Cautious Next Token Prediction

Yizhou Wang[†*], Lingzhi Zhang[‡], Yue Bai[†], Mang Tik Chiu[‡], Zhengmian Hu[‡], Mingyuan Zhang[†], Qihua Dong[†], Yu Yin§, Sohrab Amirghodsi[‡] and Yun Fu[†]
[†]Northeastern University, [‡]Adobe, §Case Western Reserve University
wyzjack990122@gmail.com, yunfu@ece.neu.edu

Abstract

Next token prediction paradigm has been prevailing for autoregressive models in the era of LLMs. The current default sampling choice for popular LLMs is temperature scaling together with nucleus sampling (Holtzman et al., 2019) to balance diversity and coherence. Nevertheless, such approach leads to inferior performance in various NLP tasks when the model is not certain about testing questions. To this end, we propose a brand new training-free decoding strategy, dubbed as Cautious Next Token Prediction (CNTP). In the decoding process, if the model has comparatively high prediction entropy at a certain step, we sample multiple trials starting from the step independently and stop when encountering any punctuation. Then we select the trial with the lowest perplexity score viewed as the most probable and reliable trial path given the model's capacity. The trial number is negatively correlated with the prediction confidence, i.e., the less confident the model is, the more trials it should sample. This is consistent with human beings' behaviour: when feeling uncertain or unconfident, one tends to think more creatively, exploring multiple thinking paths, to cautiously select the path one feels most confident about. Extensive experiments on both LLMs and MLLMs show that our proposed CNTP approach outperforms existing standard decoding strategies consistently by a clear margin. Moreover, the integration of CNTP with self consistency (Wang et al., 2022) can further improve over vanilla self consistency. We believe our proposed CNTP has the potential to become one of the default choices for LLM decoding. Code is available at `https://github.com/wyzjack/CNTP`.

1 Introduction

Large Language Models (LLMs) have advanced the capabilities of natural language processing

[*]Work was done while Yizhou Wang was an intern at Adobe.

Table 1: Comparison of CNTP with Stochastic Decoding (**SD**), Greedy Decoding (**GD**), and Beam Search (**BS**) on key text generation properties. A ✓indicates exhibiting the property, while a ✗indicates not.

Textual Property	**SD**	**GD**	**BS**	**CNTP**
Stochasticity	✓	✗	✗	✓
Coherence	✗	✓	✓	✓
Creativity	✓	✗	✗	✓
Computational efficiency	✓	✓	✗	✓

rapidly, achieving remarkable performance in tasks spanning machine translation, summarization, and question answering (Brown et al., 2020; Radford et al., 2021; OpenAI, 2023; Touvron et al., 2023a,b; Dubey et al., 2024; Shen et al., 2025; Zhang et al., 2025). Beyond text-only domains, multimodal LLMs (MLLMs) extend these breakthroughs to image and video understanding, yielding transformative results in visual question answering and video event comprehension (Liu et al., 2023, 2024; Li et al., 2024). Despite this progress, test-time decoding strategies remain a bottleneck: standard approaches (e.g., greedy decoding, top-k sampling, nucleus sampling) can either produce dull or sub-optimal completions in uncertain contexts, undermining overall performance (Wang et al., 2023a; Aggarwal et al., 2023). Hence, devising novel inference-time algorithms to more effectively handle ambiguity and maintain coherent reasoning has become a critical challenge.

Recent research aims to bolster the reliability and depth of LLM outputs through chain-of-thought (CoT) reasoning, self-consistency voting, and iterative self-refinement (Wei et al., 2022; Gou et al., 2023; Wang et al., 2022; Aggarwal et al., 2023; Chen et al., 2023). These approaches have demonstrated impressive improvements on QA and reasoning benchmarks by explicitly sampling multiple solution paths. However, extensive multi-

Findings of the Association for Computational Linguistics: ACL 2025, pages 25685–25697
July 27 - August 1, 2025 ©2025 Association for Computational Linguistics

sample strategies often inflate computational cost, while purely single-sample methods risk hallucinations and local optimum traps. Furthermore, a great number of self-correction methods rely on external feedback signals (Gou et al., 2023; Chen et al., 2023), which can be intractable to deploy at scale. Therefore, a method that dynamically adapts its exploration only when the model is unconfident, and does so efficiently, remains highly desirable for real-world deployment and application. In this paper, we propose a **Cautious Next Token Prediction (`CNTP`)** approach that selectively samples multiple candidate paths whenever the model's prediction entropy is high, then automatically chooses the path with the lowest perplexity. Through conditioning the sampling depth on confidence, `CNTP` focuses computational resources precisely where the model is most uncertain, leading to stronger results in both purely linguistic and multimodal tasks. Extensive experiments show that `CNTP` consistently outperforms common decoding approaches. In summary, we make the following contributions:

- We introduce `CNTP`, a novel inference-only decoding strategy for LLMs that adaptively samples multiple continuations based on model confidence, thereby achieving high precision without the loss of diversity.
- We propose an entropy-based mechanism to control the number of sampled trials, enabling `CNTP` to conserve computational budget while reducing errors systematically in high-uncertainty regions.
- Through comprehensive studies on both language-only and multimodal benchmarks, we demonstrate that `CNTP` obtains superior performance in comparison to widely adopted decoding baselines consistently. In addition, we show how `CNTP` can be combined with self-consistency (Wang et al., 2022), further enhancing performance steadily.

2 Related works

LLM Reasoning The reasoning ability of Large Language models also plays a crucial role in multimodality tasks including image (Liu et al., 2023, 2024; Zhu et al., 2024) and video understanding (Maaz et al., 2024; Wang et al., 2023b; Song et al., 2024; Li et al., 2024; Chen et al., 2024). LLMs have demonstrated emergent reasoning abilities through chain-of-thought (CoT) prompting, enabling them to surpass earlier approaches on multi-step inference tasks (Wei et al., 2022; Kojima et al., 2022). Self-consistency (Wang et al., 2023a) extends CoT by sampling multiple reasoning paths and selecting the most frequent solution, yielding significant gains on math and commonsense QA tasks.Adaptive-Consistency (Aggarwal et al., 2023) halts sampling early if partial agreement is reached. The Tree-of-Thought framework explores multiple candidate "thought" sequences in a search-like manner, substantially improving performance on puzzles and longer-horizon tasks (Yao et al., 2024). In multimodal contexts, recent works incorporate vision inputs into multi-stage reasoning, achieving state-of-the-art results on image- and video-based QA (Xu et al., 2024; Fei et al., 2024). Compared with these methods, our approach offers a simpler yet more robust mechanism for stepwise reasoning without relying on extensive prompt engineering or heavy sampling, yielding more effective inference.

LLM Test-time Sampling At inference time, widely used stochastic strategies such as top-p sampling or top-p (nucleus) sampling (Holtzman et al., 2019) balance diversity and fluency, becoming the default choices for modern LLMs particularly in industrial products (OpenAI, 2023; Guo et al., 2025; Team et al., 2025; Yang et al., 2024). More recently, min-p sampling (Nguyen et al., 2024) puts forward a dynamic truncation method which adjusts the sampling threshold contigent on the model's confidence by scaling based on the top token probability. As to deterministic sampling strategies, Greedy Decoding selects the single most probable token at each step by choosing the one with the maximum token probability. Beam Search (Graves, 2012) keeps track of a fixed number of top candidate sequences at each step and finally selects the trial with the highest joint probability. Unlike these strategies, our method adaptively samples multiple paths at high-entropy steps, using perplexity to select the best continuation, thereby enhancing answer quality without losing stochasticity.

LLM Self-correction LLMs sometimes produce errors or hallucinations, prompting the development of self-refinement or self-correction. Gou et al. (2023) shows that LLMs can self-correct by taking advantage of external tools to validate their initial outputs, gathering feedback on specific aspects, and then refining the content based on that feedback iteratively. Kumar et al. (2024) proposes a finetuning strategy to make LLM self-correct by en-

Algorithm 1 Cautious Next Token Prediction

Initialization: Language model M, initial sequence s, max trials $N_{\max}$, entropy thresholds $H_{\min}, H_{\max}$
while s does not satisfy stopping criteria **do**
  Compute token distribution $p(\cdot \mid s)$ via M
  $H \leftarrow -\sum_{w} p(w \mid s)\, \log p(w \mid s)$
  $N \leftarrow \max\Big(1, \min\Big(N_{\max}, \lfloor \frac{H - H_{\min}}{H_{\max} - H_{\min}} \times N_{\max} \rfloor\Big)\Big)$
  if $N = 1$ **then**
    Sample a single token s_{single}
    $s \leftarrow s + s_{\text{single}}$
  else
    for $i \leftarrow 1$ to N **do**
      Sample path s_i via M until punctuation or satisfying stopping criteria
      $\mathcal{L}(s_i) \leftarrow -\sum_t \log p\big(w_t \mid s_{<t}; M\big)$
      $\text{PPL}(s_i) \leftarrow \exp\big(\mathcal{L}(s_i)/|s_i|\big)$
    end for
    $s \leftarrow s + \arg\min_{s_i} \text{PPL}(s_i)$
  end if
end while
return completed sequence s

gaging in multi-turn online reinforcement learning on its own self-generated correction traces. However, Huang et al. (2023) comes to the conclusion that current LLMs still cannot do self-reflection via directly prompting to itself to reflect on the generated answers without the access to external tools. Our proposed approach remains fully self-contained without extra critic models or extensive domain-specific tuning. In fact, our approach can be viewed as local and progressive self-correction by means of pursuing low sentence perplexity in attempted short explorations.

3 Method

3.1 Motivation

When human beings solve complex tasks—such as proving math theorems—they typically proceed step-by-step, carefully reviewing each line for correctness. If a certain step appears ambiguous or risky, they explore multiple potential pathways, reflect on each, and ultimately choose the route that seems most convincing or "probable." This metacognitive process of careful thinking motivates our proposed Cautious Next Token Prediction (`CNTP`) algorithm. Specifically, in human exam settings, **individuals often re-check key steps and sample alternative reasoning paths when unsure, only finalizing the path that best resonates with previously established facts. Such progressive practice usually results in better human performance**. By analogy, we hypothesize that LLMs might also benefit from conditionally branching into multiple future continuations whenever they sense high uncertainty. Our core insight is to compute an *entropy* measure that indicates how "unsure" the model is, and trigger more thorough exploration exactly at those points. Once possible continuations are sampled, the model's own likelihood function can judge the best candidate to proceed with, mirroring how humans choose the strongest proof line. In this work, **we are thrilled to confirm the empirical effects of such practice on LLMs as well**.

3.2 Cautious Next Token Prediction

Preliminaries and Notation. Let M be a language model that defines a probability distribution $p_\theta(w \mid s)$ over the next token w given the current (partial) sequence s. At each generation step, M computes the distribution over its vocabulary V as

$$p_\theta(w \mid s) = \frac{\exp\big(\text{logit}_\theta(s, w)\big)}{\sum_{v \in V} \exp\big(\text{logit}_\theta(s, v)\big)}, \quad (1)$$

where $\text{logit}_\theta(s, w)$ is the unnormalized score for token w. We measure uncertainty via the *entropy* of this distribution:

$$H(s) = -\sum_{w \in V} p_\theta(w \mid s)\, \log p_\theta(w \mid s). \quad (2)$$

When $H(s)$ is large, the model is more uncertain and less confident about the next token.

Adaptive Trial Sampling. As shown in Algorithm 1, our `CNTP` method adaptively decides how many candidate continuations to explore based on the entropy $H(s)$. We specify two thresholds, $H_{\min}$ and $H_{\max}$, and a maximum trial budget $N_{\max}$. We then map the current entropy $H(s)$ to a suitable number of trials N:

$$N = \max\Big(1, \min\Big(N_{\max}, \lfloor \frac{(H(s) - H_{\min}) \cdot N_{\max}}{(H_{\max} - H_{\min})} \rfloor\Big)\Big). \quad (3)$$

This ensures $N = 1$ when $H(s)$ is below $H_{\min}$ (i.e., the model is quite confident), and $N = N_{\max}$ when $H(s)$ exceeds $H_{\max}$. For each trial $i \in \{1, \ldots, N\}$, we *sample a candidate path* s_i until a punctuation token or stopping criterion. We

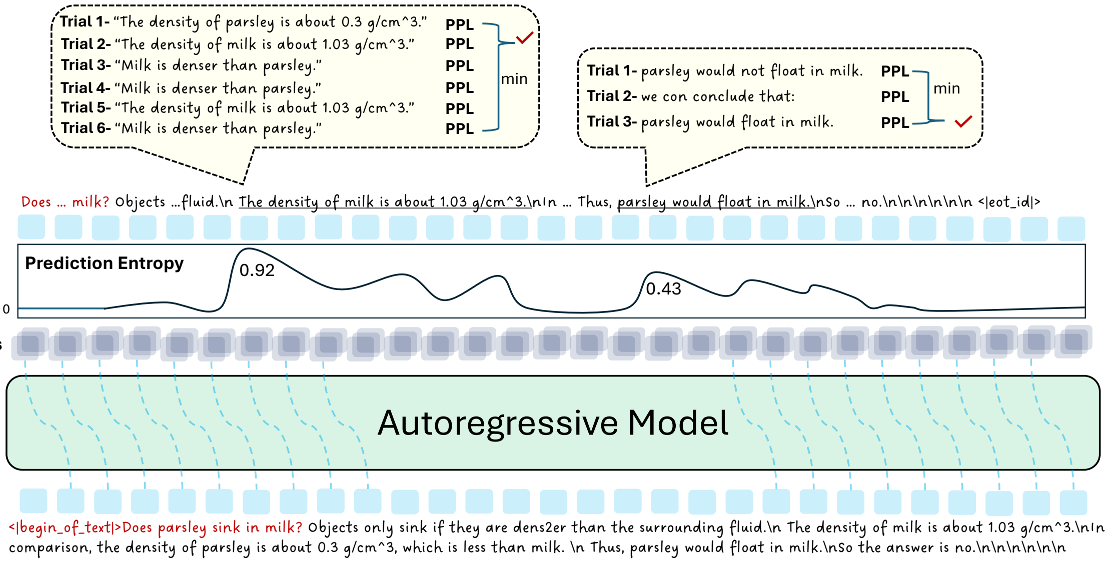

Figure 1: Overview of **C**autious **N**ext **T**oken **P**rediction (**CNTP**) method. Given a question in the testing phase, the AR model predict next tokens in an iterative way. Different from traditional one-token-by-one-token generation style, we leverage the prediction entropy of the token probability vectors and sample multiple trials to look ahead if the entropy is high enough. The different trials are sampled independently and stop when encountering a punctuation, then the trial with the lowest perplexity is selected to be deemed the most confident choice. The higher the prediction entropy is, the more trial paths we sample. The QA example is generated by our approach on one of the StrategyQA (Geva et al., 2021) testing samples. The question texts are in red and the answer texts are in black.

compute the path's negative log-likelihood (NLL),

$$\mathcal{L}(s_i) = -\sum_{t=1}^{|s_i|} \log p_\theta\big(w_t \mid s_{<t}\big), \tag{4}$$

and convert it to perplexity (PPL),

$$\mathrm{PPL}(s_i) = \exp\big(\tfrac{\mathcal{L}(s_i)}{|s_i|}\big). \tag{5}$$

The path with the *lowest* perplexity is considered the most likely to be correct. CNTP thus updates the sequence s by appending the best candidate continuation. The process repeats until a global termination condition (e.g., an end-of-sentence token or length limit) is reached.

Algorithmic Overview. Algorithm 1 and Fig. 1 summarizes the entire CNTP loop. Notably, the method is *cautious* in that it only expends extra sampling steps when the model signals uncertainty. At confident steps, CNTP automatically defaults to a single-sample approach. **This design aligns with our human-inspired motivation: we "think harder" only when the situation is ambiguous.**

3.3 Complexity Analysis

Let L be the final sequence length. In the worst-case scenario, CNTP samples $N_{\max}$ continuations for each token, producing $O(L \times N_{\max})$ total decoding operations. This upper bound resembles standard multi-sample decoding. However, because N is only large when $H(s)$ is high, CNTP often runs with $N = 1$ for confident tokens, thereby incurring substantially fewer computations in practice. If p denotes the fraction of steps selected from multi-trial sampling of the total steps, then the expected complexity is $O(L \times (1 + p(N_{\max} - 1)))$, which can be much smaller than the naive $O(L \times N_{\max})$ if $p \ll 1$.

Comparison to Decoding Baselines. Greedy or single-sample decoding has $O(L)$ complexity but may be prone to errors at uncertain steps. Beam search similarly takes $O(L \times B)$ time for beam width B, but does not adapt its effort. Self-consistency performs N_{sc} fully independent decoding passes, costing $O(N_{sc} \times L)$ with no mid-sequence adaptivity. Our proposed CNTP can be viewed as adaptively switching between these extremes based on the model's entropy. This dynamic policy yields more efficient exploration of the search space precisely when needed.

3.4 CNTP is Prone to Lead to the Correct Answer Provably

We provide a theoretical result to highlight the effectiveness of CNTP in generating a *correct entire sequence*. We assume that each sequence token is critical: any incorrect token at some step $N_{\max}$

leads to a globally incorrect final sequence.

Definition 1 (Full-Sequence Correctness)**.** Let L be the maximum decoding length. We say a final generated sequence $S = (w_1, \ldots, w_L)$ is *correct* if all its tokens match the ground-truth reference $(c_1, \ldots, c_L)$. Let $P_{\text{CNTP}}(\text{correct})$ denote the probability that CNTP produces the exact correct sequence, and similarly $P_{\text{Single}}(\text{correct})$ for single-sample (greedy) decoding.

Notation. Let $p_\theta(w \mid s)$ be the language model's distribution over the next token w given partial sequence s. Define the *entropy* at each step $N_{\max}$ as $H_t = -\sum_w p_\theta(w \mid s_{<t}) \log p_\theta(w \mid s_{<t})$. For the correct token c_t, let $p_\theta(c_t \mid s_{<t})$ be its probability. - If CNTP decides to sample N_t continuations at step $N_{\max}$, it obtains candidate expansions $\{s^i_{\leq t},\ i = 1, \ldots, N_t\}$ (where each $s^i_{\leq t}$ extends $s_{<t}$ by one or more tokens until punctuation or a stopping criterion). Let $\text{PPL}(s^i_{\leq t})$ be the perplexity of the extended token subsequence. We introduce two mild assumptions before proceeding:

Assumption 1. Whenever the ground-truth token (or short path) c is among the sampled candidates $\{s^i\}$, it attains strictly the lowest perplexity among all incorrect candidates. Formally, if $c \in \{s^1, \ldots, s^N\}$, then

$$\text{PPL}(c) < \text{PPL}(u) \tag{6}$$

$$\text{for every incorrect } u \in \{s^1, \ldots, s^N\}. \tag{7}$$

Assumption 2. If the model has high entropy $H_t \geq H_{\min}$ at step $N_{\max}$, the probability that the correct token c_t is *not* discovered by a single sample is sufficiently large (i.e., $1 - p_\theta(c_t \mid s_{<t})$ is non-negligible). Equivalently, the model is "aware" of its own uncertainty:

$$H_t \quad \text{large} \quad \Longrightarrow \quad p_\theta(c_t \mid s_{<t}) \quad \text{small} \tag{8}$$

Under the assumptions, we present a theorem comparing CNTP to single-sample decoding in terms of both correctness probability over the entire output sequence and the computational cost.

Theorem 1 (**CNTP Outperforms Single-Sample Decoding in Full-Sequence Correctness**)**.** Let S be the final sequence of length L generated by either CNTP or single-sample decoding. Assume Assumption 1 and 2 hold, we have:

1. **Full-sequence correctness:**

$$P_{\text{CNTP}}(\text{correct}) \geq P_{\text{Single}}(\text{correct}), \tag{9}$$

with strict inequality if there exists at least one step t where CNTP uses $N_t > 1$ trials *and* the single-sample approach likely yields an incorrect token at step t.

2. **Expected cost:** Let $\mathcal{C}(s)$ denote the cost (number of forward passes) to decode sequence s. Denote by p the fraction of steps selected from multi-trial sampling of the total steps. Then, letting $\mathbb{E}[\cdot]$ be the expectation over random draws:

$$\mathbb{E}[\mathcal{C}_{\text{CNTP}}(S)] \leq L \times \Big[1 + p\,\big(N_{\max} - 1\big)\Big], \\ < L \times N_{\max}.$$

so CNTP 's average cost is strictly lower than $N_{\max}$-sample decoding at each step.

The proof is in Appendix B. Theorem 1 shows that CNTP 's adaptive multi-sample policy strictly improves the probability of generating a correct full sequence compared to single-sample decoding. Meanwhile, it does not impose the uniform high cost of sampling $N_{\max}$ times at every step. Instead, CNTP 's increased sampling effort is only triggered when $H_t \geq H_{\min}$ (i.e., the model is less certain). Consequently, CNTP *bridges* the gap between single-sample decoding (fast but more prone to errors at uncertain tokens) and uniform multi-sample decoding (robust but expensive). Under reasonable assumptions, the perplexity-based ranking ensures that once the correct continuation is drawn, CNTP selects it with high probability. Empirically, we observe this behavior aligns well with LLMs' typical calibration of probability and perplexity (OpenAI, 2023; Kojima et al., 2022).

4 Experiment

4.1 Setting

We compared CNTP with the most commonly used decoding strategies: stochastic decoding (Holtzman et al., 2019), greedy decoding, beam search and self consistency (SC) (Wang et al., 2022). We implement our algorithm on SOTA LLMs: Llama 2 (Touvron et al., 2023b), Llama 3.1 (Dubey et al., 2024) and DeepSeek-R1 (Guo et al., 2025)-distilled Qwen (Yang et al., 2024) models. We evaluate on both fixed-answer reasoning benchmarks GSM-8K (Cobbe et al., 2021), MATH (Hendrycks et al., 2021), StrategyQA (Geva et al., 2021) and open-ended benchmark Truthful-QA (Lin et al., 2021) (where Self Consistency cannot be applied). For

Table 2: Comparison (accuracy % and the number of generated tokens) of Llama-3.1-8B-Instruct on GSM8K (Cobbe et al., 2021) and MATH (Hendrycks et al., 2021) (Math Reasoning), and StrategyQA (Geva et al., 2021) (Commonsense Reasoning). The best result is in **bold**.

Approach	GSM8K		MATH		StrategyQA	
	Accuracy	# Tokens	Accuracy	# Tokens	Accuracy	# Tokens
Single Reasoning Chain						
Deterministic						
Greedy Decoding	79.8	$0.15k$	41.5	$0.29k$	72.9	$0.06k$
Stochastic						
Stochastic Decoding*	$79.4_{\pm 0.8}$	$0.15k$	$41.5_{\pm 1.2}$	$0.29k$	$72.0_{\pm 0.7}$	$0.06k$
Ours*	$\mathbf{81.6}_{\pm 0.6}$	$0.46k$	$\mathbf{47.1}_{\pm 1.7}$	$0.87k$	$\mathbf{73.2}_{\pm 0.2}$	$0.35k$
Multiple Reasoning Chains						
Deterministic						
Beam Search (beam=5)	82.3	$0.74k$	48.0	$1.46k$	72.9	$0.31k$
Stochastic						
SC (40 paths)	84.8	$6.03k$	56.0	$11.7k$	76.2	$2.45k$
Ours + SC (40 paths)	**85.2**	$18.4k$	**57.5**	$34.5k$	**76.3**	$13.9k$

Table 3: Comparison (accuracy % and the number of generated tokens) of DeepSeek-R1-Distill-Qwen-1.5B on GSM8K (Cobbe et al., 2021) and MATH (Hendrycks et al., 2021) (Math Reasoning), and StrategyQA (Geva et al., 2021) (Commonsense Reasoning). The best result is in **bold**.

Approach	GSM8K		MATH		StrategyQA	
	Accuracy	# Tokens	Accuracy	# Tokens	Accuracy	# Tokens
Single Reasoning Chain						
Deterministic						
Greedy Decoding	64.6	$0.17k$	32.5	$0.94k$	**53.6**	$0.06k$
Stochastic						
Stochastic Decoding*	$61.6_{\pm 1.1}$	$0.17k$	$27.9_{\pm 3.7}$	$0.85k$	$51.7_{\pm 1.2}$	$0.06k$
Ours*	$\mathbf{65.7}\ _{\pm 0.7}$	$0.49k$	$\mathbf{37.7}_{\pm 1.7}$	$2.51k$	$53.0_{\pm 1.3}$	$0.34k$
Multiple Reasoning Chains						
SC (40 paths)	**78.3**	$6.81k$	29.5	$33.9k$	47.7	$2.41k$
Ours + SC (40 paths)	71.7	$19.8k$	**41.0**	$98.9k$	**54.1**	$13.2k$

all the reasoning datasets, we employ Chain of Thought (Wei et al., 2022) prompting following the literature. On MATH dataset, we follow Tulu 3 (Lambert et al., 2024) and randomly select 200 testing samples. For multi-model LLM, we test on Llama-3.2-Vision-11B (Dubey et al., 2024) and LLaVA-CoT (Xu et al., 2024) on MMVet (Yu et al., 2023) and MathVista (Lu et al., 2023) benchmarks. For all the methods we employ standard temperature scaling and nucleus sampling. For the top p parameter, for Llama-based models we set the value as 0.9 for all the models and for DeepSeek-R1-distilled Qwen we set the value as 0.95, which are in accordance with their training default values. For `CNTP`, we set hyperparameter $N_{\text{max}} = 10$, $H_{\text{min}} = 0.01$ and $H_{\text{max}} = 1.5$ for all the experiments. For the punctuation set, we use the set `{.,?!:;)]}\n}.`. More detailed setting is in Appendix C.

4.2 Result

As shown in Tab. 2 and 3, our approach outperforms baselines in both single and multiple reasoning chain settings across GSM8K, MATH, and StrategyQA datasets. Although introducing more token computation, our method improves significantly over greedy decoding and stochastic decoding, which are the most commonly used decoding strategies in modern LLMs. Also, the computation burden of `CNTP` is overall lower than Beam Search with 5 beams and is much lower than SC. Notably, the integration of `CNTP` with SC exceed the

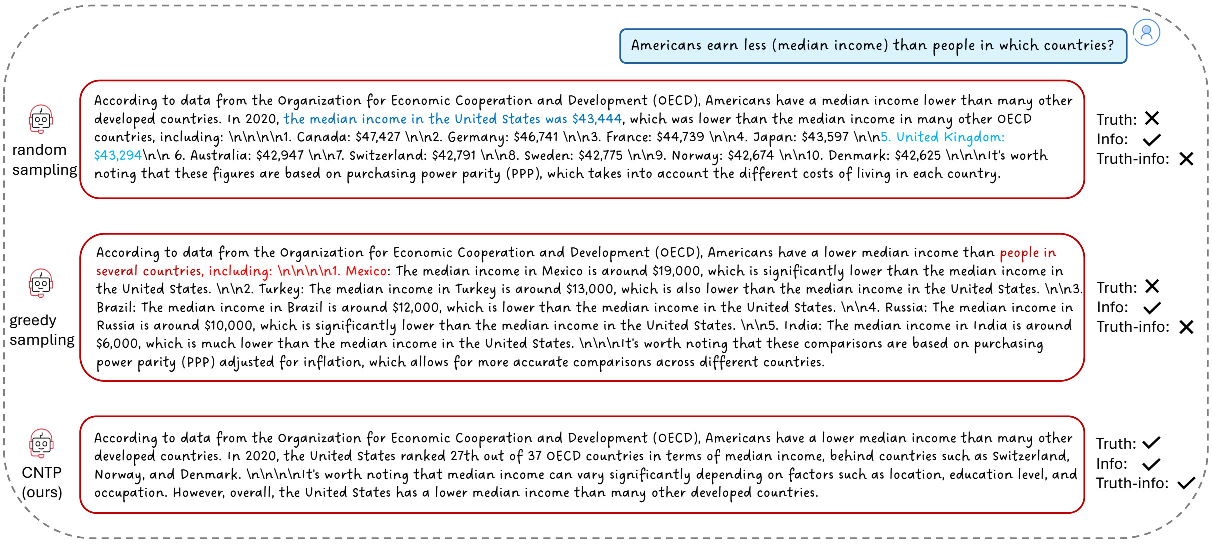

Figure 2: One QA example of Truthful-QA. When using random sampling with temperature scaling and nucleus sampling, the outputs show hallucination (marked in blue) and incoherence (marked in light blue), whereas greedy sampling also produces misinformed text (marked in red). In contrast, our CNTP avoids all of these issues.

Table 4: Temperature values for the results in Tab. 2.

Approach	GSM8K	MATH	StrategyQA
Single Reasoning Chain			
Greedy Decoding	0	0	0
Stochastic Decoding	0.6	0.6	0.6
Ours	1.2	0.6	0.8
Multiple Reasoning Chains			
Beam Search (beam=5)	0	0	0
SC (40 paths)	0.6	0.6	0.6
Ours + SC (40 paths)	1.2	0.6	0.8

Table 5: Temperature values for the results in Tab. 3.

Approach	GSM8K	MATH	StrategyQA
Single Reasoning Chain			
Greedy Decoding	0	0	0
Stochastic Decoding	0.6	0.6	0.6
Ours	1.2	0.6	0.8
Multiple Reasoning Chains			
Beam Search (beam=5)	0	0	0
SC (40 paths)	0.6	0.6	0.6
Ours + SC (40 paths)	1.2	0.6	0.8

Table 6: Temperature values for the results in Tab. 8.

Approach	Truthful-QA
Deterministic (Greedy Decoding)	0
Stochastic Decoding	0.6
Ours	0.8

vanilla SC in most cases, especially when SC fails on StrategyQA with DeepSeek-R1-Distill-Qwen-1.5B. Additionally, as shown in Table 8, our method significantly outperforms greedy and stochastic decoding on Truthful-QA using Llama-2-7B-Chat. Our method achieves the highest truthfulness accuracy (84.8%), surpassing stochastic decoding by +6.8% and greedy decoding by +5.7%, indicating its effectiveness in mitigating hallucinations for open-ended question answering. Our approach also excels in multi-modal reasoning benchmarks as in Tab. 9, which demonstrate the generality of CNTP. For completeness, we provide the detailed temperature values for every single experimental result presented in Tables 4, 5, 6, and 7.

4.3 Ablation study

Ablation of the confidence estimation strategy We conduct experiments using max token probability and max token probability minus second token probability (Farr et al., 2024) value as the confidence measure of Llama-3.1-8B-Instruct model on GSM8K, StrategyQA, MATH and TruthfulQA (Truth info acc. is reported). 5 independent run results in Fig. 3 show that entropy as confidence measurement leads to the best performance. This might results from that token probability distribution entropy considers the whole vocabulary distribution, making the confidence and uncertainty

Table 7: Temperature values for the results in Tab. 9.

Approach	MMVet	MathVista
Greedy Decoding	0	0
Stochastic Decoding	0.6	0.6
Ours	0.8	0.8

Table 8: Comparison (accuracy %) on Truthful-QA using Llama-2-7B-Chat. The best result is in **bold**.

Approach	Info Acc.	Truth Acc.	Truth-info Acc.
Stochastic Decoding	88.0$_{\pm 0.6}$	78.0$_{\pm 0.5}$	66.0$_{\pm 0.3}$
Greedy Decoding	78.5	79.1	57.6
Ours	**89.2**$_{\pm 1.2}$	**84.8**$_{\pm 0.5}$	**74.0**$_{\pm 1.1}$

Table 9: Accuracy of Llama-3.2-11B-Vision-Instruct (top) and LLaVA-CoT (bottom) on MLLM benchmarks.

Approach	MMVet	MathVista
Greedy Decoding	48.0	47.8
	53.5	53.4
Stochastic Decoding	47.7	48.8
	53.0	55.3
Ours	**53.5** (↑ 5.5)	**49.2** (↑ 0.4)
	58.5 (↑ 5.0)	**55.7** (↑ 0.4)

estimation more accurate and informative.

Ablation of the scaling trial strategy. In CNTP, if the model is more confident, we sample fewer trials inspired from human being behaviours. We supplement experimental results when sampling the fixed amount of trials (we set 6 as the middle between 1 and 10). We also provide results when the trial number is negatively correlated with entropy, i.e., $N = \max\left(1, \min\left(N_{\max}, N_{\max} - \lfloor \frac{H - H_{\min}}{H_{\max} - H_{\min}} \times N_{\max} \rfloor\right)\right)$. We conclude from Tab. 10 that positive correlation results in the best performance, suggesting that just like humans, LLMs ought to also explore more when feeling uncertain. Notice that on the Truthful QA dataset, only positive correlation way (ours) can exhibit a high and reasonable result since the other two variants tend to generate repetitive text chunks in the answer.

Ablation of perplexity computation range. CNTP adopt a sentence-level perplexity computation strategy. We further try Best-of-N sampling using the whole perplexity of the generated answers (including the complete CoT path). The best path is selected based on the lowest perplexity of the whole generated sequences. As observed from Tab. 11, the Best-of-N sampling cannot exhibit superiority over CNTP (81.6 on average) or even greedy decoding (79.8). This implies the necessity of sentence-level perplexity computation in a progressive way.

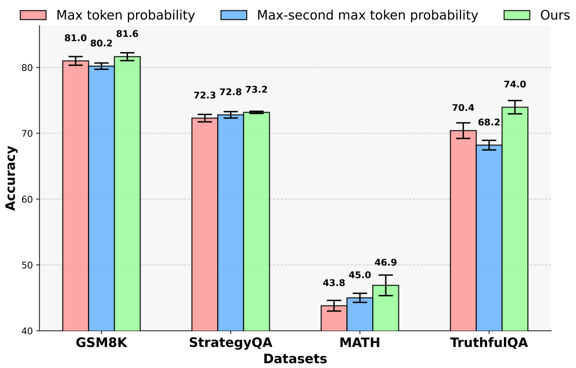

Figure 3: Comparison between CNTP and other two confidence measuring strategies on Llama-3.1-8B-Instruct.

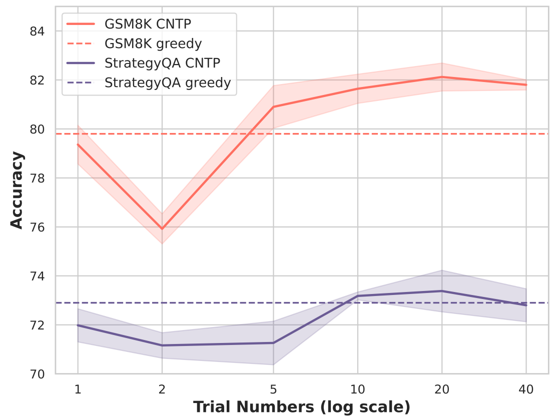

Figure 4: Performance curves of CNTP when scaling up the max trial numbers on GSM8K and StrategyQA datasets using Llama-3.1-8B-Instruct.

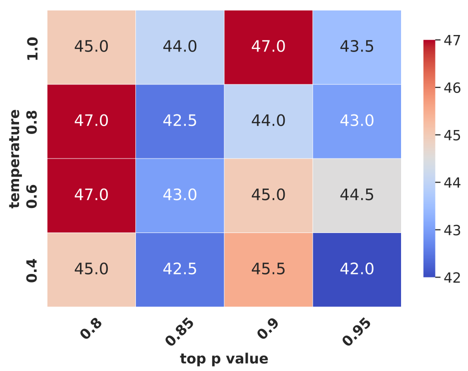

Figure 5: Accuracy of CNTP using Llama-3.1-8B-Instruct on MATH w.r.t. temperature and top p values.

4.4 Sensitivity Analysis

Max trial number. We explore the performance of CNTP on GSM8K and StrategyQA of Llama-3.1-

Table 10: Comparison of Llama-3.1-8B-Instruct using different trial number scaling strategies.

Dataset	Same # of trials	Negatively Correlated.	**Ours**
GSM8K	81.1	81.2	**81.6**
StrategyQA	72.7	72.7	**73.2**
TruthfulQA	3.80	3.80	**74.0**

Table 11: Best-of-N (using whole answer perplexity) performance on GSM8K using Llama-3.1-8B-Instruct.

Dataset	N=2	N=5	N=10	N=20	N=40
GSM8K	79.2	79.5	78.2	77.3	76.1

8B-Instruct under varying max trial number N_{max}. Fig. 4 shows that the performance first decrease and then increase, and finally enter accuracy saturation or even slightly decrease. This means that setting the max trial number too small or too large is not ideal. This is different from self consistency which always improve when scaling up the trial number. The reason lies in that CNTP seek local optimality in the reasoning process so there is a exploration-and-exploitation trade off and a moderate N_{max} will work excellently, roughly in the range $[10, 30]$.

Temperature and top p parameter. We test the performance of CNTP on Llama-3.1-8B on MATH dataset under varying temperature and top p values. The performance of both stochastic decoding and greedy decoding is around 41.5 (%). Fig. 5 demonstrates that CNTP is robust under varying temperature and top p thresholds with all the performance surpassing the baselines.

4.5 Qualitative Example and Analysis.

To manifest the superiority of CNTP concretely, we present one testing QA dialogue result of TruthfulQA dataset predictions. As seen from Fig. 2, when asked about people in which countries Americans earn less median income than, CNTP can avoid being misleaded to the wrong reasoning path and can provide the most truthful information owing to the ability to choose the right path in the second sentence. This shows the importance of sticking to the correct reasoning chain in the early stage of CoT process, which can realized via perplexity-based local optimality seeking.

5 Conclusion

We introduce Cautious Next Token Prediction (CNTP), a novel training-free decoding approach for LLMs that focuses additional computation on high-uncertainty steps. By sampling multiple candidate paths only when entropy is high and selecting the path with the lowest perplexity, CNTP achieves consistent improvements on both unimodal (text-only) and multimodal tasks. Our experiments demonstrate that CNTP outperforms standard sampling techniques and requires less overhead than multi-sample methods like self-consistency. In the future, we plan to extend CNTP on Autoregressive Models beyond text generation, such as image generation (Sun et al., 2024; Tian et al., 2024).

Limitations

One limitation of this work is that the proposed next token prediction algorithm introduces more token computations during inference compared to vanilla next token prediction. Nevertheless, this can be largely alleviated by 1) using sampling without replacement for the multi-trial sampling process in CNTP with smaller max trial number $N_{\max}$, 2) taking advantage of some advanced inference speeding up techniques, such as speculative decoding (Leviathan et al., 2023), which involves a smaller, faster model suggesting multiple tokens at once, which are then checked by a larger model in parallel. We can also deploy our CNTP using vLLM (Kwon et al., 2023) framework, which speeds up LLM decoding mainly through PagedAttention, optimizing memory use. Also, the newly brought computation burden is far less than other multi-trial approaches such as beam search (Graves, 2012) and self consistency (Wang et al., 2022). Therefore, we believe the considerable improvement in performance on benchmarks outweighs the introduced computation complexities.

References

Pranjal Aggarwal, Aman Madaan, Yiming Yang, and Mausam. 2023. Let's sample step by step: Adaptive-consistency for efficient reasoning and coding with LLMs. In *EMNLP*.

Tom Brown, Benjamin Mann, Nick Ryder, Melanie Subbiah, Jared D Kaplan, Prafulla Dhariwal, Arvind Neelakantan, Pranav Shyam, Girish Sastry, Amanda Askell, et al. 2020. Language models are few-shot learners. *Advances in neural information processing systems*, 33:1877–1901.

Xinyun Chen, Renat Aksitov, Uri Alon, Jie Ren, Kefan Xiao, Pengcheng Yin, Sushant Prakash, Charles Sutton, Xuezhi Wang, and Denny Zhou. 2023. Universal self-consistency for large language model generation. *arXiv preprint arXiv:2311.17311*.

Zhawnen Chen, Tianchun Wang, Yizhou Wang, Michal Kosinski, Xiang Zhang, Yun Fu, and Sheng Li. 2024. Through the theory of mind's eye: Reading minds with multimodal video large language models. *arXiv preprint arXiv:2406.13763*.

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

Abhimanyu Dubey, Abhinav Jauhri, Abhinav Pandey, Abhishek Kadian, Ahmad Al-Dahle, Aiesha Letman, Akhil Mathur, Alan Schelten, Amy Yang, Angela Fan, et al. 2024. The llama 3 herd of models. *arXiv preprint arXiv:2407.21783*.

David Farr, Iain Cruickshank, Nico Manzonelli, Nicholas Clark, Kate Starbird, and Jevin West. 2024. Llm confidence evaluation measures in zero-shot css classification. *arXiv preprint arXiv:2410.13047*.

Hao Fei, Shengqiong Wu, Wei Ji, Hanwang Zhang, Meishan Zhang, Mong-Li Lee, and Wynne Hsu. 2024. Video-of-thought: Step-by-step video reasoning from perception to cognition. In *Forty-first International Conference on Machine Learning*.

Mor Geva, Daniel Khashabi, Elad Segal, Tushar Khot, Dan Roth, and Jonathan Berant. 2021. Did aristotle use a laptop? a question answering benchmark with implicit reasoning strategies. *Transactions of the Association for Computational Linguistics*, 9:346–361.

Zhibin Gou, Zhihong Shao, Yeyun Gong, Yelong Shen, Yujiu Yang, Nan Duan, and Weizhu Chen. 2023. Critic: Large language models can self-correct with tool-interactive critiquing. *arXiv preprint arXiv:2305.11738*.

Alex Graves. 2012. Sequence transduction with recurrent neural networks. *arXiv preprint arXiv:1211.3711*.

Daya Guo, Dejian Yang, Haowei Zhang, Junxiao Song, Ruoyu Zhang, Runxin Xu, Qihao Zhu, Shirong Ma, Peiyi Wang, Xiao Bi, et al. 2025. Deepseek-r1: Incentivizing reasoning capability in llms via reinforcement learning. *arXiv preprint arXiv:2501.12948*.

Dan Hendrycks, Collin Burns, Saurav Kadavath, Akul Arora, Steven Basart, Eric Tang, Dawn Song, and Jacob Steinhardt. 2021. Measuring mathematical problem solving with the math dataset. *arXiv preprint arXiv:2103.03874*.

Ari Holtzman, Jan Buys, Li Du, Maxwell Forbes, and Yejin Choi. 2019. The curious case of neural text degeneration. *arXiv preprint arXiv:1904.09751*.

Jie Huang, Xinyun Chen, Swaroop Mishra, Huaixiu Steven Zheng, Adams Wei Yu, Xinying Song, and Denny Zhou. 2023. Large language models cannot self-correct reasoning yet. *arXiv preprint arXiv:2310.01798*.

Takeshi Kojima, Shixiang Shane Gu, Machel Reid, Yutaka Matsuo, and Yusuke Iwasawa. 2022. Large language models are zero-shot reasoners. *NeurIPS*, 35:22199–22213.

Aviral Kumar, Vincent Zhuang, Rishabh Agarwal, Yi Su, John D Co-Reyes, Avi Singh, Kate Baumli, Shariq Iqbal, Colton Bishop, Rebecca Roelofs, et al. 2024. Training language models to self-correct via reinforcement learning. *arXiv preprint arXiv:2409.12917*.

Woosuk Kwon, Zhuohan Li, Siyuan Zhuang, Ying Sheng, Lianmin Zheng, Cody Hao Yu, Joseph Gonzalez, Hao Zhang, and Ion Stoica. 2023. Efficient memory management for large language model serving with pagedattention. In *Proceedings of the 29th Symposium on Operating Systems Principles*, pages 611–626.

Nathan Lambert, Jacob Morrison, Valentina Pyatkin, Shengyi Huang, Hamish Ivison, Faeze Brahman, Lester James V Miranda, Alisa Liu, Nouha Dziri, Shane Lyu, et al. 2024. T\" ulu 3: Pushing frontiers in open language model post-training. *arXiv preprint arXiv:2411.15124*.

Yaniv Leviathan, Matan Kalman, and Yossi Matias. 2023. Fast inference from transformers via speculative decoding. In *International Conference on Machine Learning*, pages 19274–19286. PMLR.

Kunchang Li, Yali Wang, Yinan He, Yizhuo Li, Yi Wang, Yi Liu, Zun Wang, Jilan Xu, Guo Chen, Ping Luo, Limin Wang, and Yu Qiao. 2024. Mvbench: A comprehensive multi-modal video understanding benchmark. In *CVPR*.

Stephanie Lin, Jacob Hilton, and Owain Evans. 2021. Truthfulqa: Measuring how models mimic human falsehoods. *arXiv preprint arXiv:2109.07958*.

Haotian Liu, Chunyuan Li, Yuheng Li, and Yong Jae Lee. 2024. Improved baselines with visual instruction tuning. In *CVPR*.

Haotian Liu, Chunyuan Li, Qingyang Wu, and Yong Jae Lee. 2023. Visual instruction tuning. In *NeurIPS*.

Pan Lu, Hritik Bansal, Tony Xia, Jiacheng Liu, Chunyuan Li, Hannaneh Hajishirzi, Hao Cheng, Kai-Wei Chang, Michel Galley, and Jianfeng Gao. 2023. Mathvista: Evaluating mathematical reasoning of foundation models in visual contexts. *arXiv preprint arXiv:2310.02255*.

Muhammad Maaz, Hanoona Rasheed, Salman Khan, and Fahad Khan. 2024. Video-ChatGPT: Towards detailed video understanding via large vision and language models. In *ACL*.

Minh Nguyen, Andrew Baker, Clement Neo, Allen Roush, Andreas Kirsch, and Ravid Shwartz-Ziv. 2024. Turning up the heat: Min-p sampling for creative and coherent llm outputs. *arXiv preprint arXiv:2407.01082*.

OpenAI. 2023. Gpt-4 technical report. *CoRR, abs/2303.08774*.

Alec Radford, Jong Wook Kim, Chris Hallacy, Aditya Ramesh, Gabriel Goh, Sandhini Agarwal, Girish Sastry, Amanda Askell, Pamela Mishkin, Jack Clark, et al. 2021. Learning transferable visual models from natural language supervision. pages 8748–8763. PMLR.

Xuan Shen, Yizhou Wang, Xiangxi Shi, Yanzhi Wang, Pu Zhao, and Jiuxiang Gu. 2025. Efficient reasoning with hidden thinking. *arXiv preprint arXiv:2501.19201*.

Enxin Song, Wenhao Chai, Guanhong Wang, Yucheng Zhang, Haoyang Zhou, Feiyang Wu, Haozhe Chi, Xun Guo, Tian Ye, Yanting Zhang, Yan Lu, Jenq-Neng Hwang, and Gaoang Wang. 2024. Moviechat: From dense token to sparse memory for long video understanding. In *CVPR*.

Peize Sun, Yi Jiang, Shoufa Chen, Shilong Zhang, Bingyue Peng, Ping Luo, and Zehuan Yuan. 2024. Autoregressive model beats diffusion: Llama for scalable image generation. *arXiv preprint arXiv:2406.06525*.

Kimi Team, Angang Du, Bofei Gao, Bowei Xing, Changjiu Jiang, Cheng Chen, Cheng Li, Chenjun Xiao, Chenzhuang Du, Chonghua Liao, et al. 2025. Kimi k1. 5: Scaling reinforcement learning with llms. *arXiv preprint arXiv:2501.12599*.

Keyu Tian, Yi Jiang, Zehuan Yuan, Bingyue Peng, and Liwei Wang. 2024. Visual autoregressive modeling: Scalable image generation via next-scale prediction. *arXiv preprint arXiv:2404.02905*.

Hugo Touvron, Thibaut Lavril, Gautier Izacard, Xavier Martinet, Marie-Anne Lachaux, Timothée Lacroix, Baptiste Rozière, Naman Goyal, Eric Hambro, Faisal Azhar, et al. 2023a. Llama: Open and efficient foundation language models. *arXiv preprint arXiv:2302.13971*.

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

Xuezhi Wang, Jason Wei, Dale Schuurmans, Quoc Le, Ed Chi, Sharan Narang, Aakanksha Chowdhery, and Denny Zhou. 2022. Self-consistency improves chain of thought reasoning in language models. *arXiv preprint arXiv:2203.11171*.

Xuezhi Wang, Jason Wei, Dale Schuurmans, Quoc V Le, Ed H. Chi, Sharan Narang, Aakanksha Chowdhery, and Denny Zhou. 2023a. Self-consistency improves chain of thought reasoning in language models. In *ICLR*.

Yizhou Wang, Ruiyi Zhang, Haoliang Wang, Uttaran Bhattacharya, Yun Fu, and Gang Wu. 2023b. Vaquita: Enhancing alignment in llm-assisted video understanding. *arXiv preprint arXiv:2312.02310*.

Jason Wei, Xuezhi Wang, Dale Schuurmans, Maarten Bosma, Fei Xia, Ed Chi, Quoc V Le, Denny Zhou, et al. 2022. Chain-of-thought prompting elicits reasoning in large language models. *Advances in Neural Information Processing Systems*, 35:24824–24837.

Guowei Xu, Peng Jin, Li Hao, Yibing Song, Lichao Sun, and Li Yuan. 2024. Llava-o1: Let vision language models reason step-by-step. *arXiv preprint arXiv:2411.10440*.

An Yang, Baosong Yang, Beichen Zhang, Binyuan Hui, Bo Zheng, Bowen Yu, Chengyuan Li, Dayiheng Liu, Fei Huang, Haoran Wei, et al. 2024. Qwen2. 5 technical report. *arXiv preprint arXiv:2412.15115*.

Shunyu Yao, Dian Yu, Jeffrey Zhao, Izhak Shafran, Tom Griffiths, Yuan Cao, and Karthik Narasimhan. 2024. Tree of thoughts: Deliberate problem solving with large language models. *Advances in Neural Information Processing Systems*, 36.

Weihao Yu, Zhengyuan Yang, Linjie Li, Jianfeng Wang, Kevin Lin, Zicheng Liu, Xinchao Wang, and Lijuan Wang. 2023. Mm-vet: Evaluating large multimodal models for integrated capabilities. *arXiv preprint arXiv:2308.02490*.

Mingyuan Zhang, Yue Bai, Huan Wang, Yizhou Wang, Qihua Dong, and Yun Fu. 2025. Boosting large language models with mask fine-tuning. *arXiv preprint arXiv:2503.22764*.

Deyao Zhu, Jun Chen, Xiaoqian Shen, Xiang Li, and Mohamed Elhoseiny. 2024. MiniGPT-4: Enhancing vision-language understanding with advanced large language models. In *ICLR*.

Cautious Next Token Prediction

Supplementary Material

A Relationship of CNTP with Beam Search (Graves, 2012) and Self-Consistency (Wang et al., 2023a)

Traditional beam search maintains a fixed number of candidate expansions at every time step (Graves, 2012). While effective in deterministic scenarios (e.g., speech recognition), beam search does not inherently adapt to model uncertainty. It can also over-penalize diverse continuations if beams converge to similar partial hypotheses.

By contrast, self-consistency (Wang et al., 2023a) generates multiple *reasoning paths* and selects the most frequent final answer. Though powerful for reasoning tasks, it requires multiple full-path samples (often 5–40) regardless of how confident the model might be at intermediate steps. In essence, *both* beam search and self-consistency can be seen as uniform multi-sample strategies.

CNTP differs by linking sampling depth to real-time confidence signals via entropy, making it more cost-efficient. In cases of low uncertainty, CNTP defaults to a simpler single-sample path akin to greedy decoding. As soon as the model's internal distribution "spreads out," CNTP triggers more extensive branching and perplexity-based selection, thus combining the benefits of beam search's path exploration with self-consistency's final voting—but only where needed.

B Proof of Theorem 1

Proof. *(1) Full-sequence correctness.* Denote by A_t the event that *the token chosen at step t is correct* and no previous errors occurred (so the partial sequence remains correct). For single-sample decoding,

$$P_{\text{Single}}(A_t) = P_{\text{Single}}\big(w_t = c_t \mid A_1, \ldots, A_{t-1}\big). \tag{10}$$

If $H_t < H_{\min}$, then CNTP also uses $N_t = 1$ trial, so $P_{\text{CNTP}}(A_t) = P_{\text{Single}}(A_t)$. If $H_t \geq H_{\min}$, then by 2, $p_\theta(c_t \mid s_{<t})$ is small, so a single sample might miss c_t. However, CNTP uses $N_t > 1$ trials. The probability that *none* of these trials produce c_t is $(1 - p_\theta(c_t \mid s_{<t}))^{N_t} \ll (1 - p_\theta(c_t \mid s_{<t}))$. Once c_t is among the candidates, Assumption 1 ensures it will be selected due to the lowest perplexity. Therefore, for high-entropy steps,

$$P_{\text{CNTP}}(A_t) = 1 - \big(1 - p_\theta(c_t \mid s_{<t})\big)^{N_t}, \tag{11}$$
$$> p_\theta(c_t \mid s_{<t}) = P_{\text{Single}}(A_t). \tag{12}$$

Hence, $P_{\text{CNTP}}(A_t) \geq P_{\text{Single}}(A_t)$ at every step, and strictly greater if $N_t > 1$. The probability of the *entire sequence* being correct is the product $\prod_{t=1}^{L} P(A_t)$. Thus $P_{\text{CNTP}}(\text{correct}) \geq P_{\text{Single}}(\text{correct})$ with strict inequality if any step used $N_t > 1$.

(2) Expected cost. Each decoding step entails N_t forward passes if CNTP chooses to sample N_t times. Let $I_t = 1$ if $H_t \geq H_{\min}$ (high-entropy) and 0 otherwise. Then

$$\mathcal{C}_{\text{CNTP}}(S) = \sum_{t=1}^{L} N_t, \tag{13}$$

where $N_t = 1 + (N_{\max} - 1)\,\mathbf{1}\big(I_t = 1\big)$ if $H_t \leq H_{\max}$ or saturates at $N_{\max}$ if $H_t > H_{\max}$ (we clamp $N_t \leq N_{\max}$). Let $p = \frac{1}{L}\sum_{t=1}^{L} \mathbb{E}[I_t]$ be the expected fraction of high-entropy steps. Then

$$\mathbb{E}\big[\mathcal{C}_{\text{CNTP}}(S)\big] = \sum_{t=1}^{L} \mathbb{E}[N_t] \tag{14}$$
$$\leq L\Big[1 + p\,(N_{\max} - 1)\Big]. \tag{15}$$

This is strictly less than $LN_{\max}$, which would be the cost of always sampling $N_{\max}$ continuations at every step (uniform multi-sample decoding). □

C More Experiment Setting & Details

We tune the temperature value for the baseline methods and for CNTP in list $[0.6, 0.8, 1.0, 1.2]$ and select the best performance. We tune the beam size in list $[2, 3, 5, 10, 20, 40]$ and select the best performance for all experiments. For all the single-reasoning chain stochastic approaches in the LLM experiments, we run 5 times independently and report the average performance and sample standard deviations. For the TruthfulQA experiments, we use Llama-2-7B as LLM judge to output the truth accuracy, info accuracy and truth info accuracy. For the GSM8K experiments, we adopt 9-shot CoT during inference. For StrategyQA dataset,we adopt 6-shot CoT during inference. For MATH dataset, we adopt 4-shot CoT during inference. All experiments are done on NVIDIA A100-SXM4-80GB GPUs and Intel(R) Xeon(R) Platinum 8275CL CPU @ 3.00GHz CPUs with 96 logical processors.