Adaptive Beam Search with Shannon Entropy for Data-centric Reasoning in LLMs

Yoonji Kim*[1], Yujin Jeong*[1], Jieun Kim[2], and Sung-Bae Cho[1](✉)

[1] Dept. of Computer Science, Yonsei University, Seoul, South Korea
[2] Dept. of Artificial Intelligence, Yonsei University, Seoul, South Korea
{yoonjikim,yujinj00,lilly9928,sbcho}@yonsei.ac.kr

Abstract. Reasoning capabilities of large language models (LLMs) have been significantly enhanced by structured prompting methods that leverage search over reasoning structures (e.g., Tree-of-Thoughts), making them increasingly valuable for data-centric reasoning and data-driven decision-making. However, existing methods with fixed exploration strategies often lead to suboptimal solutions or high computational costs due to exhaustive search. We propose an adaptive beam search method that uses entropy to dynamically adjust exploration at each reasoning step, balancing accuracy and efficiency. Uncertainty at each step is quantified through Shannon entropy of the confidence distribution, which serves as a dynamic threshold calibrating beam coverage. This enables broader exploration under high uncertainty and narrower exploration when uncertainty is low. Experimental results on data-centric tasks, arithmetic, commonsense, and symbolic reasoning tasks with Llama and GPT models demonstrate substantial improvements over state-of-the-art structured prompting methods with reduced computational cost, highlighting the efficacy of entropy-based beam adjustment in enhancing the reasoning capabilities of LLMs. The code for our method is publicly available at our GitHub repository.

Keywords: Multi-step Reasoning · Uncertainty Estimation · Adaptive Beam Search

1 Introduction

Large language models (LLMs) have demonstrated impressive capabilities in solving complex reasoning problems [20,1,19], making them increasingly valuable for data-driven reasoning and analytical decision making. However, they often exhibit cognitive rigidity, committing to a single reasoning path once chosen due to their lack of genuine inferential capabilities [12,2]. This limitation has motivated recent work on human-like deliberation, in which multiple reasoning paths are maintained, compared, and revised using intermediate evaluation or structural feedback [26,3]. Deliberative methods are typically categorized into chain-based approaches, which generate multiple independent reasoning chains

* These authors contributed equally.

and aggregate their outcomes [21], and structured-search approaches, which explore tree-like search spaces via iterative expansion and evaluation [24,27,4].

However, both approaches exhibit significant limitations. Chain-based methods rely on fixed sampling budgets and simple aggregation schemes that do not adapt to variations in problem difficulty, often resulting in suboptimal results on tasks that require deeper exploration [26]. Structured search with fixed exploration, such as BFS, leads to unnecessary exploration and poor resource utilization [11]. MCTS [7] focuses on important nodes but requires extensive simulations, making it computationally expensive [15]. These limitations necessitate more efficient exploration that dynamically adjusts to varying reasoning difficulties.

We propose an adaptive beam search based on entropy that dynamically adjusts exploration breadth at each step according to model uncertainty. Shannon entropy [16] of the confidence distribution over candidate thoughts quantifies step-wise uncertainty: high entropy triggers broader exploration, while low entropy focuses on confident paths. We map entropy values to beam allocations and retain paths proportional to candidates exceeding a predefined confidence threshold. This enables efficient resource allocation without fixed budgets or expensive rollouts, balancing accuracy and computational efficiency. The key contributions are as follows.

- **Addressing cognitive rigidity in LLM reasoning:** We mitigate LLM cognitive rigidity via uncertainty-guided adaptive exploration that can switch reasoning paths to avoid cascading errors and improve reliability in data science tasks.
- **Adaptive beam search based on entropy:** We propose an uncertainty-aware beam search that uses Shannon entropy over unified likelihood and self-evaluation scores to adapt beam breadth for data-centric reasoning.
- **Training-free and scalable solution:** Our training-free method improves accuracy on diverse reasoning benchmarks while maintaining or reducing computational cost, demonstrating practical scalability to complex tasks.

2 Related Works

2.1 Structured and Uncertainty-Aware Reasoning

Improving LLM reasoning has been approached primarily by structuring the generation process. Chain-of-Thought (CoT) [22] induces stepwise reasoning, while Self-Consistency [21] improves robustness by aggregating multiple sampled reasoning paths. More advanced works formulate reasoning as a search problem, exploring alternative intermediate thoughts through tree- or graph-structured expansions [26,3]. These methods employ classical exploration algorithms such as BFS, DFS, and beam search, as well as more sophisticated strategies including A*, MCTS, and deductive search [24,27]. Neuro-symbolic approaches have also been explored, integrating structured logical inference with pre-trained model knowledge to enable interpretable reasoning without task-specific training [8]

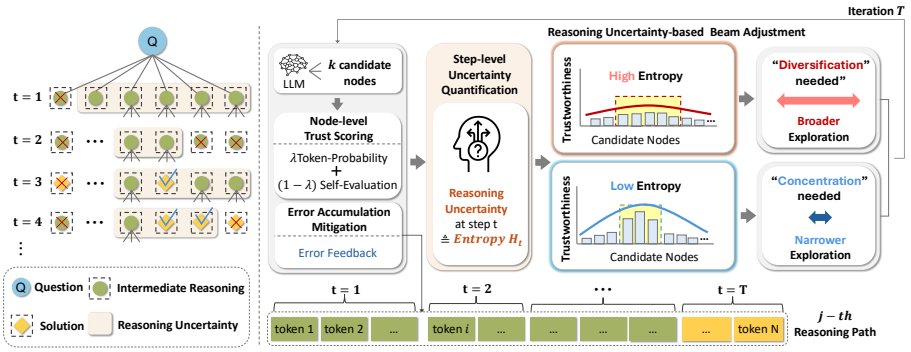


Fig. 1: Overview of our entropy-guided adaptive beam search.

However, these methods exhibit clear limitations. Learning-based methods incur additional training overhead, while search-based methods often suffer from inefficiency because they operate with fixed exploration settings that do not consider variations in problem difficulty or structure. Furthermore, fixed exploration strategies ignore uncertainty signals embedded in internal representations, potentially distorting intermediate reasoning paths [13].

To address this issue, we introduce an uncertainty-aware mechanism that estimates the reliability of intermediate reasoning steps. Specifically, we define a trustworthiness score by linearly interpolating the language model's generation probability with its self-evaluation probability, yielding a more robust uncertainty estimate without requiring additional supervision.

3 Proposed Method

Figure 1 illustrates the overall architecture, comprising node generation, trust scoring, reasoning uncertainty estimation, and adaptive beam search. By adapting the search based on node-level trust and step-wise uncertainty, our method supports robust and flexible multi-step reasoning.

3.1 Node Generation and Trust Scoring

Node Generation. Let θ denote model parameters and P_θ the distribution over generated token sequences. A *node* is a partial reasoning path at step t. We denote by S_t the set of candidate nodes and by $S'_t \subseteq S_t$ the subset selected for expansion. The sequence $S'_1, \ldots, S'_t$ forms a *reasoning tree*.

Given demonstrations D, input x, and a selected parent node $s'^m_{t-1} \in S'_{t-1}$, the model generates a candidate node s^n_t with

$$P_\theta(s^n_t) = \prod_{i=1}^{|s^n_t|} P_\theta\big(s^n_t[i] \mid s^n_t[1{:}i{-}1], x, D, s'^m_{t-1}\big)\,.$$

At each step, we sample $k = B \cdot K$ candidates $S_t = \{s_t^n\}_{n=1}^k$, where B is the beam size and K is the sampling multiplicity. The index m denotes the m-th selected node in S'_{t-1}.

Self-Evaluation. To assess the validity of each candidate node s_t^n, we prompt the model with a binary verification task that determines whether s_t^n is *correct* given input x and context s'^m_{t-1}. The prompt includes few-shot demonstrations D of correct and incorrect reasoning, and we use the token probability of `"yes"` as the self-evaluation score.

We denote this probability as $E(s_t^n) \in \mathbb{R}$:

$$\mathrm{E}(s_t^n) = P_\theta(\texttt{'yes'} \mid x, D, s'^m_{t-1}, s_t^n). \tag{1}$$

The evaluation probabilities at step t are

$$\mathcal{E}_t = \{\mathrm{E}(s_t^n) \mid s_t^n \in S_t\}. \tag{2}$$

If s_t^n is predicted as `incorrect`, we treat it as a reasoning error at step t and append corrective feedback to the next prompt, reducing error accumulation during autoregressive decoding.

Trustworthiness Score. In multi-step reasoning, the reliability of intermediate steps is crucial for guiding subsequent decisions. We therefore introduce a *trustworthiness score* that integrates *node generation* and *self-evaluation*. As shown in Figure 2, this score better separates correct and incorrect reasoning paths than existing confidence metrics.

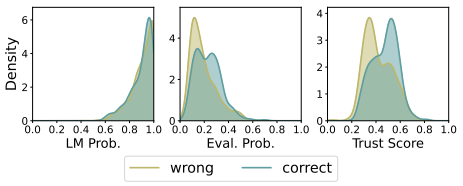


Fig. 2: Density distributions of confidence metrics, grouped by final answer correctness.

For each candidate node s_t^n at step t, we combine the model generation probability and the self-evaluation probability using a weighted log-linear form:

$$\mathrm{C}(s_t^n) = \lambda \cdot \log P_\theta(s_t^n) + (1 - \lambda) \cdot \log \mathcal{E}(s_t^n),$$

where $\lambda \in [0, 1]$ is a tunable hyperparameter that controls their relative contributions.

Let $\mathcal{C}_t = \{\mathrm{C}(s_t^n) \mid s_t^n \in S_t\}$ denote the trust scores at step t. To obtain a normalized distribution for estimating step-level uncertainty, we apply softmax:

$$\tilde{p}(s_t^n) = \frac{\exp(\mathrm{C}(s_t^n))}{\sum_i \exp(\mathrm{C}(s_t^i))}.$$

3.2 Adaptive Beam Search

Step-level reasoning uncertainty is quantified by the Shannon entropy of the normalized trustworthiness distribution $\tilde{p}(s_t^n)$ over candidate nodes S_t:

$$\mathcal{H}_t = -\sum_{s_t^n \in S_t} \tilde{p}(s_t^n) \log \tilde{p}(s_t^n).$$

We use the normalized entropy $\hat{\mathcal{H}}_t \in [0,1]$ to ensure comparability across steps with varying candidate-set sizes, defined as $\hat{\mathcal{H}}_t = \mathcal{H}_t / \log |S_t|$. A sharply peaked $\tilde{p}(s_t^n)$ yields low $\hat{\mathcal{H}}_t$, indicating concentrated trust, whereas a more uniform distribution yields high $\hat{\mathcal{H}}_t$, reflecting increased uncertainty.

Figure 1 illustrates this relationship and how $\hat{\mathcal{H}}_t$ determines the retention threshold. The full procedure is given in Algorithm 1.

Algorithm 1: Adaptive Beam Search

Input: $T, K, \theta, b_{\min}, b_{\max}$
$k \leftarrow K$ **for** $t \leftarrow 1$ **to** T **do**
  $S_t \leftarrow \{s_t^n\}_{n=1}^{k}$
  $\mathcal{C}_t \leftarrow \{\mathrm{C}(s_t^n) \mid s_t^n \in S_t\}$
  $\tilde{p}(s_t^n) \leftarrow \frac{\exp(\mathrm{C}(s_t^n))}{\sum_{s_t^i \in S_t} \exp(\mathrm{C}(s_t^i))}$
  $\mathcal{H}_t \leftarrow -\sum_{s_t^n \in S_t} \tilde{p}(s_t^n) \log \tilde{p}(s_t^n)$
  $\hat{\mathcal{H}}_t \leftarrow \mathcal{H}_t / \log |S_t|$
  $\{s_t^{(j)}\}_{j=1}^{|S_t|} \leftarrow \mathrm{Sort}_{\downarrow}(S_t; \tilde{p})$
  $S_t' \leftarrow \emptyset;\ U_t^0 \leftarrow 0;\ r \leftarrow 0$
  if $\hat{\mathcal{H}}_t > \theta$ **then**
    $\mathcal{M}_t^c \leftarrow \frac{1}{|S_t|} \sum_{s_t^n \in S_t} \mathrm{C}(s_t^n)$
    $b_t \leftarrow \mathcal{F}(\mathcal{M}_t^c; [0,1], [b_{\min}, b_{\max}])$
    $S_t' \leftarrow \{s_t^{(j)}\}_{j=1}^{\lfloor b_t \rfloor}$
  else
    while $U_t^r < \hat{\mathcal{H}}_t$ ***and*** $r < |S_t|$ **do**
      $r \leftarrow r + 1$
      $U_t^r \leftarrow U_t^{r-1} + \tilde{p}(s_t^{(r)})$
      $S_t' \leftarrow S_t' \cup \{s_t^{(r)}\}$
  $k \leftarrow K \cdot |S_t'|$

At each step t, candidates are ranked by trust score, and the beam size is adaptively calibrated using the entropy $\hat{\mathcal{H}}_t$. The cumulative probability up to rank k is

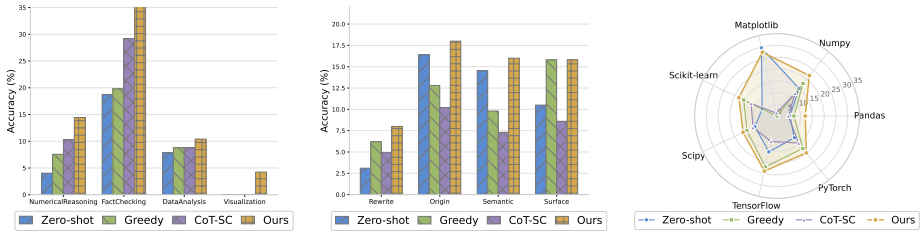
$$U_t^k = \sum_{j=1}^{k} \tilde{p}(s_t^{(j)}).$$

We select the smallest k satisfying $U_t^k \geq \hat{\mathcal{H}}_t$, forming the active set S_t' with beam size $B_t = |S_t'|$. This criterion retains more candidates under high uncertainty (larger $\hat{\mathcal{H}}_t$) and fewer when the model is confident (smaller $\hat{\mathcal{H}}_t$), balancing exploration and exploitation.

When $\hat{\mathcal{H}}_t > \theta$, indicating saturated uncertainty, we instead restrict expansion using a linear min–max mapping of the average trust score $\mathcal{M}_t^c$:

$$\mathcal{F}(\mathcal{M}_t^c; [a_{\min}, a_{\max}], [b_{\min}, b_{\max}]) = b_{\min} + \frac{\mathcal{M}_t^c - a_{\min}}{a_{\max} - a_{\min}} (b_{\max} - b_{\min}),$$

where $[a_{\min}, a_{\max}] = [0,1]$ and $[b_{\min}, b_{\max}]$ defines the target beam-size range. This scaling constrains search breadth under high uncertainty.



(a) Results on TableBench question types.

(b) Results on DS-1000 perturbation types.

(c) Results on DS-1000 libraries.

Fig. 3: Accuracy across TableBench question types and DS-1000 perturbation types and libraries.

4 Experiments

4.1 Experimental Setup

Benchmarks. We evaluate on eight benchmarks: GSM8K [5], AQuA [10], CommonsenseQA [18], StrategyQA [6], Date Understanding (BBH) [17], Game of 24 [26], DS-1000 [9], and TableBench [23]. **Baselines.** We compare against chain-based methods (Greedy [22], CoT-SC [21], ADoT [25], DCoT [14]) and tree-based methods (TD-CoT [24], DBS [27], ToT [26]). **Models and Hyperparameters.** Experiments are conducted on GPT-4o-mini, GPT-4, Llama-2 (7B, 13B) [20], and Llama-3.1 (8B, 70B). We use adaptive beam search with $b \in [2, 10]$, $n = 4$, temperature $\tau = 0.7$, and task-specific λ. **Evaluation Metrics.** We report answer accuracy, difficulty, beam size, token usage, and query cost (for closed-source models). Difficulty is defined as $1 - x$, where x is the fraction of sampled paths producing correct solutions. Efficiency is measured by average beam size, tokens per instance, and query cost.

4.2 Main Results

Responsible LLM-assisted Data Science. Figure 3 summarizes our results on TableBench and DS-1000. On TableBench, our method improves accuracy across all question types by an average of 7.3%. The largest gain appears in the Visualization category, where baselines perform poorly but our adaptive exploration recovers substantially more correct solutions. Across DS-1000 perturbation types, our method improves accuracy on the Difficult-Rewrite and Semantic splits from 6.2% to 8.0% and from 16.4% to 18.0%, respectively. The average accuracy across all perturbation types increases from 11.3% to 14.0%. At the library level, our method outperforms the strongest baseline in most cases. These results validate the effectiveness of uncertainty-guided adaptive search for multi-step data-centric reasoning and code generation.

Method	Venue	Arithmetic			Commonsense		
		GSM	AQuA	Avg	CSQA	SQA	Avg
Greedy	-	18.88	20.47	19.68	63.14	68.56	65.85
CoT-SC	NeurIPS'22	24.18	28.35	26.27	68.14	65.94	67.04
AdoT	EMNLP'24	39.50	31.10	35.30	69.50	65.80	67.65
DCoT	ACL'25	54.51	23.13	38.82	46.45	65.16	55.81
DBS	COLM'24	45.20	9.30	27.25	69.80	66.60	68.20
TD-CoT	NeurIPS'23	46.10	31.50	38.80	74.40	70.60	72.50
Ours (Beam)	-	56.67	38.19	52.74	**88.31**	68.56	78.43
Ours (Adaptive)	-	**69.69**	**47.24**	**58.47**	88.29	**71.18**	**79.73**

Table 1: Performance on arithmetic and commonsense benchmarks with Llama2-13B. "Ours (Beam)" uses trust-based beam selection only, whereas "Ours (Adaptive)" includes entropy-based adjustment.

Mitigating Intermediate Error Accumulation. Table 1 shows our method outperforms chain-based and tree-based baselines by effectively mitigating intermediate errors. Chain-based methods ADoT [25] and DCoT [14] achieve only 31.10% and 23.13% on AQuA, respectively, compared to our 47.24%. Tree-based methods TD-CoT [24] and DBS [27] incorporate structured exploration but suffer from heuristic breadth control, yielding lower arithmetic averages (38.80% and 27.25%, respectively). Our entropy-guided strategy dynamically adjusts reasoning breadth per step, promoting diverse exploration while selectively allocating resources to uncertain steps. This achieves the best overall performance: 58.47% on arithmetic and 79.73% on commonsense tasks.

Overcoming Fixed-Breadth Exploration. Tables 1 and 2 show that our method mitigates premature elimination of correct candidates. *Ours (B, Beam)* uses a fixed beam size of 2, while *Ours (A, Adaptive)* adjusts the beam size according to step-level uncertainty. Adaptive improves performance by 13.02 points on GSM8K, 9.05 points on AQuA, and 1.09 points on Date Understanding with GPT-4o mini, as well as 2.17 points with Llama-2-13B. Although gains are smaller on CommonsenseQA—where per-step uncertainty is lower—consistent improvements across all tasks indicate that our method overcomes the limitations of fixed-breadth exploration and enhances multi-step reasoning accuracy.

Model	Method	Accuracy
GPT-4o-mini	Greedy	76.15
	CoT-SC	81.57
	Ours (B)	84.82
	Ours (A)	**85.91**
Llama2-13B	Greedy	46.07
	CoT-SC	52.84
	Ours (B)	52.30
	Ours (A)	**54.47**

Table 2: Impact of dynamic beam breadth adjustment on Date Understanding task.

Cost Efficiency. Table 3 reports average token usage and cost per instance, where fewer output tokens indicate higher efficiency. With $b = 2$, ToT incurs low cost but achieves zero accuracy due to insufficient exploration, pruning correct candidates early. Larger b improves accuracy at much higher cost. In contrast, our method achieves a better accuracy–cost trade-off by adapting beam size based on step-level uncertainty, while maintaining an average beam size of 2.5.

Metric	ToT ($b = 2$)	ToT ($b = 3$)	Ours
Input (T)	7,374	7,796	**7,182**
Output (T)	1,392	1,353	**585**
Acc (%)	0	**40**	39
Cost ($)	0.3	0.32	**0.3**

Table 3: Performance comparison on 100 randomly selected Game of 24 problems using GPT-4.

4.3 Further Analysis

Ablation Study. We validate our trustworthiness estimation through an ablation on LLaMA-2-13B with three variants: (1) LM probability, (2) Evaluation probability, and (3) their interpolation. As shown in Table 4, the interpolated trust score better captures step-level uncertainty and improves performance. On AQuA ($\lambda = 0.2$), it achieves 47.24% accuracy, exceeding the LM and Eval variants by 3.93% and 9.05%, respectively. Similar gains are observed on StrategyQA ($\lambda = 0.9$) and Date Understanding ($\lambda = 0.2$), indicating the robustness of the combined signal during adaptive decoding.

Task	LM P.	Eval P.	Accuracy
SQA	✓		64.63
		✓	66.81
	✓	✓	**69.87**
AQuA	✓		43.31
		✓	38.19
	✓	✓	**47.24**
Date	✓		47.97
		✓	50.41
	✓	✓	**54.47**

Table 4: Ablation of LM and evaluation probabilities on StrategyQA, AQuA, and Date Understanding.

Scalability in Model Sizes. We adopt tree decoding with beam selection based on the highest LM probability as the base model. As shown in Table 5, our method consistently improves performance across all model sizes, demonstrating strong robustness and scalability. GPT-4o-mini achieves a substantial gain of over 12 points under our approach, while smaller models such as Llama3.1-8B and Llama2-7B also exhibit clear improvements. These findings indicate that the

Model	Base	Ours	Gap
Llama3.1-8B	57.09	58.27	+**1.18**
Llama2-7B	38.98	42.52	+**3.54**
Llama3.1-70B	66.93	73.62	+**6.69**
GPT-4o-mini	61.02	73.23	+**12.21**

Table 5: Scalability analysis: Accuracy on AQuA across model capacities.

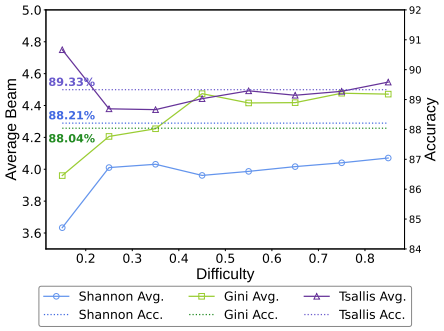


Fig. 4: Comparison of uncertainty metrics for beam search effectiveness.

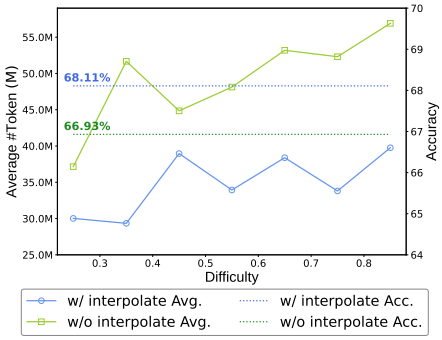


Fig. 5: Cost–accuracy trade-off under interpolation-based adaptive beam adjustment.

proposed reasoning framework remains effective across varying levels of model capacity.

Analysis of Adjustment Criteria Selection. To assess Shannon entropy as our uncertainty metric, we compare it with the Gini index

$$G = 1 - \sum_{s_t^n \in S_t} \tilde{p}(s_t^n)^2$$

and Tsallis entropy

$$S_q = \frac{1 - \sum_{s_t^n \in S_t} \tilde{p}(s_t^n)^q}{q - 1}.$$

on CommonsenseQA using Llama2-7B. We evaluate reasoning accuracy and average beam size for each metric (Figure 4). Shannon entropy achieves the most balanced accuracy–cost trade-off across difficulty levels. While its accuracy is 1.12 points lower than Tsallis entropy and 0.17 points higher than the Gini index, it incurs the lowest computational cost among the three.

Analysis of Interpolation. To assess the impact of adaptive search restriction, we conduct ablation experiments comparing our interpolation-based beam adjustment with and without the additional restriction. The interpolation mechanism dynamically constrains the beam size at each step based on the mean trust score, preventing unnecessary expansion in regions of high uncertainty. We evaluate both computational cost (average tokens) and task performance (accuracy) across difficulty levels on the AQuA dataset using GPT-4o-mini. As shown in Figure 5, this mechanism yields a more balanced accuracy-cost trade-off as difficulty increases, effectively reducing redundant exploration while maintaining or improving performance on more challenging problems. Figure 6 shows inference accuracy on CommonsenseQA as λ varies.

Hyperparameter Sensitivity Analysis. We investigate the effect of hyperparameter λ, which balances model generation confidence and self-evaluation probability. For GPT-4o-mini, accuracy ranges from 89.93% ($\lambda = 1.0$) to 91.56% ($\lambda = 0.6$), and LLaMA2-7B ranges from 85.67% ($\lambda = 0.8$) to 87.22% ($\lambda = 0.2$). Despite this variation, performance remains robust across different λ values, with the gap between best and worst settings being only 1.63%p for GPT-4o-mini and 1.55%p for LLaMA2-7B. Notably, even at the worst λ setting, our method consistently outperforms baseline approaches (see Table 1), demonstrating that the framework is robust to hyperparameter choice while still finding optimal balance points around $\lambda \in [0.2, 0.6]$.

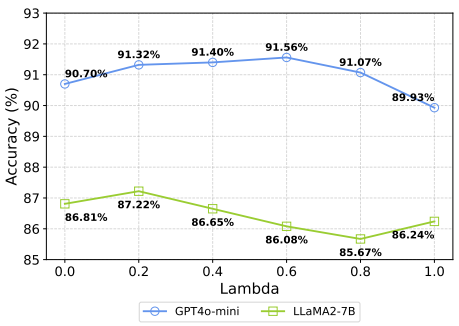


Fig. 6: Hyperparameter sensitivity analysis of λ.

Case Study. In this CommonsenseQA case study with GPT-4o, we analyze the search behavior of our method compared to TD-CoT [24]. Fixed-beam search methods such as TD-CoT generate four candidates per reasoning step but maintain a constant beam size of two, selecting beams from the sampled pool via probabilistic sampling. This rigid search often leads to premature pruning of high-quality reasoning chains: correct paths with low initial confidence may be discarded and never revisited. In contrast, our method adaptively expands the beam size in uncertain reasoning states (e.g., in the first step, where the beam size is increased to three), allocating additional exploration capacity when ambiguity is high. This flexibility enables the model to recover promising paths that would otherwise be pruned, and in this example our method maintains the correct trajectory that TD-CoT fails to preserve.

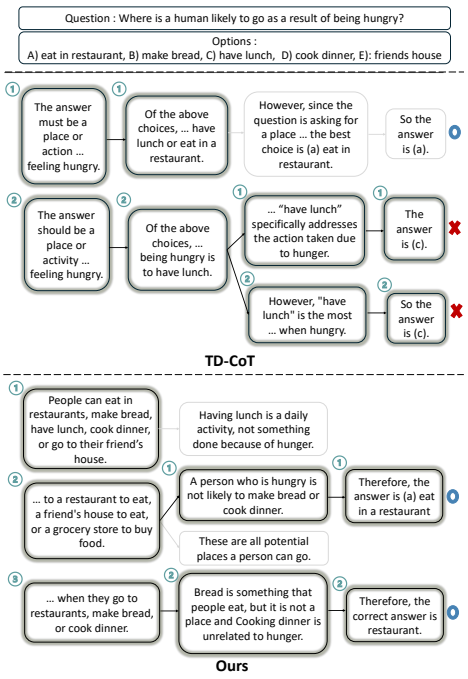


Fig. 7: Selected beams are shown in green; numbers indicate candidate ranks by selection probability.

5 Concluding Remarks

This paper proposes an entropy-based adaptive beam search method for structured prompting in LLMs. By leveraging Shannon entropy over trust distributions, the method adjusts the beam size at each reasoning step according to the model's uncertainty, enabling efficient allocation of computation that expands exploration when necessary and prunes suboptimal paths with confidence. Experiments across diverse benchmarks demonstrate consistent accuracy gains with reduced decoding costs and improved scalability over fixed-breadth and existing structured prompting baselines across model sizes and reasoning types. One limitation is that the entropy-based adjustment depends on the reliability of the model's probability estimates. Future work will explore alternative or learned uncertainty measures to enable more robust control.

Acknowledgements. This work was supported by IITP grant funded by the Korea government (MSIT) (No. RS-2020-II201361, Artificial Intelligence Graduate School Program (Yonsei University); No. RS-2022-II220113, Developing a Sustainable Collaborative Multi-modal Lifelong Learning Framework), and Air Force Defense Research Sciences Program funded by AFOSR.

References

1. Achiam, J., Adler, S., Agarwal, S., Ahmad, L., Akkaya, I., Aleman, F.L., Almeida, D., Altenschmidt, J., Altman, S., Anadkat, S., et al.: Gpt-4 technical report. arXiv preprint arXiv:2303.08774 (2023)
2. Ahn, J., Verma, R., Lou, R., Liu, D., Zhang, R., Yin, W.: Large language models for mathematical reasoning: Progresses and challenges. arXiv preprint arXiv:2402.00157 (2024)
3. Besta, M., Blach, N., Kubicek, A., Gerstenberger, R., Podstawski, M., Gianinazzi, L., Gajda, J., Lehmann, T., Niewiadomski, H., Nyczyk, P., et al.: Graph of thoughts: Solving elaborate problems with large language models. In: AAAI. vol. 38, pp. 17682–17690 (2024)
4. Chen, S., Li, B., Niu, D.: Boosting of thoughts: Trial-and-error problem solving with large language models. In: ICLR (2024)
5. Cobbe, K., Kosaraju, V., Bavarian, M., Chen, M., Jun, H., Kaiser, L., Plappert, M., Tworek, J., Hilton, J., Nakano, R., et al.: Training verifiers to solve math word problems. arXiv preprint arXiv:2110.14168 (2021)
6. Geva, M., Khashabi, D., Segal, E., Khot, T., Roth, D., Berant, J.: Did aristotle use a laptop? a question answering benchmark with implicit reasoning strategies. TACL **9**, 346–361 (2021)
7. Hao, S., Gu, Y., Ma, H., Hong, J., Wang, Z., Wang, D., Hu, Z.: Reasoning with language model is planning with world model. In: EMNLP. pp. 8154–8173 (2023)
8. Kim, J., Cho, S.B.: Neuro-symbolic reasoning with multiple large language models combined by first-order logic. In: International Conference on Hybrid Artificial Intelligence Systems. pp. 227–238. Springer (2025)
9. Lai, Y., Li, C., Wang, Y., Zhang, T., Zhong, R., Zettlemoyer, L., Yih, W.t., Fried, D., Wang, S., Yu, T.: Ds-1000: A natural and reliable benchmark for data science code generation. In: ICML. pp. 18319–18345. PMLR (2023)

10. Ling, W., Yogatama, D., Dyer, C., Blunsom, P.: Program induction by rationale generation: Learning to solve and explain algebraic word problems. In: ACL (2017)
11. Long, J.: Large language model guided tree-of-thought. arXiv preprint arXiv:2305.08291 (2023)
12. Mirzadeh, I., Alizadeh, K., Shahrokhi, H., Tuzel, O., Bengio, S., Farajtabar, M.: Gsm-symbolic: Understanding the limitations of mathematical reasoning in large language models. arXiv preprint arXiv:2410.05229 (2024)
13. Park, J., Cho, S.B.: Multi-feature quantized self-attention for fair large language models. In: The Fourteenth International Conference on Learning Representations (2026), https://openreview.net/forum?id=0UvgQxsi7S
14. Puerto, H., Chubakov, T., Zhu, X., Madabushi, H.T., Gurevych, I.: Fine-tuning on diverse reasoning chains drives within-inference cot refinement in llms. In: ACL. pp. 3789–3808 (2025)
15. Sel, B., Al-Tawaha, A., Khattar, V., Jia, R., Jin, M.: Algorithm of thoughts: Enhancing exploration of ideas in large language models. arXiv preprint arXiv:2308.10379 (2023)
16. Shannon, C.E.: A mathematical theory of communication. The Bell system technical journal **27**(3), 379–423 (1948)
17. Suzgun, M., Scales, N., Schärli, N., Gehrmann, S., Tay, Y., Chung, H.W., Chowdhery, A., Le, Q.V., Chi, E.H., Zhou, D., , Wei, J.: Challenging big-bench tasks and whether chain-of-thought can solve them. In: ACL. pp. 13003–13051 (2023)
18. Talmor, A., Herzig, J., Lourie, N., Berant, J.: Commonsenseqa: A question answering challenge targeting commonsense knowledge. In: NAACL (2019)
19. Team, G., Anil, R., Borgeaud, S., Alayrac, J.B., Yu, J., Soricut, R., Schalkwyk, J., Dai, A.M., Hauth, A., Millican, K., et al.: Gemini: a family of highly capable multimodal models. arXiv preprint arXiv:2312.11805 (2023)
20. Touvron, H., Lavril, T., Izacard, G., Martinet, X., Lachaux, M.A., Lacroix, T., Rozière, B., Goyal, N., Hambro, E., Azhar, F., et al.: Llama: Open and efficient foundation language models. arXiv preprint arXiv:2302.13971 (2023)
21. Wang, X., Wei, J., Schuurmans, D., Le, Q., Chi, E., Narang, S., Chowdhery, A., Zhou, D.: Self-consistency improves chain of thought reasoning in language models. In: ICLR (2023)
22. Wei, J., Wang, X., Schuurmans, D., Bosma, M., Xia, F., Chi, E., Le, Q.V., Zhou, D., et al.: Chain-of-thought prompting elicits reasoning in large language models. NIPS **35**, 24824–24837 (2022)
23. Wu, X., Yang, J., Chai, L., Zhang, G., Liu, J., Du, X., Liang, D., Shu, D., Cheng, X., Sun, T., et al.: Tablebench: A comprehensive and complex benchmark for table question answering. In: AAAI. vol. 39, pp. 25497–25506 (2025)
24. Xie, Y., Kawaguchi, K., Zhao, Y., Zhao, J.X., Kan, M.Y., He, J., Xie, M.: Self-evaluation guided beam search for reasoning. In: NIPS. vol. 36, pp. 41618–41650 (2023)
25. Xu, M., Li, Y., Sun, K., Qian, T.: Adaption-of-thought: Learning question difficulty improves large language models for reasoning. In: Al-Onaizan, Y., Bansal, M., Chen, Y.N. (eds.) EMNLP. pp. 5468–5495 (Nov 2024)
26. Yao, S., Yu, D., Zhao, J., Shafran, I., Griffiths, T., Cao, Y., Narasimhan, K.: Tree of thoughts: Deliberate problem solving with large language models. In: NIPS. vol. 36, pp. 11809–11822 (2023)
27. Zhu, T., Zhang, K., Xie, J., Su, Y.: Deductive beam search: Decoding deducible rationale for chain-of-thought reasoning (2024)