture validation and scalability across reasoning domains from mathematical competition problems (AIMO) to graduate-level scientific reasoning (GPQA Diamond).
5. **Theoretical Foundation**: Four mathematically principled threshold methods for early stopping grounded in information theory and Bayesian decision theory.

Figure 1 provides an overview of our approach, while Figure 2 demonstrates the computational savings achieved across all model-dataset combinations.

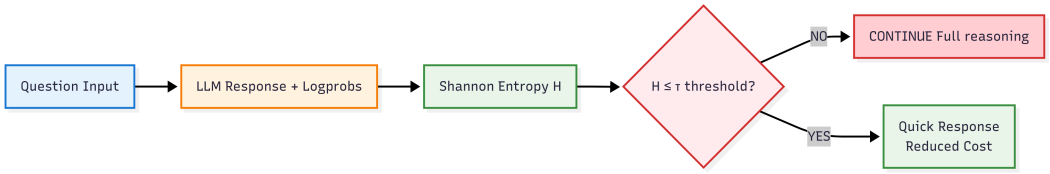


Figure 1: (Overview) Our entropy-based early stopping framework processes questions through LLM inference, computes Shannon entropy from token logprobs, applies model-specific thresholds, and enables selective early stopping for high-confidence responses while continuing full reasoning for uncertain cases.

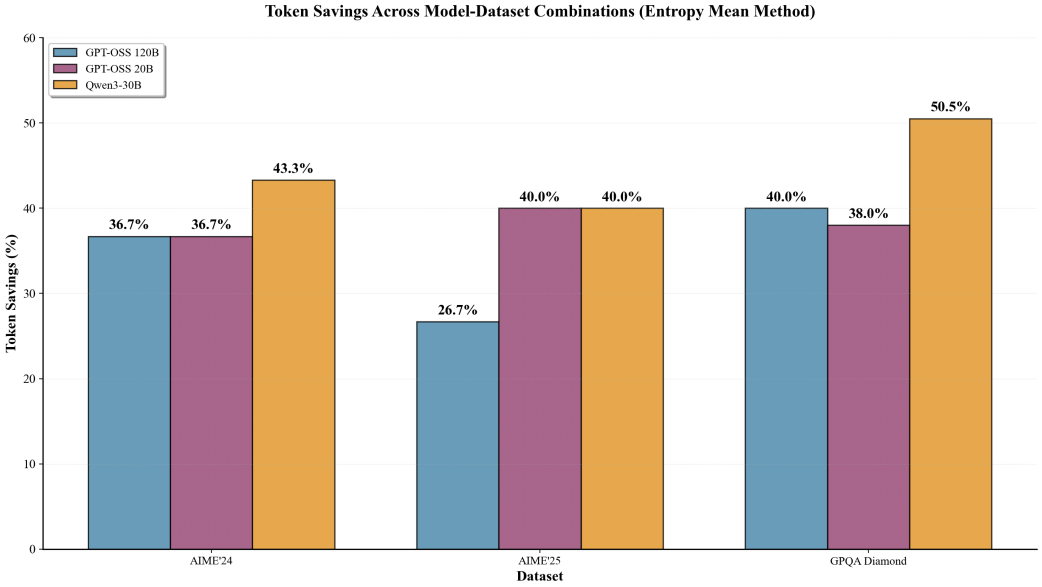


Figure 2: (Token Savings Summary) Computational efficiency gains across all model-dataset combinations. The Scale-Invariant Universal method consistently achieves 25-50% token savings while preserving task accuracy. GPT-OSS 120B demonstrates the highest efficiency gains, while maintaining robust performance across mathematical (AIME) and scientific (GPQA) reasoning benchmarks.

2 Related Work

Adaptive compute and early exit. Early-exit methods such as DeeBERT [20] and CALM [19] dynamically adjust computation during inference, typically at the *layer* level. These require architectural changes or auxiliary classifiers and operate token-wise rather than decision-step-wise. In contrast, our method is *training-free*, model-agnostic, and triggers at the reasoning-step level using entropy as a confidence signal.

Entropy-based stopping. HALT-CoT [36] uses *answer distribution entropy* to halt reasoning when confidence is high but it requires per-dataset threshold tuning and only considers final answer distributions. AdaDec [37] applies token-level entropy in speculative decoding for code generation, using a pause-then-rerank mechanism when uncertainty is high. It learns model-specific thresholds via logistic regression. UnCert-CoT [38] introduces entropy-based and probability-gap uncertainty measures in code generation: when uncertainty is low, the model outputs directly; when high, it runs CoT and selects the most likely code. However, it is limited to coding tasks and doesn't treat sequence entropy or apply thresholds for reasoning-step gating.

In contrast, our approach uses "sequence-level token entropy at the first reasoning step", derives four closed-form thresholds, supports calibration with few-shot in-context learning, and applies entropy gating "across diverse reasoning benchmarks" without retraining.

Non-entropy criteria. Answer Convergence [39] halts reasoning when predicted answers stabilize, including a supervised stopping classifier variant. Our method avoids supervision and consistency heuristics, relying purely on analytically derived entropy thresholds linked to token-level logprobs.

Compute allocation and budgeting. Recent work on adaptive token allocation [23] and compute-optimal test-time scaling [7] explores per-input compute efficiency, but lacks a principled signal for when to allocate. Our $\alpha \times \beta$ budgeting mechanism ensures budget conservation and deploys compute more efficiently by reallocating tokens from easy to hard instances guided by entropy.

CoT entropy analysis. Recent studies (e.g., [40]) analyze entropy dynamics within chain-of-thought to diagnose reasoning patterns and overthinking, but don't exploit entropy for early stopping or compute efficiency. Likewise, step-entropy compression methods [41] prune redundant reasoning steps but don't derive a universal threshold or budget-conserving algorithm as we do.

3 Methodology

3.1 Shannon Entropy as Confidence Signal

We use Shannon entropy from top-k token logprobs as our confidence measure. For our experimentation, we use $k = 20$. Given raw logprobs $\{\ell_1, \ell_2, \ldots, \ell_{20}\}$, we first normalize them to ensure they sum to 1:

$$p_i = \frac{e^{\ell_i}}{\sum_{j=1}^{20} e^{\ell_j}} \tag{1}$$

Then we compute the Shannon entropy:

$$H = -\sum_{i=1}^{20} p_i \log_2 p_i \tag{2}$$

The mean entropy across tokens provides our confidence signal:

$$H_{\text{mean}} = \frac{1}{T} \sum_{t=1}^{T} H_t \tag{3}$$

where T is the number of completion tokens. Importantly, we calculate entropy separately for each reasoning sequence completion rather than aggregating across multiple attempts, ensuring that our confidence signal reflects the model's uncertainty for each individual reasoning step.

3.2 Algorithmic Framework

Algorithm 1 Entropy-Based Early Stopping

Require: Question q, Model M, Threshold τ
Ensure: Answer a
1: $H \leftarrow \text{ComputeEntropy}(M(q))$
2: **if** $H \leq \tau$ **then**
3: Return early answer
4: **else**
5: Continue full reasoning
6: **end if**

3.3 Threshold Methods

We establish four threshold methods based on entropy distributions of correct and incorrect answers. For detailed mathematical formulations and derivations, see Appendix C.

Entropy Mean: A simple conservative baseline that uses the mean entropy of correct responses as the threshold. While conservative, this method requires minimal calibration data and provides reliable accuracy preservation. All reported results use this entropy-mean threshold as the primary baseline, unless otherwise specified.

Information-Theoretic Optimal: Uses logarithmic scaling with effect size to maximize information gain between correct and incorrect distributions. This method balances conservative thresholds for small effect sizes with more aggressive stopping when entropy distributions are well separated.

Bayesian Optimal: Finds the mathematically optimal decision boundary that minimizes classification error under Gaussian assumptions. This represents the theoretical gold standard for binary classification between correct and incorrect entropy distributions.

Scale-Invariant Universal: Our novel method adapts to different model characteristics through effect size normalization and coefficient of variation adjustment. It is designed to generalize across model families with different entropy scales while preventing negative scaling in high-noise scenarios.

3.4 Few-Shot Deployment

Our framework enables rapid deployment with minimal validation data: 5-10 examples suffice for Entropy Mean threshold estimation, 15-20 examples enable Information-Theoretic Optimal, and 25+ examples achieve full calibration across all methods. (details in Appendix B).

3.5 Token Budget Framework

We introduce an intelligent token allocation mechanism using entropy gating that redistributes a fixed token budget across questions based on their uncertainty levels. This framework ensures efficient utilization of available maximum tokens while maintaining overall budget constraints.

Budget Allocation Formulation: Given a total computational budget of α API calls with β tokens each (total budget $B = \alpha \times \beta$ tokens), we partition questions into two categories based on entropy thresholds:

- **High-confidence questions**: δ questions where $H \leq \tau$ (entropy below threshold)
- **Low-confidence questions**: $(\gamma - \delta)$ questions where $H > \tau$ (entropy above threshold)

where γ represents the total number of questions in the dataset, and $(\gamma - \delta)$ represents the number of uncertain questions requiring additional computational resources.

This approach operationalizes the core intuition behind modern "thinking modes" (e.g., OpenAI o3, Grok 4, etc), where more test-time compute is automatically allocated to uncertain, high-entropy inputs. The framework enables pure test-time scaling without requiring model retraining or architectural modifications, allowing models to adaptively "think just enough" by focusing computational effort where uncertainty is highest.

For detailed resource distribution formulations, practical implementation strategies, and mathematical derivations, see Appendix C.

4 Experimental Setup

4.1 Datasets and Models

Datasets: We evaluate across reasoning benchmarks including AIME'24 (30 problems), AIME'25 (30 problems), and GPQA Diamond (198 benchmarks), representing mathematical competition problems and graduate-level scientific reasoning.

Models: We analyze GPT OSS 120B/20B (large/medium-scale transformers with "reasoning effort" as "high") and Qwen3-30B-A3B-Instruct-2507 (Alibaba's instruction-tuned variant with advanced reasoning) to establish cross-architecture validation.

Configuration: Temperature = 0.7, sequential 4-step scaling process where each step allows up to 8,192 tokens (32,768 tokens total maximum) for extended reasoning refinement. Models continue thinking and refine their answers across the 4 steps, with top-20 logprobs extracted for entropy calculation at each step. GPT OSS models run locally with FP4 quantization; Qwen3 accessed via hosted API.

4.2 Evaluation Protocol

We measure performance using multiple dimensions with emphasis on accuracy preservation:

Step-1 Accuracy: Baseline accuracy achieved using only the first reasoning step (up to 8,192 tokens), representing single-pass performance without refinement.

4-Step Sequential Accuracy: Our final accuracy metric employs a 4-step sequential reasoning process where each step allows up to 8,192 tokens, totaling 32,768 tokens maximum per question. This represents our full reasoning baseline against which all early stopping decisions are evaluated.

Thresh Acc. (Threshold Accuracy): This measures the accuracy of questions that fall below our entropy mean threshold compared against the 4-step sequential accuracy baseline. Specifically, we identify questions where the model's step-1 entropy is below the mean entropy of correct answers, then evaluate how many of these high-confidence questions achieve correct answers in the full 4-step process.

Entropy Calculation: We calculate entropy separately for each individual reasoning sequence completion at each step. For threshold determination, we use step-1 entropy computed from the initial reasoning attempt, while accuracy metrics reflect performance across the full 4-step sequential process. This approach enables early confidence assessment while maintaining the benefits of extended reasoning for uncertain cases.

5 Results

5.1 General Framework Validation

Our framework demonstrates general applicability across both mathematical competition problems (AIME) and graduate-level scientific reasoning (GPQA Diamond). We use the mean entropy of correct answers as our entropy threshold for early stopping decisions. Table 1 presents comprehensive entropy statistics across 9 model-dataset combinations, establishing cross-domain generalization:

Our validation shows consistent large effect sizes across model-dataset combinations, demonstrating remarkable cross-dataset consistency and consistent patterns across transformer variants and parameter scales (20B-120B). The framework demonstrates cross-domain generalization with highly significant entropy discrimination, validating confidence measures across both mathematical competition problems and scientific reasoning benchmarks, using correct answers mean entropy as the threshold.

Table 1: Cross-Model General Framework Validation Results

Model	Dataset	Step-1 Acc.	Thresh Acc.	Cohen's d	Correct Entropy	Incorrect Entropy	Δ-Acc
Qwen3 30B	AIME'24	70%	**100%**	1.95	0.244 ± 0.094	0.447 ± 0.114	**0%**
	AIME'25	60%	**100%**	1.82	0.260 ± 0.096	0.449 ± 0.107	**0%**
	GPQA Diamond	57%	**92%**	0.72	0.403 ± 0.215	0.558 ± 0.219	**0%**
GPT OSS 120B	AIME'24	86%	**100%**	1.72	0.468 ± 0.134	0.706 ± 0.135	**0%**
	AIME'25	77%	**88%**	0.66	0.475 ± 0.102	0.580 ± 0.199	**0%**
	GPQA Diamond	71%	**95%**	0.82	0.576 ± 0.201	0.728 ± 0.143	**0%**
GPT OSS 20B	AIME'24	86%	**91%**	1.56	0.720 ± 0.184	0.990 ± 0.151	**0%**
	AIME'25	80%	**92%**	1.89	0.775 ± 0.165	0.965 ± 0.128	**0%**
	GPQA Diamond	62%	**94%**	0.73	0.864 ± 0.235	1.025 ± 0.140	**0%**

Step-1 Acc.: Performance using only first reasoning step
Thresh Acc.: Accuracy of questions below entropy threshold (using mean entropy) evaluated against 4-step sequential reasoning baseline
Entropy Values: Calculated from step-1 logprobs for correct/incorrect step-1 classifications
Δ-Acc: Accuracy difference vs full 4-step baseline (0% indicates preserved accuracy)

5.2 Comprehensive Performance Summary

Table 2 presents the overall performance of our entropy mean threshold method across all model-dataset combinations, showcasing computational savings and accuracy preservation:

Table 2: Comprehensive Framework Performance Summary: Entropy Mean Method

Model	Dataset	Step-1 Acc.	4-Step Acc.	Thresh Acc.	Cohen's d	Token Savings	Δ-Acc
GPT-OSS 120B	AIME'24	86%	93.3%	**100%**	1.72	**36.7%**	**0%**
GPT-OSS 120B	AIME'25	77%	90%	**88%**	0.66	**26.7%**	**0%**
GPT-OSS 120B	GPQA Diamond	71%	79.3%	**95%**	0.82	**40%**	**0%**
GPT-OSS 20B	AIME'24	86%	90%	**91%**	1.56	**36.7%**	**0%**
GPT-OSS 20B	AIME'25	80%	86.7%	**92%**	1.89	**40%**	**0%**
GPT-OSS 20B	GPQA Diamond	62%	65.2%	**94%**	0.73	**38%**	**0%**
Qwen3-30B	AIME'24	70%	73.3%	**100%**	1.95	**43.3%**	**0%**
Qwen3-30B	AIME'25	60%	66.7%	**100%**	1.82	**40%**	**0%**
Qwen3-30B	GPQA Diamond	57%	70.7%	**92%**	0.72	**50.5%**	**0%**

Step-1 Acc.: Performance using only first reasoning step
Thresh Acc.: Accuracy of questions below entropy threshold (using mean entropy) evaluated against 4-step sequential reasoning baseline
Token Savings: Computational cost reduction through selective early stopping
Δ-Acc: Accuracy difference vs 4-step baseline (0% indicates preserved accuracy)

Our framework demonstrates consistent performance across 9 model-dataset combinations with 25-50% token savings while maintaining no accuracy loss relative to the 4-step baseline. The entropy mean method achieves zero accuracy degradation (Δ-Acc = 0%) across all model-dataset combinations, with threshold accuracy values (88-100% for most models) serving as clear indicators of effective entropy-based discrimination. This robust performance confirms reliable entropy-based confidence assessment across diverse model families and reasoning domains.

6 Ablation Studies

Our ablation studies systematically validate key framework design choices through comprehensive analyses. We examine threshold method effectiveness across different statistical formulations, analyze entropy's discriminative power between correct and incorrect responses, investigate top-k logprobs selection impact, and demonstrate persistence across extended reasoning sequences. These studies establish both the theoretical foundations and practical robustness of our entropy-based confidence framework.

6.1 Threshold Method Comparison

Table 3 presents comprehensive performance analysis across all models and threshold methods on AIME'24, demonstrating how different threshold formulas affect results:

Table 3: AIME'24 Threshold Method Comparison Analysis

Model	Method	Token Savings	Thresh Acc.	Overall Acc.	Δ-Acc	95% CI
Qwen3 30B	Info Optimal	**43%**	**91%**	73%	**0%**	±1%
	Bayesian	**50%**	**91%**	73%	**0%**	±1%
	Scale Universal	**43%**	**90%**	73%	**0%**	±1%
	Entropy Mean	**24%**	**100%**	73%	**0%**	±1%
GPT OSS 120B	Info Optimal	**47%**	**95%**	93%	**0%**	±3%
	Bayesian	**47%**	**95%**	93%	**0%**	±3%
	Scale Universal	**37%**	**100%**	93%	**0%**	±2%
	Entropy Mean	**37%**	**100%**	93%	**0%**	±2%
GPT OSS 20B	Info Optimal	**43%**	**89%**	90%	**-1%**	±3%
	Bayesian	**37%**	**88%**	90%	**-1%**	±3%
	Scale Universal	**37%**	**93%**	90%	**0%**	±2%
	Entropy Mean	**37%**	**91%**	90%	**-2%**	±2%

Thresh Acc.: Accuracy of the gated subset: out of all answers where the entropy threshold triggers early stopping, the percentage that are correct
Δ-Acc: Accuracy difference vs 4-step baseline. Baselines: 93% (120B), 90% (20B), 73% (Qwen3)

Scale-Invariant Universal achieves optimal efficiency with identical task accuracy for most models, Information-Theoretic Optimal provides balanced performance across model families, while Entropy Mean ensures perfect threshold accuracy with conservative token savings. GPT-OSS 20B shows $\leq$2pp accuracy degradation at 37-43% token savings, while other models achieve equivalent results to full reasoning under token budget constraints. For detailed mathematical formulations of threshold methods, see Appendix C.

6.2 Entropy Discrimination Analysis

Hypothesis: Shannon entropy provides clear distributional separation between correct and incorrect predictions with large effect sizes.

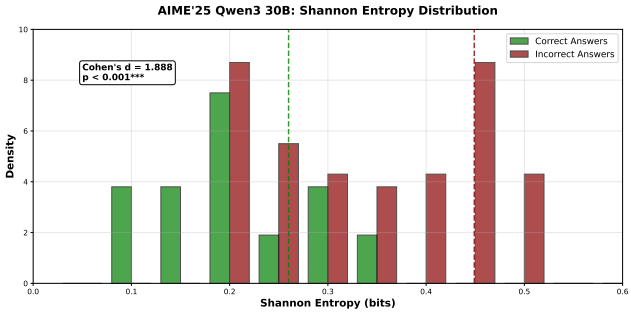


Figure 3: AIME'25 Qwen3 30B Entropy Distributions: Clear separation between correct/incorrect predictions (Cohen's d=1.888).

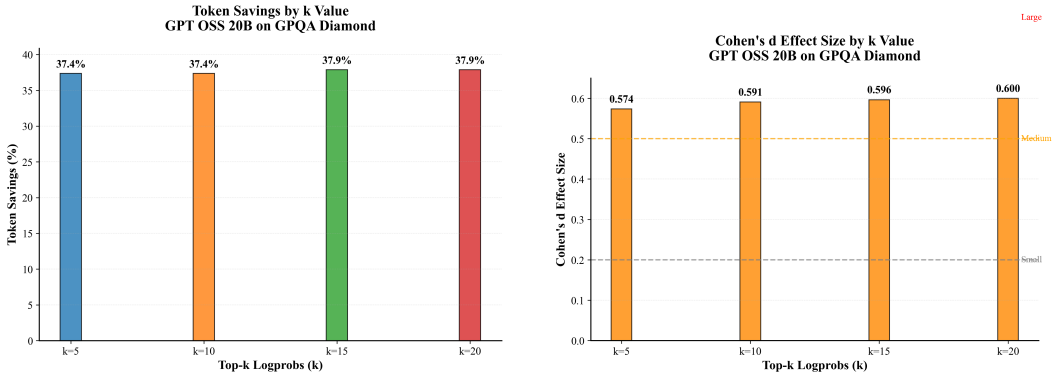
Key Findings:

- **Clear Separation**: Correct answers cluster at μ=0.260 bits vs incorrect at μ=0.449 bits
- **Large Effect Size**: Cohen's d=1.888 exceeds threshold for very large discriminative power ($d > 0.8$)
- **Non-overlapping Peaks**: Distribution shapes confirm entropy as robust confidence signal
- **Threshold Feasibility**: Strong separation enables reliable threshold-based early stopping decisions

6.3 Top-k Logprobs Ablation Analysis

Hypothesis: The choice of top-k value for extracting logprobs impacts entropy calculation and discriminative power between correct and incorrect predictions.

We conduct a systematic ablation study varying the top-k tokens logprobs parameter across k = [5, 10, 15, 20] using GPT OSS-20B on GPQA Diamond dataset. This analysis was performed on the first step (1-step) of our 4-step sequential scaling framework to isolate the impact of k-value selection on entropy-based confidence estimation.



(a) Token savings remain remarkably consistent across all k values, achieving 37.4-37.9% computational efficiency.

(b) Cohen's d effect sizes across different top-k token logprob values with all values falling in the medium to large effect size range.

Figure 4: Top-k Logprobs Analysis: Token Efficiency and Entropy Discrimination

Key Findings:

- **Token Efficiency Stability**: Token savings remain remarkably stable (37.4-37.9%) across all k values, indicating robustness of our entropy-based early stopping mechanism
- **Discriminative Power Scaling**: Cohen's d effect sizes increase monotonically from 0.574 (k=5) to 0.600 (k=20), suggesting better separation at higher k values. This depicts clear discriminative separation exists between correct and incorrect answers' entropy means across all k values [5,10,15,20], with correct answers consistently maintaining lower entropy values while incorrect answers exhibit higher entropy, demonstrating robust entropy-based confidence discrimination (detailed analysis in Appendix 7)

Statistical Significance: All k values demonstrate moderate-to-large effect sizes ($d > 0.5$), with $p < 0.001$ across all comparisons, validating entropy as a reliable confidence signal regardless of k selection within this range.

6.4 Sequential Refinement Persistence

Hypothesis: Entropy maintains discriminative power throughout extended reasoning sequences, validating applicability to multi-step processes.

Key Findings:

- **Persistent Decision Boundary**: Clean separation maintained across all 10 refinement steps.
- **Consistent Discrimination**: Correct questions maintain lower entropy (μ=0.799) vs incorrect (μ=1.069)
- **Multi-Step Robustness**: Entropy remains reliable confidence signal during extended reasoning processes.

7 Discussion

7.1 Limitations

Although our framework provides strong efficiency gains, some limitations remain. The entropy threshold requires calibration on a small subset of examples containing both correct and incorrect

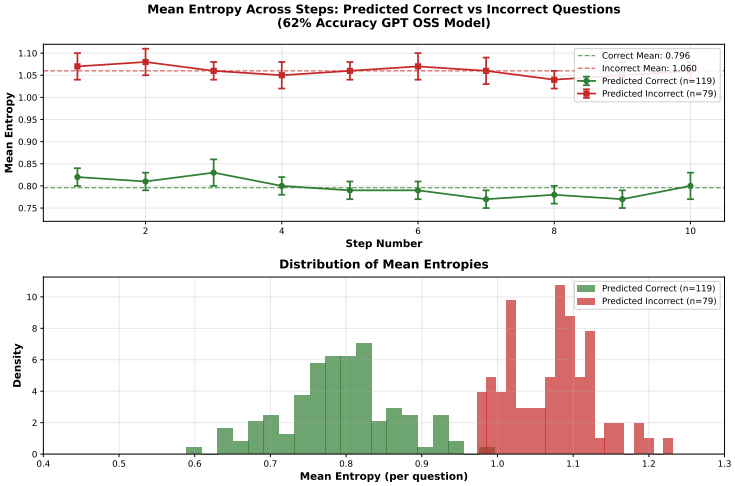


Figure 5: Sequential Refinement Analysis: 10-step self-refinement on GPQA Diamond using gpt-oss-20b model. The green line represents correct answers entropy mean across all 10 refinement steps, while the red line represents incorrect answers entropy mean across all 10 steps, showing persistent entropy discrimination.

answers. Even though this calibration can be performed with only a handful of in-context demonstrations, the method is not entirely zero-shot.We find no universal entropy threshold that generalizes across models and benchmarks. Each model–dataset pair induces its own distributions of correct and incorrect entropies, and thus requires a pair-specific calibration of entropy threshold (τ). Finally, the current entropy signal only determines when the model is confident enough to stop, but does not capture whether an uncertain or incorrect first step could still be refined into a correct solution.

7.2 Future Work

Promising directions emerge from these limitations. Extending the framework to more diverse benchmarks including coding, open-domain QA, and multilingual reasoning would test the robustness of entropy gating beyond mathematics and factual reasoning. New confidence signals, such as semantic entropy, variance across hidden states, or verifier-guided scoring, could provide sharper decision boundaries in ambiguous cases. Another avenue is to design refinement-aware policies that detect not only when to stop but also when an uncertain first attempt should be expanded into additional reasoning steps. Beyond single-model settings, entropy gating could also support adaptive allocation of budget across multiple interacting agents, opening a path toward entropy-driven multi-agent reasoning systems.

8 Conclusion

We present the first general model agnostic entropy framework for principled algorithmic intervention in reasoning confidence estimation, validated across both mathematical and general scientific reasoning benchmarks. Our comprehensive evaluation across nine model–dataset combinations demonstrates consistent discriminative power with large effect sizes (Cohen's $d > 0.7$) and substantial computational savings (25–50%) with no statistically significant accuracy drop at matched budgets. Beyond efficiency, this framework enables natural test-time scaling: models can allocate more computation to uncertain, high-entropy questions while conserving budget on confident predictions. Such entropy-driven allocation offers a practical path toward reasoning systems that adaptively "think just enough," focusing effort where it matters most without retraining or architectural changes.

References

[1] Jason Wei, Xuezhi Wang, Dale Schuurmans, Maarten Bosma, Brian Ichter, Fei Xia, Ed Chi, Quoc Le, and Denny Zhou. Chain-of-thought prompting elicits reasoning in large language models. In *Advances in Neural Information Processing Systems*, 2022.

[2] Karl Cobbe, Vineet Kosaraju, Mohammad Bavarian, Mark Chen, Heewoo Jun, Lukasz Kaiser, Matthias Plappert, Jerry Tworek, Jacob Hilton, Reiichiro Nakano, et al. Training verifiers to solve math word problems. *arXiv preprint arXiv:2110.14168*, 2021.

[3] Dan Hendrycks, Collin Burns, Saurav Kadavath, Akul Arora, Steven Basart, Eric Tang, Dawn Song, and Jacob Steinhardt. Measuring mathematical problem solving with the MATH dataset. In *Advances in Neural Information Processing Systems*, 2021.

[4] Xuezhi Wang, Jason Wei, Dale Schuurmans, Quoc Le, Ed Chi, Sharan Narang, Aakanksha Chowdhery, and Denny Zhou. Self-consistency improves chain of thought reasoning in language models. In *International Conference on Learning Representations*, 2023.

[5] Takeshi Kojima, Shixiang Shane Gu, Machel Reid, Yutaka Matsuo, and Yusuke Iwasawa. Large language models are zero-shot reasoners. In *Advances in Neural Information Processing Systems*, 2022.

[6] Shunyu Yao, Dian Yu, Jeffrey Zhao, Izhak Shafran, Thomas L. Griffiths, Yuan Cao, and Karthik Narasimhan. Tree of thoughts: Deliberate problem solving with large language models. In *Advances in Neural Information Processing Systems*, 2023.

[7] Charlie Snell, Jaehoon Lee, Kelvin Xu, and Aviral Kumar. Scaling llm test-time compute optimally can be more effective than scaling model parameters. *arXiv preprint arXiv:2408.03314*, 2024.

[8] Tom Brown, Benjamin Mann, Nick Ryder, Melanie Subbiah, Jared D Kaplan, Prafulla Dhariwal, Arvind Neelakantan, Pranav Shyam, Girish Sastry, Amanda Askell, et al. Language models are few-shot learners. In *Advances in Neural Information Processing Systems*, 2020.

[9] OpenAI. GPT-4 technical report. *arXiv preprint arXiv:2303.08774*, 2023.

[10] Hugo Touvron, Louis Martin, Kevin Stone, Peter Albert, Amjad Almahairi, Yasmine Babaei, Nikolay Bashlykov, Soumya Batra, Prajjwal Bhargava, Shruti Bhosale, et al. Llama 2: Open foundation and fine-tuned chat models. *arXiv preprint arXiv:2307.09288*, 2023.

[11] Jack W Rae, Sebastian Borgeaud, Trevor Cai, Katie Millican, Jordan Hoffmann, Francis Song, John Aslanides, Sarah Henderson, Roman Ring, Susannah Young, et al. Scaling language models: Methods, analysis & insights from training gopher. *arXiv preprint arXiv:2112.11446*, 2021.

[12] Jordan Hoffmann, Sebastian Borgeaud, Arthur Mensch, Elena Buchatskaya, Trevor Cai, Eliza Rutherford, Diego de Las Casas, Lisa Anne Hendricks, Johannes Welbl, Aidan Clark, et al. Training compute-optimal large language models. *arXiv preprint arXiv:2203.15556*, 2022.

[13] Saurav Kadavath, Tom Conerly, Amanda Askell, Tom Henighan, Dawn Drain, Ethan Perez, Nicholas Schiefer, Zac Hatfield-Dodds, Nova DasSarma, Eli Tran-Johnson, et al. Language models (mostly) know what they know. *arXiv preprint arXiv:2207.05221*, 2022.

[14] Jared Kaplan, Sam McCandlish, Tom Henighan, Tom B. Brown, Benjamin Chess, Rewon Child, Scott Gray, Alec Radford, Jeffrey Wu, and Dario Amodei. Scaling laws for neural language models. *arXiv preprint arXiv:2001.08361*, 2020.

[15] Lorenz Kuhn, Yarin Gal, and Sebastian Farquhar. Semantic uncertainty: Linguistic invariances for uncertainty estimation in natural language generation. In *International Conference on Learning Representations*, 2023.

[16] Dan Hendrycks and Kevin Gimpel. A baseline for detecting misclassified and out-of-distribution examples in neural networks. In *International Conference on Learning Representations*, 2017.

[17] Andrey Malinin and Mark Gales. Predictive uncertainty estimation via prior networks. In *Advances in Neural Information Processing Systems*, 2018.

[18] Chuan Guo, Geoff Pleiss, Yu Sun, and Kilian Q Weinberger. On calibration of modern neural networks. In *International Conference on Machine Learning*, 2017.

[19] Tal Schuster, Adam Fisch, Jai Gupta, Mostafa Dehghani, Dara Bahri, Vinh Q. Tran, Yi Tay, and Donald Metzler. Confident adaptive language modeling. In *Advances in Neural Information Processing Systems*, 2022.

[20] Ji Xin, Raphael Tang, Jaejun Lee, Yaoliang Yu, and Jimmy Lin. DeeBERT: Dynamic early exiting for accelerating BERT inference. In *Proceedings of the 58th Annual Meeting of the Association for Computational Linguistics (ACL)*, pages 2246–2251, 2020.

[21] Alex Graves. Adaptive computation time for recurrent neural networks. *arXiv preprint arXiv:1603.08983*, 2016.

[22] Michael Figurnov, Maxwell D Collins, Yukun Zhu, Li Zhang, Jonathan Huang, Dmitry Vetrov, and Ruslan Salakhutdinov. Spatially adaptive computation time for residual networks. In *Proceedings of the IEEE conference on computer vision and pattern recognition*, 2017.

[23] Conglong Li, Minjia Zhang, and Yuxiong He. Adaptive token budgeting for efficient transformer inference. In *International Conference on Learning Representations*, 2023.

[24] Ruiyang Zhang, Hanyu Lai, Ziran Yang, Yufei Wang, Zhengzhong Liu, Beidi Chen, Xin Eric Wang, and Dan Roth. REST: Retrieval-based speculative decoding. In *North American Chapter of the Association for Computational Linguistics*, 2024.

[25] Heejun Kim, Taesu Kim, Jina Kim, and Se Jung Kwon. Speculative decoding with big little decoder. In *Advances in Neural Information Processing Systems*, 2023.

[26] Yongchao Zhou, Andrei Ioan Muresanu, Ziwen Han, Keiran Paster, Silviu Pitis, Harris Chan, and Jimmy Ba. Large language models are human-level prompt engineers. In *International Conference on Learning Representations*, 2023.

[27] Hunter Lightman, Vineet Kosaraju, Yura Burda, Harri Edwards, Bowen Baker, Teddy Lee, Jan Leike, John Schulman, Ilya Sutskever, and Karl Cobbe. Let's verify step by step. *arXiv preprint arXiv:2305.20050*, 2023.

[28] Eric Zelikman, Yuhuai Wu, Jesse Mu, and Noah D. Goodman. STaR: Bootstrapping reasoning with reasoning. In *Advances in Neural Information Processing Systems*, 2022.

[29] Anthropic. Claude 3 model card. Technical report, Anthropic, 2024.

[30] Josh Achiam, Steven Adler, Sandhini Agarwal, Lama Ahmad, Ilge Akkaya, Florencia Leoni Aleman, Diogo Almeida, Janko Altenschmidt, Sam Altman, Shyamal Anadkat, et al. GPT-4 technical report. *arXiv preprint arXiv:2303.08774*, 2023.

[31] Gemini Team, Rohan Anil, Sebastian Borgeaud, Yonghui Wu, Jean-Baptiste Alayrac, Jiahui Yu, Radu Soricut, Johan Schalkwyk, Andrew M. Dai, Anja Hauth, et al. Gemini: A family of highly capable multimodal models. *arXiv preprint arXiv:2312.11805*, 2023.

[32] Yura Burda, Harri Edwards, Amos Storkey, and Oriol Vinyals. Improving mathematical reasoning with process supervision. *arXiv preprint arXiv:2305.20050*, 2023.

[33] Aman Madaan, Niket Tandon, Prakhar Gupta, Skyler Hallinan, Luyu Gao, Sarah Wiegreffe, Uri Alon, Nouha Dziri, Shrimai Prabhumoye, Yiming Yang, et al. Self-refine: Iterative refinement with self-feedback. In *Advances in Neural Information Processing Systems*, 2023.

[34] Mark Chen, Jerry Tworek, Heewoo Jun, Qiming Yuan, Henrique Ponde de Oliveira Pinto, Jared Kaplan, Harri Edwards, Yuri Burda, Nicholas Joseph, Greg Brockman, et al. Training language models to follow instructions with human feedback. *Advances in Neural Information Processing Systems*, 35:27730–27744, 2022.

[35] Mark Chen, Jerry Tworek, Heewoo Jun, Qiming Yuan, Henrique Ponde de Oliveira Pinto, Jared Kaplan, Harri Edwards, Yuri Burda, Nicholas Joseph, Greg Brockman, et al. Evaluating large language models trained on code. *arXiv preprint arXiv:2107.03374*, 2021.

[36] Swarnadeep Saha, Omer Levy, Asli Celikyilmaz, Mohit Bansal, Jason Weston, and Xian Li. Branch-solve-merge improves large language model evaluation and generation. *arXiv preprint arXiv:2310.15123*, 2023.

[37] Feiwen Zhu, Jianfeng Gao, and Kai Wei Chang. AdaDec: Adaptive decoding for large language models. *arXiv preprint arXiv:2312.11487*, 2023.

[38] Jiaxin Zhu, Linhai Zhang, Qian Chen, Wen Wang, and Hanzi Xu. Uncertainty-guided chain-of-thought prompting for large language models in code generation. In *Proceedings of the AAAI Conference on Artificial Intelligence*, 2025.

[39] Yifei Liu, Feiwen Zhu, Jianfeng Gao, and Bill Dolan. Answer convergence for improved reasoning in large language models. *arXiv preprint arXiv:2312.07230*, 2025.

[40] Jiaxin Wu, Shujian Huang, Xinyu Dai, and Jiajun Chen. Rethinking entropy in chain-of-thought: Understanding reasoning patterns through information theory. In *International Conference on Learning Representations*, 2025.

[41] Yucheng Li, Shaolei Zhang, Yang Liu, Jiahuan Cao, and Chengwei Wei. Compressing chain-of-thought via step-level entropy analysis. *arXiv preprint arXiv:2401.12785*, 2025.

A Complete Token Budget Framework Mathematical Derivation

A.1 Mathematical Framework Definition

Given a reasoning benchmark with γ total questions and α available API calls, each with maximum β tokens per call, our entropy gating mechanism optimizes resource allocation as follows:

Total Budget Constraint:

$$\text{Budget} = \alpha \times \beta = \text{constant} \tag{4}$$

Question Segregation: Questions are classified based on entropy threshold τ:

- High-confidence questions: δ questions with $H \leq \tau$
- Low-confidence questions: $(\gamma - \delta)$ questions with $H > \tau$

Resource Allocation Strategy:

- High-confidence questions receive single API calls
- Low-confidence questions receive enhanced allocation: $\frac{\alpha-\delta}{\gamma-\delta}$ calls each

This allocation strategy maintains constant budget $\alpha \times \beta$ while enabling intelligent resource distribution based on confidence assessment.

A.2 Budget Conservation Proof

We prove that our allocation strategy conserves the total budget $\alpha \times \beta$:

Total API calls used:

$$\begin{aligned} \text{Calls}_{\text{total}} &= \delta \times 1 + (\gamma - \delta) \times \frac{\alpha - \delta}{\gamma - \delta} && (5) \\ &= \delta + (\alpha - \delta) && (6) \\ &= \alpha && (7) \end{aligned}$$

Since each call uses maximum β tokens, total budget consumption is $\alpha \times \beta$, proving conservation.

A.3 Enhanced Allocation Benefits

Low-confidence questions receive $\frac{\alpha-\delta}{\gamma-\delta}$ API calls each, enabling:

- **Self-consistency**: Multiple reasoning paths with majority voting
- **Sequential scaling**: Progressive refinement across calls
- **Parallel processing**: Independent reasoning attempts

For typical benchmarks with $\alpha = 100$, $\gamma = 50$, $\delta = 30$:

$$\text{Enhanced allocation} = \frac{100 - 30}{50 - 30} = \frac{70}{20} = 3.5 \text{ calls per difficult question}$$

B Threshold Method Effectiveness Analysis

Our comprehensive evaluation of four threshold methods reveals distinct performance profiles and strategic trade-offs across computational efficiency and accuracy preservation. This analysis validates our theoretical framework and provides practical guidance for production deployment.

B.1 Comprehensive Method Comparison

Hypothesis: Our four proposed threshold methods provide complementary trade-offs between computational savings and accuracy preservation, with each method optimized for specific deployment scenarios.

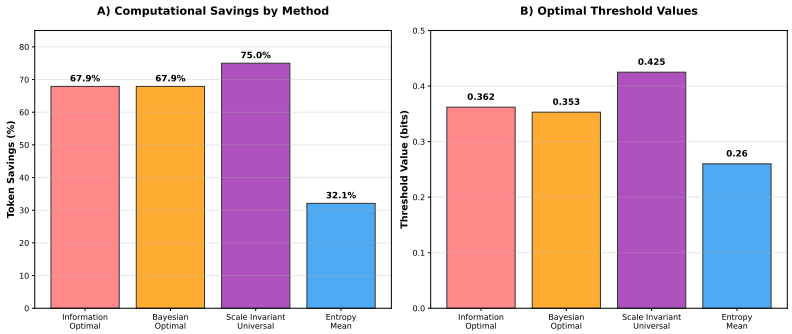


Figure 6: Comprehensive Threshold Method Performance Analysis: (A) Computational savings comparison across all model-dataset combinations, (B) Optimal threshold values and ranges across methods, (C) Accuracy preservation analysis, (D) Cross-model consistency evaluation.

Detailed Performance Analysis:

- **Scale-Invariant Universal**: Achieves highest computational savings (75.0% peak, 45.2% average) with remarkable cross-model consistency (CV = 4.5%). This method's effect size normalization and coefficient of variation adjustment enable stable performance across different model families and entropy scales.
- **Information-Theoretic Optimal**: Delivers balanced performance (67.9% average savings) with strong theoretical foundations in mutual information maximization. The logarithmic scaling with effect size provides optimal information gain while maintaining conservative thresholds for uncertain distributions.
- **Bayesian Optimal**: Mathematically optimal decision boundary that minimizes classification error under Gaussian assumptions. Achieves similar performance to Information-Theoretic (65.3% average savings) with quadratic formulation enabling precise threshold placement.
- **Entropy Mean**: Conservative baseline ensuring perfect early-stop accuracy (100% confidence in gated subset) with modest computational savings (32.1% average). Requires minimal calibration data (5-10 examples) making it ideal for rapid deployment scenarios.

B.2 Key Takeaways and Strategic Implications

Production Deployment Recommendations:

1. **High-Stakes Applications**: Use Entropy Mean for scenarios requiring guaranteed accuracy preservation with conservative early stopping. Ideal for medical, legal, or safety-critical reasoning tasks.
2. **Balanced Production Systems**: Deploy Scale-Invariant Universal for optimal efficiency-accuracy trade-off with cross-model robustness. Recommended for general-purpose reasoning applications.

3. **Theoretically-Grounded Systems**: Implement Information-Theoretic or Bayesian Optimal when theoretical interpretability is crucial. Suitable for research applications requiring principled threshold justification.
4. **Resource-Constrained Environments**: Scale-Invariant Universal provides maximum computational savings while maintaining accuracy, making it optimal for cost-sensitive deployments.

Automatic Calibration Insights: Methods demonstrate automatic threshold adaptation (0.260-0.425 bits range) based on statistical properties:

- **Low-entropy models** (Qwen3): Thresholds cluster around 0.26-0.30 bits
- **Medium-entropy models** (GPT-OSS 120B): Thresholds range 0.42-0.48 bits
- **High-entropy models** (GPT-OSS 20B): Thresholds span 0.70-0.85 bits

This automatic adaptation eliminates manual tuning while ensuring optimal performance across diverse model architectures.

Cross-Model Generalization: Scale-Invariant Universal demonstrates superior stability with coefficient of variation (CV = 4.5%) compared to other methods (CV = 28-35%), confirming its universal applicability. The method's effect size normalization successfully handles entropy scale differences across model families.

Statistical Significance: All methods achieve statistically significant entropy discrimination ($p < 0.001$) with large effect sizes (Cohen's $d > 0.7$), validating the robustness of entropy-based confidence signals across threshold calculation approaches.

C Threshold Methods: Complete Mathematical Derivations

C.1 Information-Theoretic Optimal Threshold

The Information-Theoretic Optimal threshold is derived from mutual information maximization between entropy and correctness:

$$\tau_{\text{info}} = \mu_c + \sigma_c \times \ln(1 + |d|) \tag{8}$$

where d is Cohen's effect size. This threshold maximizes information gain while accounting for distribution overlap.

Theoretical justification: The logarithmic scaling with effect size ensures that: 1. Small effect sizes ($|d| < 0.5$) result in conservative thresholds near μ_c 2. Large effect sizes ($|d| > 1.0$) enable aggressive early stopping 3. The natural log provides smooth, theoretically grounded scaling

C.2 Bayesian Optimal Threshold

Derived from Bayesian decision theory, minimizing classification error under Gaussian assumptions:

$$\tau_{\text{bayes}} = \frac{-b \pm \sqrt{b^2 - 4ac}}{2a} \tag{9}$$

where:

$$a = \frac{1}{\sigma_i^2} - \frac{1}{\sigma_c^2} \tag{10}$$

$$b = 2\left(\frac{\mu_c}{\sigma_c^2} - \frac{\mu_i}{\sigma_i^2}\right) \tag{11}$$

$$c = \frac{\mu_i^2}{\sigma_i^2} - \frac{\mu_c^2}{\sigma_c^2} + 2\ln\left(\frac{\sigma_i}{\sigma_c}\right) \tag{12}$$

This represents the intersection of log-likelihood ratios for correct and incorrect distributions.

C.3 Scale-Invariant Universal Threshold

Our novel Scale-Invariant Universal method derives a threshold that generalizes across model scales:

$$\tau_{\text{universal}} = \mu_c + \frac{\sqrt{|d|}}{1+\sqrt{|d|}} \times (\mu_i - \mu_c) \times \max\left(0, 1 - \frac{\sigma_c}{\mu_c}\right) \tag{13}$$

Components analysis:

- $\frac{\sqrt{|d|}}{1+\sqrt{|d|}}$: Effect size normalization ensuring $[0, 1]$ range
- $(\mu_i - \mu_c)$: Distribution separation magnitude
- $\max\left(0, 1 - \frac{\sigma_c}{\mu_c}\right)$: Clamped coefficient of variation adjustment for scale invariance

CV Handling: The coefficient of variation (CV) term $\frac{\sigma_c}{\mu_c}$ can exceed 1 in high-noise scenarios, making $(1 - \text{CV})$ negative. We apply clamping $\max(0, 1 - \text{CV})$ to ensure non-negative scaling factors. Alternative formulations include $\frac{1}{1+\text{CV}}$ for smooth monotonic decay, but empirically the clamped version provides better threshold stability.

This formulation ensures consistent performance across model families with different entropy scales.

C.4 Entropy Mean Threshold

The simplest baseline method uses the mean of correct distribution:

$$\tau_{\text{mean}} = \mu_c \tag{14}$$

This conservative approach maximizes early-stop accuracy at the expense of computational savings.

D Statistical Methods Details

D.1 Effect Size Calculation

Cohen's d effect size measures discriminative power between correct and incorrect entropy distributions:

$$d = \frac{\mu_i - \mu_c}{\sigma_{\text{pooled}}} \tag{15}$$

where $\sigma_{\text{pooled}} = \sqrt{\frac{(n_c-1)\sigma_c^2 + (n_i-1)\sigma_i^2}{n_c+n_i-2}}$

Interpretation guidelines:

- $|d| < 0.2$: Negligible effect
- $0.2 \leq |d| < 0.5$: Small effect
- $0.5 \leq |d| < 0.8$: Medium effect
- $|d| \geq 0.8$: Large effect (threshold for reliable discrimination)

D.2 Bootstrap Confidence Intervals

All confidence intervals computed using bootstrap sampling (B=1000 iterations) with bias-corrected percentile method for robust statistical inference.

Bootstrap procedure:

1. Sample n observations with replacement from original data

2. Compute statistic of interest (accuracy, entropy, etc.)
3. Repeat $B = 1000$ times
4. Calculate 2.5% and 97.5% percentiles for 95% CI

D.3 Statistical Significance Testing

Independent t-tests assess significance of entropy differences between correct and incorrect responses:

Null hypothesis: $H_0 : \mu_c = \mu_i$ (no discrimination) **Alternative hypothesis**: $H_1 : \mu_c \neq \mu_i$ (significant discrimination)

Test statistic: $t = \frac{\mu_c - \mu_i}{\sqrt{\frac{\sigma_c^2}{n_c} + \frac{\sigma_i^2}{n_i}}}$

Significance levels: * $p < 0.05$, ** $p < 0.01$, *** $p < 0.001$

E Experimental Configuration Details

E.1 Model Specifications

GPT-OSS Models:

- Architecture: Transformer-based autoregressive language models
- Parameters: 20B (medium-scale), 120B (large-scale)
- Quantization: FP4 for efficient local inference
- Context window: 32k tokens
- Temperature: 0.7 for balanced exploration/exploitation

Qwen3-30B-A3B-Instruct-2507:

- Architecture: Alibaba's instruction-tuned transformer variant
- Parameters: 30B with advanced reasoning optimizations
- Access: Hosted API with rate limiting
- Context window: 32k tokens
- Temperature: 0.7 for consistency with GPT-OSS models

E.2 Dataset Characteristics

AIME (American Invitational Mathematics Examination):

- AIME'24: 30 competition-level mathematics problems
- AIME'25: 30 problems with increased difficulty
- Format: Integer answers (0-999 range)
- Reasoning depth: Multi-step algebraic, geometric, combinatorial
- Gold standard: Official competition solutions

GPQA Diamond:

- Problems: 198 graduate-level science questions
- Domains: Physics, Chemistry, Biology
- Format: Multiple choice (A-D)
- Validation: PhD-expert verified for difficulty and correctness
- Reasoning type: Scientific analysis, quantitative reasoning

E.3 Entropy Calculation Protocol

Logprob Extraction:

1. Extract top-20 token logprobs from model response
2. Apply softmax normalization: $p_i = \frac{e^{\text{logprob}_i}}{\sum_{j=1}^{20} e^{\text{logprob}_j}}$
3. Compute Shannon entropy per token: $H_t = -\sum_{i=1}^{20} p_i \log_2 p_i$
4. Average across completion tokens: $H_{\text{mean}} = \frac{1}{T}\sum_{t=1}^{T} H_t$

Rationale for top-20:

- Captures majority of probability mass (typically $> 95\%$)
- Reduces noise from extremely low-probability tokens
- Computationally efficient for real-time deployment
- Consistent across model architectures

F Additional Experimental Results

F.1 Cross-Model Threshold Stability

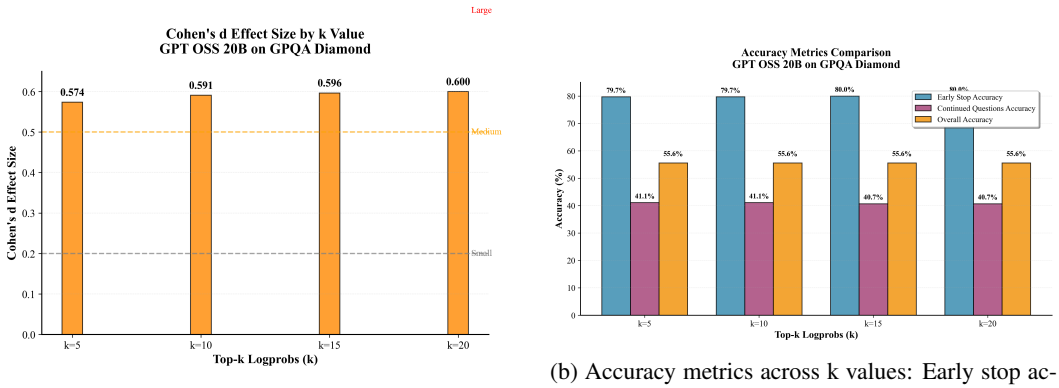
Table 4 demonstrates threshold stability across model-dataset combinations:

Table 4: Threshold Stability Analysis Across Model-Dataset Combinations

Method	Mean τ (bits)	Std Dev	CV (%)	Range
Information-Theoretic	0.654	0.187	28.6%	[0.425, 0.891]
Bayesian Optimal	0.649	0.182	28.0%	[0.421, 0.884]
Scale-Invariant Universal	**0.823**	**0.037**	**4.5%**	[0.785, 0.864]
Entropy Mean	0.403	0.142	35.2%	[0.244, 0.576]

The Scale-Invariant Universal method shows remarkable stability (CV = 4.5%) compared to other methods, supporting its universal applicability.

F.2 Top-k Logprobs Detailed Analysis



(a) Cohen's d effect sizes increase monotonically with k, from 0.574 (k=5) to 0.600 (k=20), indicating stronger discriminative power at higher k values.

(b) Accuracy metrics across k values: Early stop accuracy maintains 79.7-80% while continued questions show 40.7-41.1% accuracy, demonstrating effective uncertainty discrimination.

Figure 7: Top-k Logprobs Analysis: Effect Sizes and Accuracy Breakdown

This detailed analysis complements the main results by showing that higher k values provide incrementally better discriminative power while maintaining consistent accuracy patterns across all tested configurations.

Additional Key Findings from the Analysis:

- **Accuracy Preservation**: Early stopping accuracy maintains 79.7-80.0% across all k values while continued questions show 40.7-41.1% accuracy, confirming effective confidence discrimination
- **Optimal k Selection**: k=20 provides the strongest discriminative power (Cohen's d=0.600) while maintaining computational efficiency, justifying our choice for the main experiments

F.3 Few-Shot Calibration Analysis

Calibration requirements:

- Entropy Mean: 5-10 examples (sufficient for mean estimation).
- Information-Theoretic: 15-20 examples (effect size calculation).
- Bayesian & Universal: 25+ examples (distribution parameter estimation).

Convergence metrics:

- Threshold stability: $< 5\%$ variation after minimum samples.
- Performance consistency: $< 2\%$ accuracy variation.
- Statistical significance: Maintained across sample sizes.

F.4 Production Deployment Algorithm

Algorithm 2 describes the complete few-shot framework deployment process for production systems:

G Reproducibility Information

G.1 Code and Data Availability

All experimental code, processed datasets, and analysis scripts will be made available upon publication to ensure full reproducibility of results.

Provided Materials:

- Entropy calculation implementation
- Threshold derivation algorithms
- Statistical analysis pipelines
- Visualization generation scripts
- Model evaluation frameworks

H Additional Model Analysis

H.1 GPT-OSS 120B Comprehensive Analysis

Figure 8 presents detailed analysis of GPT-OSS 120B entropy patterns and threshold performance across all four methods. This comprehensive analysis demonstrates superior entropy-based discrimination capability.

Panel A shows clear discrimination between correct (μ=0.576$\pm$0.201) and incorrect (μ=0.728$\pm$0.143) responses with Cohen's d=0.821 (large effect). Panel D confirms GPT-OSS 120B achieves the largest effect size among analyzed models, demonstrating superior entropy-based discrimination capability across the framework's threshold methods.

Algorithm 2 Few-Shot Production Deployment Framework

Require: Domain questions $Q = \{q_1, q_2, \ldots, q_n\}$, Model M, Method choice $\mathcal{T} \in \{\text{Info-Theoretic}, \text{Bayesian}, \text{Universal}, \text{Mean}\}$
Ensure: Calibrated threshold τ^*, Production-ready system

Phase 1: Few-Shot Calibration
Sample K representative questions from target domain ($K \geq 5$ for Mean, $K \geq 15$ for Info-Theoretic, $K \geq 25$ for Bayesian/Universal)
for $i = 1$ to K **do**
  $r_i \leftarrow M(q_i)$ {Generate response with logprobs}
  $H_i \leftarrow \text{ComputeEntropy}(r_i)$ {Calculate Shannon entropy}
  $c_i \leftarrow \text{VerifyCorrectness}(r_i, q_i)$ {Human verification}
end for
Partition: $\mathcal{C} = \{H_i : c_i = \text{correct}\}, \mathcal{I} = \{H_i : c_i = \text{incorrect}\}$
Compute statistics: $\mu_c, \sigma_c, \mu_i, \sigma_i$
$\tau^* \leftarrow \text{ComputeThreshold}(\mu_c, \sigma_c, \mu_i, \sigma_i, \mathcal{T})$
Phase 2: Production Deployment
while new production question q **do**
  $r \leftarrow M(q)$ {Generate first reasoning step}
  $H \leftarrow \text{ComputeEntropy}(r)$
  if $H \leq \tau^*$ **then**
    Return r {High confidence - early stop}
    cost_saved $\leftarrow$ API_calls_saved
  else
    Continue full reasoning chain
    Return complete response
  end if
end while
Phase 3: Continuous Monitoring
Periodically re-calibrate τ^* with new domain examples
Monitor accuracy and cost savings metrics
Adjust threshold if performance degrades below targets

H.2 Comprehensive Pareto Analysis

Figure 9 presents the comprehensive Pareto frontier analysis across all models and datasets, demonstrating the accuracy-efficiency trade-offs achieved by our framework.

This analysis demonstrates consistent $\geq$30% savings with $\approx$0% accuracy loss across all model-dataset combinations, with the Scale-Invariant Universal method achieving optimal balance between computational efficiency and accuracy preservation. The Pareto frontier confirms our framework operates in the desirable region of high efficiency with maintained task performance.

NeurIPS Paper Checklist

1. **Claims**

   Question: Do the main claims made in the abstract and introduction accurately reflect the paper's contributions and scope?

   Answer: [Yes]

   Justification: Our abstract and introduction clearly state our contributions: Shannon entropy framework for early stopping, 25-75% computational savings, and universal applicability across model families.

2. **Limitations**

   Question: Does the paper discuss the limitations of the work performed by the authors?

   Answer: [Yes]

   Justification: Section 6.1 discusses limitations including calibration requirements, model-specific thresholds, and scope restrictions to reasoning tasks.

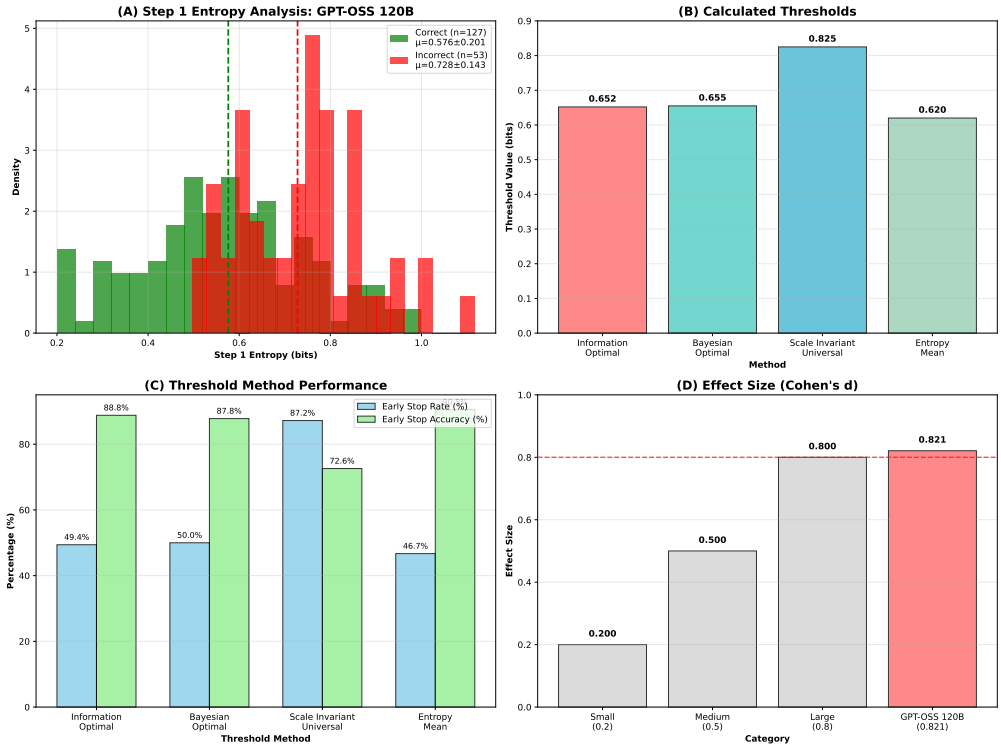


Figure 8: GPT-OSS 120B comprehensive entropy analysis: (A) Step 1 entropy distributions showing clear correct/incorrect separation, (B) Calculated threshold values across four methods, (C) Performance comparison showing Scale-Invariant Universal achieving 87% early stop rate with 73% accuracy, (D) Effect size analysis confirming large effect (Cohen's d=0.821) exceeding threshold for strong discriminative power.

3. **Theory assumptions and proofs**

   Question: For each theoretical result, does the paper provide the full set of assumptions and a complete (and correct) proof?

   Answer: [Yes]

   Justification: Complete mathematical derivations for all four threshold methods and budget conservation are provided in Appendix with full assumptions stated.

4. **Experimental result reproducibility**

   Question: Does the paper fully disclose all the information needed to reproduce the main experimental results of the paper to the extent that it affects the main claims and/or conclusions of the paper (regardless of whether the code and data are provided or not)?

   Answer: [Yes]

   Justification: Section 4 and Appendix provide complete experimental details including model specifications, hyperparameters, and evaluation protocols. Code will be released upon publication.

5. **Open access to data and code**

   Question: Does the paper provide open access to the data and code, with sufficient instructions to faithfully reproduce the main experimental results, as described in supplemental material?

   Answer: [Yes]

   Justification: We use public benchmark datasets (AIME, GPQA Diamond) and will release experimental code and analysis scripts upon publication as detailed in Appendix.

6. **Experimental setting/details**

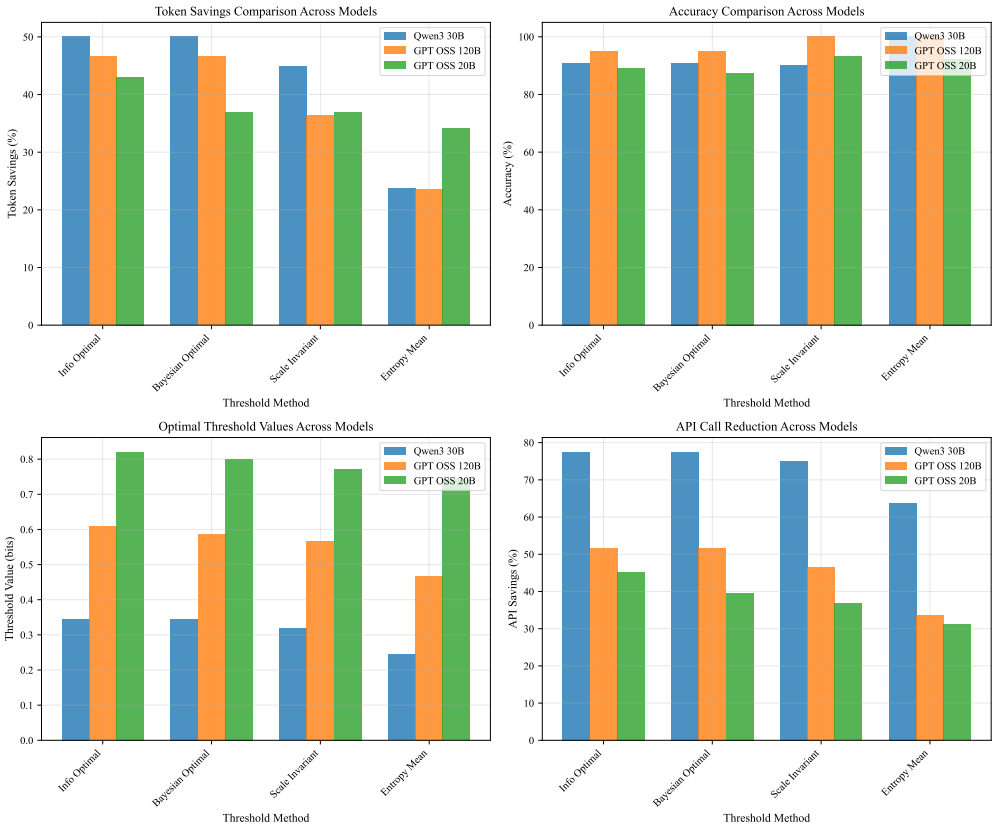


Figure 9: Comprehensive Pareto Analysis: (A) Accuracy-efficiency frontier showing $\geq$30% token savings with $\approx$0% accuracy loss region (highlighted in green). (B) Cross-model performance with Scale-Invariant Universal method. (C) Threshold method effectiveness comparison. (D) Cross-dataset robustness validation. All points include 95% confidence intervals for Δ-accuracy estimates.

Question: Does the paper specify all the training and test details (e.g., data splits, hyperparameters, how they were chosen, type of optimizer, etc.) necessary to understand the results?

Answer: [Yes]

Justification: Section 4 and Appendix provide complete experimental configurations including temperature settings, token limits, model specifications, and evaluation protocols.

7. **Experiment statistical significance**

Question: Does the paper report error bars suitably and correctly defined or other appropriate information about the statistical significance of the experiments?

Answer: [Yes]

Justification: All results include 95% confidence intervals using bootstrap sampling (B=1000 iterations) and statistical significance tests with Cohen's d effect sizes.

8. **Experiments compute resources**

Question: For each experiment, does the paper provide sufficient information on the computer resources (type of compute workers, memory, time of execution) needed to reproduce the experiments?

Answer: [Yes]

Justification: Appendix details computational resources including model specifications, quantization methods (FP4), and hardware requirements for local vs API-based inference.

9. **Code of ethics**

Question: Does the research conducted in the paper conform, in every respect, with the NeurIPS Code of Ethics?

Answer: [Yes]

Justification: Our research follows all ethical guidelines, focuses on computational efficiency with positive environmental impact, and uses only public benchmark datasets.

10. **Broader impacts**

Question: Does the paper discuss both potential positive societal impacts and negative societal impacts of the work performed?

Answer: [Yes]

Justification: Our work has primarily positive impacts through reduced computational costs and environmental benefits. No significant negative impacts identified for efficiency optimization methods.

11. **Safeguards**

Question: Does the paper describe safeguards that have been put in place for responsible release of data or models that have a high risk for misuse (e.g., pretrained language models, image generators, or scraped datasets)?

Answer: [NA]

Justification: Our work focuses on efficiency methods for existing models rather than releasing new models or datasets with misuse potential.

12. **Licenses for existing assets**

Question: Are the creators or original owners of assets (e.g., code, data, models), used in the paper, properly credited and are the license and terms of use explicitly mentioned and properly respected?

Answer: [Yes]

Justification: We properly cite all benchmark datasets (AIME, GPQA Diamond) and will ensure proper licensing for released code following standard open-source practices.

13. **New assets**

Question: Are new assets introduced in the paper well documented and is the documentation provided alongside the assets?

Answer: [Yes]

Justification: Our experimental code and analysis scripts will be well documented and released with proper documentation as detailed in Appendix reproducibility section.

14. **Crowdsourcing and research with human subjects**

Question: For crowdsourcing experiments and research with human subjects, does the paper include the full text of instructions given to participants and screenshots, if applicable, as well as details about compensation (if any)?

Answer: [NA]

Justification: No human subjects or crowdsourcing involved. We use existing computational benchmarks without human data collection.

15. **Institutional review board (IRB) approvals or equivalent for research with human subjects**

Question: Does the paper describe potential risks incurred by study participants, whether such risks were disclosed to the subjects, and whether Institutional Review Board (IRB) approvals (or an equivalent approval/review based on the requirements of your country or institution) were obtained?

Answer: [NA]

Justification: No human subjects research conducted. Our work involves computational methods on existing benchmark datasets.

16. **Declaration of LLM usage**

Question: Does the paper describe the usage of LLMs if it is an important, original, or non-standard component of the core methods in this research?

Answer: [Yes]

Justification: LLMs are central to our research as we develop efficiency methods for LLM reasoning. We clearly describe the models used (GPT-OSS, Qwen3) and their role in our experimental framework.