Hyperparameter-Free-Sampling: Entropy Equilibrium for Text Generation

Anonymous ACL submission

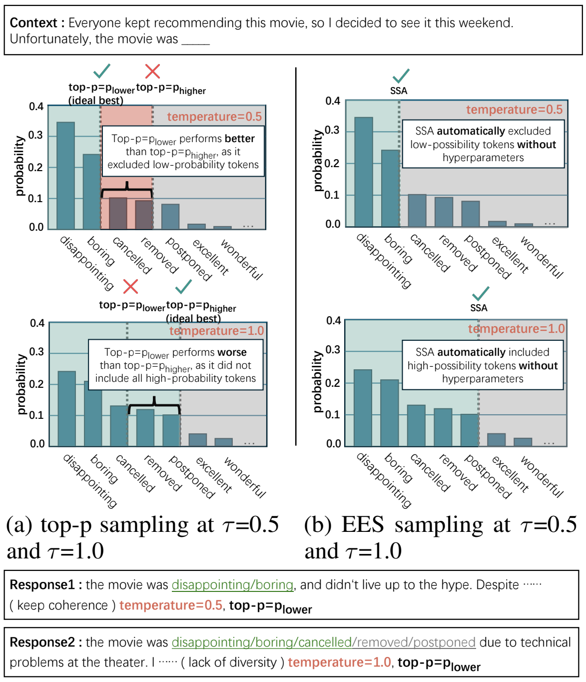
Abstract

Token sampling strategies critically influence text generation quality in large language models (LLMs). However, existing methods introduce additional hyperparameters, requiring extensive tuning and complicating deployment. We present Entropy Equilibrium Sampling (EES), a hyperparameter-free approach inspired by information theory that can dynamically adjust candidate sets by balancing normalized entropy with probability mass. We evaluate EES on both reasoning and generation tasks across a range of model architectures. Our results show that EES consistently performs well across temperature settings, delivering competitive accuracy and coherence while maintaining diversity. By eliminating the need for hyperparameter tuning, EES greatly simplifies deployment while improving performance. Code is released at https://anonymous.4open.science/r/Entropy-Equilibrium-Sampling-B196.

1 Introduction

The rapid advancement of large language models (LLMs) has revolutionized natural language generation, enabling diverse applications ranging from creative writing to scientific reasoning (Brown et al., 2020; Chowdhery et al., 2023; Touvron et al., 2023). As LLMs continue to improve, the quality of generated text depends not only on model architecture and training data, but also greatly on the sampling methods employed during inference (Xing et al., 2024; Zhan et al., 2025a,b).

Modern text generation relies heavily on stochastic sampling methods to balance two fundamental objectives: maintaining coherence with the learned distribution while introducing sufficient diversity to avoid repetitive or overly predictable outputs (Nguyen et al., 2024; Fan et al., 2018; Welleck et al., 2019). This coherence-diversity trade-off has led to the development of numerous sampling techniques, including nucleus sampling (top-p) (Holtzman et al., 2019), typical sampling (Meister et al., 2023), and eta sampling (Hewitt et al., 2022). Each method introduces specialized hyperparameters to control generation behavior (Welleck et al., 2024).



(a) top-p sampling at τ=0.5 and τ=1.0

(b) EES sampling at τ=0.5 and τ=1.0

Figure 1: Hyperparameter sensitivity across temperatures. EES achieves consistent optimal performance without tuning, while top-p requires temperature-specific hyperparameter adjustment.

However, the reliance on hyperparameters presents significant challenges for deployment (Liao et al., 2022). As demonstrated in our experiments (see Appendix G), the generation quality of current sampling methods exhibits substantial variance under different hyperparameter configurations, confirming their high sensitivity to manual tuning. This instability complicates adaptation across diverse domains (Liang et al., 2022), as achieving optimal results often requires extensive, condition-specific adjustments.

To address these fundamental limitations, we introduce *Entropy Equilibrium Sampling (EES)*, a novel, streamlined, hyperparameter-free approach grounded in information theory (Cover, 1999), which maintains high generation quality while simplifying deployment. Our method leverages the insight that competitive sampling can be achieved by balancing normalized entropy and probability mass within dynamically constructed candidate sets. Figure 1 demonstrates the critical challenge of maintaining optimal performance when temperature conditions change. The result reveals that methods like top-p achieve optimal token selection at specific temperatures (e.g., $\tau = 0.5$) through careful hyperparameter tuning but fail to maintain this performance when the temperature changes (e.g., $\tau = 1.0$) without re-tuning their hyperparameters. In contrast, our proposed EES method consistently achieves optimal token truncation across varying temperature conditions without requiring any hyperparameter adjustment.

In summary, the contributions of this paper are as follows:

- **Theoretical analysis for entropy-probability equilibrium**: We provide rigorous theoretical analysis proving the existence and uniqueness of the equilibrium between normalized entropy and probability mass, establishing a solid mathematical foundation for hyperparameter-free sampling methods.
- **Hyperparameter-free sampling method**: We present a method that dynamically adjusts candidate set size based on entropy-probability mass relationships, maintaining effective coherence-diversity balance without hyperparameter tuning.
- **Comprehensive empirical validation**: We conducted extensive experiments across diverse benchmarks, model families, and scales, demonstrating that our method effectively balances coherence and diversity while maintaining robustness across different temperature settings.

2 Preliminary

2.1 Autoregressive Sampling

In autoregressive text generation, given a context sequence $x_{1:t-1}$, a LLM computes a score vector $\mathbf{s} \in \mathbf{R}^{|\mathcal{V}|}$ where each element s_i represents the unnormalized logit for the i-th vocabulary token in vocabulary $\mathcal{V}$ (Vaswani et al., 2017; Radford et al., 2019). This score vector undergoes temperature scaling with hyperparameter $\tau > 0$, followed by softmax normalization to obtain the probability distribution:

$$P(x_t = v_i | x_{1:t-1}) = \frac{\exp(s_i/\tau)}{\sum_{j=1}^{|\mathcal{V}|} \exp(s_j/\tau)}, \quad (1)$$

where different sampling methods then employ various strategies to select tokens from this distribution, balancing between generation coherence and diversity (Fan et al., 2018; Holtzman et al., 2019).

2.2 Normalized Entropy

Normalized entropy measures the degree of uncertainty in probability distributions relative to the maximum possible uncertainty. Given the top-k tokens sorted by descending probability, the normalized entropy is defined as:

$$\bar{H}_k := \frac{H_k}{\log k}, \quad (2)$$

where H_k is the Shannon entropy (Shannon, 1948) of the top-k subset:

$$H_k = -\sum_{i=1}^{k} \hat{p}_i \log \hat{p}_i, \quad (3)$$

and $\hat{p}_i = \frac{p_i}{\sum_{j=1}^{k} p_j}$ represents the renormalized probability of the i-th token within the top-k subset.

The denominator $\log k$ represents the maximum entropy achievable when all k tokens have equal probability ($\hat{p}_i = \frac{1}{k}$ for all i). Thus, $\bar{H}_k \in [0, 1]$, where:

- $\bar{H}_k \to 0$: The distribution is highly concentrated, indicating low uncertainty and high confidence in token selection.
- $\bar{H}_k \to 1$: The distribution approaches uniform, indicating high uncertainty and approximately equal likelihood among tokens.

This normalized entropy measure provides a scale-invariant metric for assessing distributional uncertainty within candidate token sets, enabling consistent comparison across different vocabulary subset sizes.

2.3 Probability Mass

Probability mass measures the cumulative probability concentration within a token subset. Given the top-k tokens ranked by descending probability, probability mass is defined as:

$$P_k := \sum_{i=1}^{k} p_i, \quad (4)$$

in contrast to entropy, which captures the shape of the distribution, probability mass directly reflects candidate subset coverage, making it suitable for truncation thresholds.

3 Method

3.1 Problem Formulation

In LLM sampling, the fundamental challenge lies in balancing exploration and exploitation within the vocabulary probability distribution (Meister and Cotterell, 2021; Brown et al., 2024). This balance is characterized by two key information-theoretic measures: Normalized Entropy, which quantifies the distributional trend of the candidate set and guides how tokens should be utilized, and Probability Mass, which represents the cumulative probability coverage of selected candidates and thus quantifies the scope of exploration.

We formalize this as an optimization problem over vocabulary subsets. Given a conditional probability distribution $P(x_t|x_{<t})$ at generation step t, we seek a candidate set $A^* \subseteq \mathcal{V}$ that optimizes the balance between these two measures:

$$A^*_{x<t} = \arg \max_{A^*_{x<t} \subseteq \mathcal{V}} \mathcal{F}(\bar{H}_k(A^*_{x<t}), \mathrm{P}(A^*_{x<t})), \tag{5}$$

where $\mathcal{F}$ represents the objective function balancing coherence-diversity trade-offs. We propose that an effective balance can be achieved when normalized entropy and probability mass are in equilibrium, suggesting that $\mathcal{F}$ should consider candidate sets where these measures are aligned.

3.2 Algorithm Overview

To implement the equilibrium-based approach outlined in Section 3.1, we instantiate the objective function $\mathcal{F}$ by seeking candidate sets where the normalized entropy $\bar{H}_k$ aligns with the scaled probability mass θP_k, where θ is a scaling coefficient. This alignment establishes a flexible criterion for balancing uncertainty and confidence when identifying effective candidate sets. Since the selection of the candidate set size is inherently discrete, we formulate this as finding the largest threshold k^* such that the normalized entropy remains greater than or equal to the scaled probability mass:

$$k^* = \arg\max_k \left\{ k : \bar{H}_k \geq \theta P_k \right\}. \tag{6}$$

It is worth noting that both $\bar{H}_k$ and P_k naturally span the range $[0, 1]$, making them dual measures of distributional characteristics. The choice of θ significantly influences the behavior of the algorithm. If $\theta < 0$, the condition typically fails to identify a meaningful intersection for truncation, resulting in the fallback use of the entire vocabulary. Conversely, if $\theta > 0$, the algorithm applies stricter truncation. However, for highly uniform distributions, this might cause excessive cutting, where excluded tokens have probabilities almost identical to those retained."

Consequently, we set $\theta = 1$ and simplify the optimization problem to:

$$k^* = \arg\max_k \left\{ k : \bar{H}_k \geq P_k \right\}. \tag{7}$$

Once k^* is determined, we truncate the vocabulary distribution at this threshold, retaining only the top-k^* tokens for subsequent sampling. To validate this configuration, we provide supporting experimental results in Appendix B, demonstrating that $\theta = 1$ yields the most competitive performance.

3.3 Convergence Analysis

We establish the theoretical foundation for our algorithm by proving the existence and uniqueness of the threshold k^*.

Theorem 1. *For any probability distribution $\{p_i\}_{i=1}^n$ sorted in descending order ($p_1 \geq p_2 \geq \cdots \geq p_n > 0$), there exists a unique $k^* \in \{1, 2, \ldots, n\}$ such that the algorithm converges.*

Proof. We define the objective function:

$$f(k) = \bar{H}_k - P_k, \tag{8}$$

where the algorithm seeks the largest k^* such that $f(k^*) \geq 0$. The proof consists of three steps.

Step 1: Monotonicity of the objective function.

We establish that $f(k) = \bar{H}_k - P_k$ is strictly decreasing. Let $g(k) = \bar{H}_k$ denote the normalized entropy. We demonstrate that $g(k+1) \leq g(k)$ for all valid k. For the boundary case $k = 1$, we set $g(1) = 1$ and observe that $g(1) = 1 \geq g(2) = \frac{H_2}{\log 2}$ since $H_2 \leq \log 2$.

For $k \geq 2$, we prove:

$$\frac{H_{k+1}}{\log(k+1)} \leq \frac{H_k}{\log k}. \tag{9}$$

This is equivalent to:

$$H_{k+1} \cdot \log k \leq H_k \cdot \log(k+1). \tag{10}$$

We decompose H_{k+1} as:

$$\begin{aligned} H_{k+1} &= -\sum_{i=1}^{k} \frac{p_i}{P_{k+1}} \log \frac{p_i}{P_{k+1}} - \frac{p_{k+1}}{P_{k+1}} \log \frac{p_{k+1}}{P_{k+1}} \\ &= \frac{P_k}{P_{k+1}} H_k + \frac{P_k}{P_{k+1}} \log \frac{P_{k+1}}{P_k} \\ &\quad - \left(1 - \frac{P_k}{P_{k+1}}\right) \log \left(1 - \frac{P_k}{P_{k+1}}\right). \end{aligned} \tag{11}$$

Defining $\alpha = \frac{P_k}{P_{k+1}}$, we obtain:

$$H_{k+1} = \alpha H_k + h(\alpha), \tag{12}$$

where $h(\alpha) = -\alpha \log \alpha - (1-\alpha)\log(1-\alpha)$ is the binary entropy function.

Since $\{p_i\}_{i=1}^n$ is sorted in descending order, we have $\alpha = \frac{P_k}{P_{k+1}} \geq \frac{k}{k+1}$.

When $\alpha = \frac{k}{k+1}$ (uniform case), we have $p_1 = p_2 = \cdots = p_{k+1}$, yielding $H_k = \log k$ and $H_{k+1} = \log(k+1)$, so inequality (10) holds with equality.

When $\alpha > \frac{k}{k+1}$, substituting eq. (12) into eq. (10) and rearranging:

$$H_k \cdot (\log(k+1) - \alpha \log k) \geq \log k \cdot h(\alpha). \tag{13}$$

Define $\phi(\alpha) = H_k \cdot (\log(k+1) - \alpha \log k) - \log k \cdot h(\alpha)$. Taking the derivative:

$$\phi'(\alpha) = -\log k \cdot \left(H_k + \log \frac{1-\alpha}{\alpha}\right). \tag{14}$$

For $\alpha > \frac{k}{k+1}$, we have $\frac{1-\alpha}{\alpha} < \frac{1}{k}$, implying $\log\left(\frac{1-\alpha}{\alpha}\right) < -\log k$. Therefore:

$$\phi'(\alpha) > -\log k \cdot (H_k - \log k) \geq 0, \tag{15}$$

where the inequality follows from $H_k \leq \log k$.

Since $\phi\left(\frac{k}{k+1}\right) = 0$ and $\phi'(\alpha) > 0$ for $\alpha \in \left(\frac{k}{k+1}, 1\right)$, we have $\phi(\alpha) > 0$ in this interval, establishing inequality (13).

Therefore, $g(k)$ is strictly decreasing. Since P_k is strictly increasing, $f(k) = g(k) - P_k$ is strictly decreasing.

Step 2: Boundary behavior analysis.

We examine the behavior of $f(k)$ at the boundaries to establish the existence of a zero crossing.

For $k = 1$: Since we define $g(1) = 1$ and $P_1 = p_1 \leq 1$, we have $f(1) = 1 - p_1 \geq 0$.

As k approaches the vocabulary size n: We have $P_n = 1$ and $\bar{H}_n \leq 1$, with strict inequality for non-uniform distributions due to entropy concavity. Thus:

$$\lim_{k \to n} f(k) = \bar{H}_n - 1 \leq 0. \tag{16}$$

Step 3: Existence and uniqueness.

Since $f(k)$ is strictly decreasing on the discrete domain $\{1, 2, \ldots, n\}$, with $f(1) \geq 0$ and $f(n) \leq 0$, there exists a unique threshold k^* defined as:

$$k^* = \arg\max_k \{k : f(k) \geq 0\}. \tag{17}$$

The strict monotonicity of $f(k)$ ensures that:

- If $k^* < n$, then $f(k^*) \geq 0$ and $f(k^*+1) < 0$
- If $k^* = n$, then $f(n) \geq 0$ (uniform distribution case)

This guarantees the uniqueness of the threshold and ensures our algorithm converges to this well-defined solution. □

3.4 Algorithm Implementation

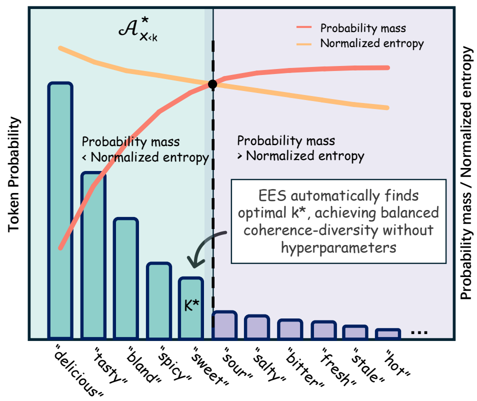


Figure 2: Mechanism of EES

Figure 2 illustrates the mechanism, showing how the algorithm determines the threshold k^* through iterative comparison of normalized entropy and probability mass. Algorithm 1 presents the complete algorithmic procedure for EES, which dynamically selects the sampling threshold at each generation step by maintaining an entropy-probability mass equilibrium condition.

Complexity Analysis. The computational complexity per generation step primarily consists of: (1) probability computation ($\mathcal{O}(|\mathcal{V}|)$), (2) sorting ($\mathcal{O}(|\mathcal{V}| \log |\mathcal{V}|)$), (3) threshold selection ($\mathcal{O}(k^2)$ or $\mathcal{O}(k)$ depending on implementation), and (4) sampling ($\mathcal{O}(k)$). Since $k \ll |\mathcal{V}|$ in practice, the naive implementation is dominated by the sorting step, resulting in an overall complexity of $\mathcal{O}(|\mathcal{V}| \log |\mathcal{V}|)$.

LM	Sampling Method	CommonsenseQA				StrategyQA			
		$\tau = 0.5$	$\tau = 0.8$	$\tau = 1.0$	Avg	$\tau = 0.5$	$\tau = 0.8$	$\tau = 1.0$	Avg
Qwen2.5-7B	Temperature	83.70	82.39	82.01	82.70	75.75	75.90	74.67	75.44
	Top-p	$83.77_{(0.75)}$	$82.64_{(0.80)}$	$81.93_{(0.80)}$	82.78	$74.79_{(0.90)}$	$75.87_{(0.85)}$	$75.05_{(0.80)}$	75.24
	Top-k	$82.75_{(10)}$	$82.42_{(100)}$	$82.65_{(20)}$	82.61	$75.87_{(5)}$	$75.11_{(5)}$	$74.44_{(100)}$	75.14
	Eta	$82.98_{(0.0009)}$	$81.88_{(0.002)}$	$82.10_{(0.0006)}$	82.32	$75.43_{(0.0006)}$	$74.99_{(0.0006)}$	$75.11_{(0.002)}$	75.18
	Mirostat	$83.88_{(2.5)}$	$83.23_{(2.5)}$	$83.55_{(2.5)}$	83.55	$75.72_{(2.5)}$	$76.33_{(3.5)}$	$75.37_{(2.5)}$	75.81
	Typical	$83.36_{(0.95)}$	$\mathbf{84.11}_{(0.2)}$	$82.95_{(0.2)}$	83.47	$75.34_{(0.2)}$	$75.57_{(0.2)}$	$75.23_{(0.2)}$	75.38
	Adaptive	$83.62_{(0.0005)}$	$83.46_{(0.005)}$	$83.39_{(0.01)}$	83.49	$75.63_{(0.005)}$	$75.52_{(0.005)}$	$75.46_{(0.005)}$	75.54
	Ours	**84.42**	83.95	**83.64**	**84.00**	**76.33**	**76.56**	**75.95**	**76.28**
Llama3.1-8B	Temperature	76.69	75.71	73.63	75.34	73.80	72.34	69.02	71.72
	Top-p	$76.81_{(0.75)}$	$76.25_{(0.75)}$	$75.28_{(0.75)}$	76.11	$73.54_{(0.95)}$	$73.54_{(0.85)}$	$72.66_{(0.75)}$	73.25
	Top-k	$77.33_{(20)}$	$76.23_{(5)}$	$74.76_{(5)}$	76.11	$\mathbf{75.28}_{(5)}$	$72.20_{(20)}$	$72.52_{(5)}$	73.33
	Eta	$77.17_{(0.0006)}$	$75.22_{(0.0003)}$	$74.25_{(0.004)}$	75.55	$74.06_{(0.002)}$	$71.99_{(0.0003)}$	$70.39_{(0.0009)}$	72.15
	Mirostat	$77.20_{(2.5)}$	$76.38_{(3.0)}$	$76.54_{(2.5)}$	76.71	$74.12_{(4.0)}$	$\mathbf{75.46}_{(3.0)}$	$74.50_{(4.0)}$	74.69
	Typical	$77.22_{(0.2)}$	$75.81_{(0.2)}$	$75.18_{(0.2)}$	76.07	$75.05_{(0.2)}$	$73.57_{(0.2)}$	$72.02_{(0.90)}$	73.55
	Adaptive	$77.17_{(0.01)}$	$\mathbf{77.38}_{(0.0005)}$	$\mathbf{77.20}_{(0.01)}$	**77.25**	$75.11_{(0.0005)}$	$75.17_{(0.01)}$	$\mathbf{75.14}_{(0.0005)}$	**75.14**
	Ours	**77.44**	76.72	76.90	77.02	75.14	74.50	74.61	74.75
Qwen2.5-32B	Temperature	88.98	88.27	88.52	88.59	80.29	79.39	79.04	79.57
	Top-p	$88.63_{(0.90)}$	$88.21_{(0.95)}$	$88.19_{(0.85)}$	88.34	$\mathbf{80.44}_{(0.80)}$	$79.42_{(0.95)}$	$79.24_{(0.80)}$	79.70
	Top-k	$88.83_{(10)}$	$\mathbf{89.12}_{(5)}$	$88.11_{(50)}$	88.69	$79.77_{(5)}$	$80.38_{(5)}$	$78.57_{(100)}$	79.57
	Eta	$88.99_{(0.002)}$	$88.48_{(0.002)}$	$88.47_{(0.002)}$	88.65	$80.15_{(0.0009)}$	$78.95_{(0.004)}$	$79.39_{(0.004)}$	79.50
	Mirostat	$89.24_{(3.0)}$	$88.75_{(4.0)}$	$\mathbf{89.09}_{(2.5)}$	**89.03**	$80.32_{(3.5)}$	$80.12_{(2.5)}$	$80.15_{(3.5)}$	80.20
	Typical	$89.19_{(0.2)}$	$88.85_{(0.2)}$	$88.70_{(0.92)}$	88.91	$80.06_{(0.90)}$	$80.32_{(0.92)}$	$79.45_{(0.92)}$	79.94
	Adaptive	$88.98_{(0.0005)}$	$88.76_{(0.01)}$	$88.83_{(0.001)}$	88.86	$80.32_{(0.0005)}$	$\mathbf{80.79}_{(0.0005)}$	$\mathbf{80.99}_{(0.01)}$	**80.70**
	Ours	**89.30**	88.85	88.94	**89.03**	80.35	80.52	80.23	80.37

Table 1: Cross-model performance comparison of sampling methods under different temperature settings on two QA datasets. Bold: best performance; underlined: second-best; parentheses: optimal hyperparameter.

However, in Appendix A, we introduce an optimized implementation that eliminates the need for full sorting, reducing the overall complexity to $\mathcal{O}(|\mathcal{V}|)$ and provide experimental validation to demonstrate the practical efficiency and feasibility of this algorithm.

Algorithm 1 Entropy Equilibrium Sampling (EES)

Require: Context $\mathcal{C}$; Language Model P_θ
Ensure: Generated Sequence $\mathcal{G}$

```
1:  G ← ∅
2:  for each generation step do
3:      p ← Sort(P_θ(·|C), descending)
4:      P_0 ← 0, k* ← 1
5:      for k = 1 to |V| do
6:          P_k ← P_{k−1} + p_k
7:          if k ≥ 2 then
8:              H̄_k ← −Σ_{i=1}^{k} (p_i/P_k) log(p_i/P_k) / log k
9:              if H̄_k < P_k then
10:                 k* ← k − 1; break
11:             end if
12:         end if
13:     end for
14:     x_t ← sample from top-k* distribution
15:     Update C and G with x_t
16: end for
17: return G
```

4 Experiment

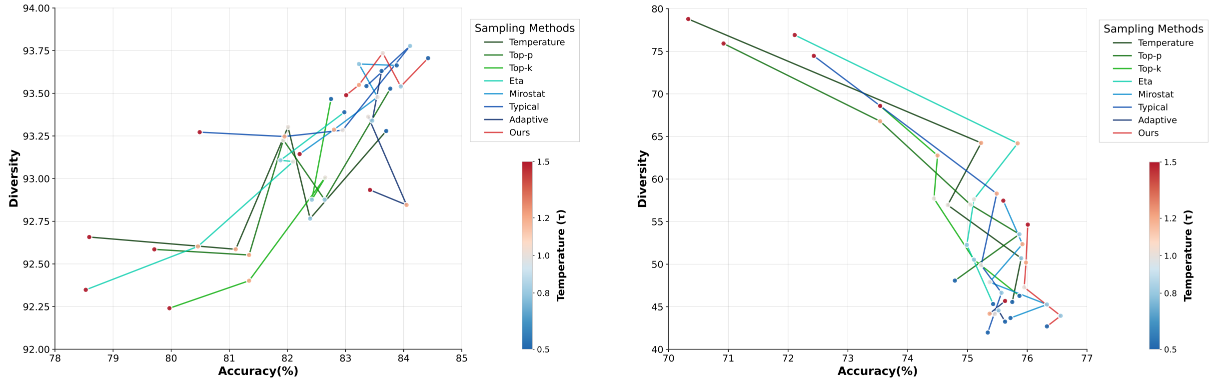
4.1 Setups

Models and Datasets We evaluated our method using Qwen2.5-7B, Qwen2.5-32B (Yang et al., 2025), and Llama3.1-8B (Dubey et al., 2024) on CommonsenseQA (Talmor et al., 2018) and StrategyQA (Geva et al., 2021). Detailed statistical information regarding all datasets is provided in Appendix D.

Baseline We compared against seven established sampling methods: temperature sampling, top-p, top-k sampling, eta sampling, mirostat, typical sampling, and adaptive sampling.

Experimental Protocol We randomly sampled 20% of the training data as a validation set to identify optimal hyperparameters for each sampling method across different temperature settings. We configured the hyperparameter search space for each method based on recommendations from the original papers and subsequent literature (Shi et al., 2024):

- **Top-p**: $p \in \{0.75, 0.8, 0.85, 0.9, 0.95\}$
- **Top-k**: $k \in \{5, 10, 20, 50, 100\}$
- **Eta sampling**: $\eta \in \{3 \times 10^{-4}, \ldots, 4 \times 10^{-3}\}$
- **Mirostat**: $\tau \in \{2.5, 3.0, 3.5, 4.0\}$
- **Typical sampling**: $p \in \{0.2, 0.9, 0.92, 0.95\}$
- **Adaptive sampling**: $\epsilon \in \{5 \times 10^{-4}, \ldots, 1 \times 10^{-2}\}$



(a) CommonsenseQA

(b) StrategyQA

Figure 3: Accuracy-diversity performance across sampling methods and temperatures on two QA datasets using Llama3.1-8B.

For each configuration, we conducted three independent runs and selected the hyperparameter setting that achieves the highest average accuracy on the validation set. On the test set, we evaluated each method using its optimal hyperparameter with five independent runs, reporting average performance as the final score. Following (Shi et al., 2024), we constructed model inputs by randomly sampling multiple examples from the training set to provide in-context learning examples.

4.2 Results

Table 1 reveals that our EES method consistently achieves competitive performance, ranking among the top methods across all experimental configurations while demonstrating robust performance across varying temperature settings and model architectures. While some baseline methods occasionally achieve slightly higher individual scores (e.g., Adaptive sampling on Llama3.1-8B), our approach offers a distinct advantage in practical deployment scenarios.

Most significantly, our method eliminates hyperparameter sensitivity: baseline methods require diverse optimal configurations across different models and temperatures, with parameters varying dramatically between settings (e.g., top-k from 5 to 100, Eta from 0.0003 to 0.004), whereas our approach maintains consistent performance without any hyperparameter adjustment. Notably, we observed a clear scaling effect where performance variance between methods decreases substantially as model size increases—accuracy differences compress from 2-3% gaps in smaller models to $<1\%$ in the 32B parameter regime, suggesting improved calibration in larger models.

4.3 Diversity analysis

We evaluated diversity using the repetition-based metric $Diversity = \prod_{n=2}^{4}(1.0 - \frac{\text{rep-}n}{100})$ from (Meister et al., 2023) on correctly classified responses from Qwen2.5-7B. Figure 3 demonstrates that our method achieves competitive diversity while maintaining superior accuracy across all temperature settings. Notably, we observed that the presumed accuracy-diversity trade-off is highly task-dependent: on CommonsenseQA (Figure 3a), all sampling methods reach high diversity scores ($>93\%$) at $\tau = 0.5$, with higher temperatures providing negligible diversity gains while significantly degrading accuracy. This suggests that task complexity fundamentally modulates the optimal operating point for sampling strategies, and that the traditional assumption of a universal accuracy-diversity trade-off may not hold across different reasoning tasks.

4.4 Mathematical Reasoning

We evaluated mathematical reasoning performance on the GSM8K dataset (Cobbe et al., 2021) using Qwen3-8B (Yang et al., 2025). We examined two distinct generation modes: **w/o thinking** (standard direct prompting) and **w/ thinking** (reasoning mode), across temperature settings $\tau \in \{0.5, 0.8, 1.0\}$ to assess robustness. We adhered to the protocol in Section 4.1, using the same hyperparameter search pipeline. Final results represent the average of 5 independent runs. Each evaluation involves generating the full reasoning path and final answer, with correctness determined by exact match with the ground truth.

LM	Sampling Method	GSM8K (w/o thinking)				GSM8K (w/ thinking)			
		$\tau = 0.5$	$\tau = 0.8$	$\tau = 1.0$	**Avg**	$\tau = 0.5$	$\tau = 0.8$	$\tau = 1.0$	**Avg**
Qwen3-8B	Temperature	61.49	61.83	61.91	61.74	66.87	66.55	66.16	66.53
	Top-p	62.64$_{(0.85)}$	62.38$_{(0.95)}$	61.61$_{(0.85)}$	62.21	<u>66.94</u>$_{(0.90)}$	66.64$_{(0.75)}$	66.31$_{(0.80)}$	66.63
	Top-k	61.93$_{(50)}$	62.11$_{(20)}$	61.59$_{(20)}$	61.88	66.50$_{(50)}$	66.10$_{(100)}$	66.26$_{(100)}$	66.29
	Eta	62.58$_{(0.004)}$	62.30$_{(3e-4)}$	<u>62.44</u>$_{(0.004)}$	62.44	66.72$_{(0.002)}$	<u>66.75</u>$_{(3e-4)}$	<u>66.70</u>$_{(0.004)}$	**66.72**
	Mirostat	<u>62.71</u>$_{(4.0)}$	<u>62.52</u>$_{(4.0)}$	**62.70**$_{(3.0)}$	**62.64**	66.61$_{(3.0)}$	66.64$_{(3.0)}$	66.55$_{(4.0)}$	66.60
	Typical	62.56$_{(0.2)}$	62.00$_{(0.9)}$	62.09$_{(0.9)}$	62.22	**66.96**$_{(0.9)}$	66.35$_{(0.2)}$	66.54$_{(0.2)}$	66.62
	Adaptive	62.68$_{(5e-4)}$	61.93$_{(0.005)}$	62.08$_{(0.001)}$	62.23	66.49$_{(0.005)}$	66.73$_{(5e-4)}$	66.22$_{(0.01)}$	66.48
	Ours	**62.85**	**62.55**	62.18	<u>62.53</u>	66.64	**66.76**	**66.75**	**66.72**

Table 2: Comparison of sampling methods on **GSM8K** using **Qwen3-8B**, evaluating performance **w/o thinking** and **w/ thinking**. Values represent accuracy (%), with optimal hyperparameters in parentheses.

Results. Table 2 demonstrates that our method achieves superior performance, consistently ranking first or second across the majority of model configurations. In the **w/o thinking** mode, our method dominates at lower temperatures ($\tau = 0.5, 0.8$) and secures the second-best average accuracy.

Crucially, in the **w/ thinking** mode, our approach exhibits remarkable robustness against temperature variations. While baseline methods like Top-p and Typical sampling achieve high accuracy at low temperatures, they suffer noticeable degradation at $\tau = 1.0$ (e.g., Top-p drops from 66.94% to 66.31%). In contrast, our method maintains consistently high performance, achieving the best result at $\tau = 1.0$ (66.75%) and tying for the highest average accuracy (66.72%). This stability highlights the effectiveness of our entropy-equilibrium strategy in filtering irrational tokens during complex reasoning paths, offering a reliable, hyperparameter-free solution that performs optimally across varying conditions where other methods require extensive tuning.

4.5 Creative Writing Evaluation

To assess the generation quality of our proposed method in open-ended text generation, we conducted experiments on the WikiText-103 dataset (Merity et al., 2016). We utilized the Qwen2.5-7B and Llama3.1-8B models, with generation performed at an elevated temperature of $\tau = 1.0$ to promote diversity. Each generational task utilized 32-word prefixes as prompts, with model outputs limited to 256 tokens. All hyperparameters of baseline methods were optimized on a 200-sample validation set by maximizing the MAUVE score (Pillutla et al., 2021) against human references.

Instead of relying on automatic metrics, which often fail to align with human preferences (Nimah et al., 2023), we adopted a pairwise comparison framework (Zhu et al., 2024) involving both human annotators and a LLM ensemble as judges. Following (Nguyen et al., 2024), we conducted evaluations across two key dimensions: (1) **Quality**, covering coherence, relevance to the prompt, and overall fluency; and (2) **Diversity**, assessing the creativity and distinctiveness of the content.

Evaluation Protocol. For human evaluation, we recruited 10 diverse, fluent English speakers. Each participant evaluated 350 text pairs (50 per baseline) on a 3-point Likert scale: +1 indicates our method is clearly superior, -1 indicates the baseline is superior, and 0 indicates comparable quality. We excluded temperature sampling due to its poor generation quality but included human-written reference texts from WikiText-103 as a gold-standard benchmark. Detailed information regarding the evaluation interface, participant demographics, and compensation is provided in Appendix F.

For automated evaluation, we employed a suite of state-of-the-art LLM as judges, including Claude-4.0 (Anthropic, 2025a), Claude-4.5 (Anthropic, 2025b), GPT-4.1 (OpenAI, 2025), and Gemini-2.5-Pro (DeepMind, 2025). We utilized carefully designed prompts following (Nguyen et al., 2024) to leverage their demonstrated capability in aligning with human judgment.

Results. Table 3 summarizes the findings. Our method achieves robust **Quality**, securing positive win-loss ratios against the majority of baselines in both evaluations. Human evaluators frequently rated our text as equivalent to strong baselines like Top-p (54.2% tie rate), while the LLM ensemble favored our approach in most comparisons, with the exception of Top-p. In **Diversity**, our method significantly outperforms Adaptive sampling across both assessment types, though results against Eta were mixed in automated evaluation. Notably, against human references, the high human

Comparison	Human Evaluators (%)			LLM Ensemble (%)		
	Win	Tie	Loss	Win	Tie	Loss
Quality Evaluation						
vs Human Ref.	9.20	68.40	22.40	2.00	2.00	96.00
vs Top-p	**24.60**	54.20	21.20	44.50	7.50	48.00
vs Top-k	**23.80**	57.60	18.60	**46.00**	9.50	44.50
vs Eta	**44.20**	38.40	17.40	**48.00**	7.00	45.00
vs Mirostat	**25.40**	58.20	16.40	**53.50**	17.50	29.00
vs Typical	**34.20**	40.60	25.20	**61.00**	5.50	33.50
vs Adaptive	**38.40**	52.80	8.80	**56.50**	18.00	25.50
Diversity Evaluation						
vs Human Ref.	14.80	30.20	55.00	3.00	9.00	88.00
vs Top-p	21.60	31.40	47.00	20.00	22.00	58.00
vs Top-k	28.40	27.60	44.00	31.00	20.50	48.50
vs Eta	**42.80**	33.20	24.00	25.00	24.50	50.50
vs Mirostat	**38.60**	26.40	35.00	**40.50**	29.50	30.00
vs Typical	32.40	18.60	49.00	32.50	25.00	42.50
vs Adaptive	**46.20**	35.40	18.40	**51.00**	30.00	19.00

Table 3: Pairwise comparison results from Human evaluators and LLM judges.

tie rate (68.4%) suggests our method approaches human-level coherence, confirming its superior performance in open-ended text generation tasks.

5 Related Works

5.1 Deterministic Methods

Deterministic sampling methods provide reproducible outputs and strong coherence, but suffer from limited diversity due to their conservative selection strategies. Greedy sampling selects the highest probability token at each step (Holtzman et al., 2019). Beam search (Freitag and Al-Onaizan, 2017) maintains multiple candidate sequences but still tends to generate bland text due to its preference for high-probability paths (Vijayakumar et al., 2016). Contrastive Search (Su et al., 2022) balances model confidence with lexical diversity penalties, while Frustratingly Simple sampling (Yang et al., 2023) uses auxiliary anti-language models to guide generation. DoLa (Chuang et al., 2023) leverages layer-wise representational differences in transformers for improved sampling.

5.2 Stochastic Methods

Stochastic sampling methods aim to identify an appropriate truncation point to construct a candidate set from the original vocabulary, followed by probabilistic sampling within this subset. The widely adopted top-k (Fan et al., 2018) and top-p (Holtzman et al., 2019) sampling methods employ fixed truncation criteria based on candidate set size and probability mass, respectively. Recent methods introduce adaptive sampling strategies that dynamically adjust candidate sets based on distribution characteristics. η-sampling (Hewitt et al., 2022) uses entropy thresholds to control candidate set size based on prediction uncertainty. Mirostat sampling (Basu et al., 2020) maintains text perplexity within a predetermined range by adaptively selecting tokens based on the current generation's perplexity. Typical sampling (Meister et al., 2023) ranks vocabulary tokens by the divergence between their individual probabilities and the distribution's entropy, then applies a coverage threshold to determine the candidate set. Adaptive sampling (Zhu et al., 2024) uses entropy-based confidence metrics to determine token inclusion in the candidate set during generation. Min-p (Nguyen et al., 2024) dynamically adjusts sampling thresholds by utilizing the top token's probability as a scaling factor.

6 Conclusion

We presented Entropy Equilibrium Sampling (EES), a hyperparameter-free sampling strategy that dynamically adapts candidate sets by balancing normalized entropy and probability mass. EES achieves robust performance across varying temperatures without manual tuning. Extensive evaluations on mathematical reasoning and open-ended generation show that EES consistently matches or outperforms baselines, particularly in maintaining logical consistency under high-temperature settings. Future work will focus on optimizing inference latency, extending the framework to multimodal generation, and investigating its interaction with model calibration techniques.

Limitations

We identify two primary limitations of our method:

Inference Latency. Although the theoretical time complexity of EES can be reduced to $\mathcal{O}(|\mathcal{V}|)$ (as detailed in Appendix A), the practical runtime overhead remains a challenge. Current GPU implementations for dynamic thresholding operations (such as iterative summation or heap-based selection) are not as highly optimized as the mature kernels used for static Top-k or Top-p sampling. Consequently, despite the theoretical efficiency, the actual inference latency is higher than that of static methods, which may impact throughput in latency-sensitive applications.

Dependence on Model Calibration. Our framework assumes the model's probability distribution accurately reflects semantic uncertainty. Since LLMs are prone to miscalibration (e.g., overconfidence), the derived entropy may occasionally mislead the truncation mechanism, leading to suboptimal candidate sets. Future work could investigate integrating calibration techniques to mitigate this issue.

References

Anthropic. 2025a. Claude 4 technical report. https://www.anthropic.com/news/claude-4. Accessed: 2026-01-06.

Anthropic. 2025b. Claude 4.5: Pushing the boundaries of ai reasoning. https://www.anthropic.com/research/claude-4-5. Accessed: 2026-01-06.

Sourya Basu, Govardana Sachitanandam Ramachandran, Nitish Shirish Keskar, and Lav R Varshney. 2020. Mirostat: A neural text decoding algorithm that directly controls perplexity. *arXiv preprint arXiv:2007.14966*.

Bradley Brown, Jordan Juravsky, Ryan Ehrlich, Ronald Clark, Quoc V Le, Christopher Ré, and Azalia Mirhoseini. 2024. Large language monkeys: Scaling inference compute with repeated sampling. *arXiv preprint arXiv:2407.21787*.

Tom Brown, Benjamin Mann, Nick Ryder, Melanie Subbiah, Jared D Kaplan, Prafulla Dhariwal, Arvind Neelakantan, Pranav Shyam, Girish Sastry, Amanda Askell, and 1 others. 2020. Language models are few-shot learners. *Advances in neural information processing systems*, 33:1877–1901.

Aakanksha Chowdhery, Sharan Narang, Jacob Devlin, Maarten Bosma, Gaurav Mishra, Adam Roberts, Paul Barham, Hyung Won Chung, Charles Sutton, Sebastian Gehrmann, and 1 others. 2023. Palm: Scaling language modeling with pathways. *Journal of Machine Learning Research*, 24(240):1–113.

Yung-Sung Chuang, Yujia Xie, Hongyin Luo, Yoon Kim, James Glass, and Pengcheng He. 2023. Dola: Decoding by contrasting layers improves factuality in large language models. *arXiv preprint arXiv:2309.03883*.

Karl Cobbe, Vineet Kosaraju, Mohammad Bavarian, Mark Chen, Heewoo Jun, Lukasz Kaiser, Matthias Plappert, Jerry Tworek, Jacob Hilton, Reiichiro Nakano, and 1 others. 2021. Training verifiers to solve math word problems. *arXiv preprint arXiv:2110.14168*.

Thomas M Cover. 1999. *Elements of information theory*. John Wiley & Sons.

Google DeepMind. 2025. Gemini 2.5: Highly capable multimodal models. https://deepmind.google/technologies/gemini/. Accessed: 2026-01-06.

Abhimanyu Dubey, Abhinav Jauhri, Abhinav Pandey, Abhishek Kadian, Ahmad Al-Dahle, Aiesha Letman, Akhil Mathur, Alan Schelten, Amy Yang, Angela Fan, and 1 others. 2024. The llama 3 herd of models. *arXiv e-prints*, pages arXiv–2407.

Angela Fan, Mike Lewis, and Yann Dauphin. 2018. Hierarchical neural story generation. *arXiv preprint arXiv:1805.04833*.

Markus Freitag and Yaser Al-Onaizan. 2017. Beam search strategies for neural machine translation. *arXiv preprint arXiv:1702.01806*.

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.

John Hewitt, Christopher D Manning, and Percy Liang. 2022. Truncation sampling as language model desmoothing. *arXiv preprint arXiv:2210.15191*.

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

Percy Liang, Rishi Bommasani, Tony Lee, Dimitris Tsipras, Dilara Soylu, Michihiro Yasunaga, Yian Zhang, Deepak Narayanan, Yuhuai Wu, Ananya Kumar, and 1 others. 2022. Holistic evaluation of language models. *arXiv preprint arXiv:2211.09110*.

Lizhi Liao, Heng Li, Weiyi Shang, and Lei Ma. 2022. An empirical study of the impact of hyperparameter tuning and model optimization on the performance properties of deep neural networks. *ACM Transactions on Software Engineering and Methodology (TOSEM)*, 31(3):1–40.

Clara Meister and Ryan Cotterell. 2021. Language model evaluation beyond perplexity. *arXiv preprint arXiv:2106.00085*.

Clara Meister, Tiago Pimentel, Gian Wiher, and Ryan Cotterell. 2023. Locally typical sampling. *Transactions of the Association for Computational Linguistics*, 11:102–121.

Stephen Merity, Caiming Xiong, James Bradbury, and Richard Socher. 2016. Pointer sentinel mixture models. *arXiv preprint arXiv:1609.07843*.

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*.

Iftitahu Nimah, Meng Fang, Vlado Menkovski, and Mykola Pechenizkiy. 2023. Nlg evaluation metrics beyond correlation analysis: An empirical metric preference checklist. In *Proceedings of the 61st Annual Meeting of the Association for Computational Linguistics (Volume 1: Long Papers)*, pages 1240–1266.

OpenAI. 2025. Gpt-4.1 system card. `https://openai.com/research/gpt-4-1`. Accessed: 2026-01-06.

Krishna Pillutla, Swabha Swayamdipta, Rowan Zellers, John Thickstun, Sean Welleck, Yejin Choi, and Zaid Harchaoui. 2021. Mauve: Measuring the gap between neural text and human text using divergence frontiers. *Advances in Neural Information Processing Systems*, 34:4816–4828.

Alec Radford, Jeffrey Wu, Rewon Child, David Luan, Dario Amodei, Ilya Sutskever, and 1 others. 2019. Language models are unsupervised multitask learners. *OpenAI blog*, 1(8):9.

Claude E Shannon. 1948. A mathematical theory of communication. *The Bell system technical journal*, 27(3):379–423.

Chufan Shi, Haoran Yang, Deng Cai, Zhisong Zhang, Yifan Wang, Yujiu Yang, and Wai Lam. 2024. A thorough examination of decoding methods in the era of llms. *arXiv preprint arXiv:2402.06925*.

Yixuan Su, Tian Lan, Yan Wang, Dani Yogatama, Lingpeng Kong, and Nigel Collier. 2022. A contrastive framework for neural text generation. *Advances in Neural Information Processing Systems*, 35:21548–21561.

Alon Talmor, Jonathan Herzig, Nicholas Lourie, and Jonathan Berant. 2018. Commonsenseqa: A question answering challenge targeting commonsense knowledge. *arXiv preprint arXiv:1811.00937*.

Hugo Touvron, Thibaut Lavril, Gautier Izacard, Xavier Martinet, Marie-Anne Lachaux, Timothée Lacroix, Baptiste Rozière, Naman Goyal, Eric Hambro, Faisal Azhar, and 1 others. 2023. Llama: Open and efficient foundation language models. *arXiv preprint arXiv:2302.13971*.

Ashish Vaswani, Noam Shazeer, Niki Parmar, Jakob Uszkoreit, Llion Jones, Aidan N Gomez, Łukasz Kaiser, and Illia Polosukhin. 2017. Attention is all you need. *Advances in neural information processing systems*, 30.

Ashwin K Vijayakumar, Michael Cogswell, Ramprasath R Selvaraju, Qing Sun, Stefan Lee, David Crandall, and Dhruv Batra. 2016. Diverse beam search: Decoding diverse solutions from neural sequence models. *arXiv preprint arXiv:1610.02424*.

Sean Welleck, Amanda Bertsch, Matthew Finlayson, Hailey Schoelkopf, Alex Xie, Graham Neubig, Ilia Kulikov, and Zaid Harchaoui. 2024. From decoding to meta-generation: Inference-time algorithms for large language models. *arXiv preprint arXiv:2406.16838*.

Sean Welleck, Ilia Kulikov, Stephen Roller, Emily Dinan, Kyunghyun Cho, and Jason Weston. 2019. Neural text generation with unlikelihood training. *arXiv preprint arXiv:1908.04319*.

Peng Xing, Yinghui Li, Shirong Ma, Xinnian Liang, Haojing Huang, Yangning Li, Hai-Tao Zheng, Wenhao Jiang, and Ying Shen. 2024. Mitigating catastrophic forgetting in multi-domain chinese spelling correction by multi-stage knowledge transfer framework. *ArXiv*, abs/2402.11422.

An Yang, Anfeng Li, Baosong Yang, Beichen Zhang, Binyuan Hui, Bo Zheng, Bowen Yu, Chang Gao, Chengen Huang, Chenxu Lv, and 1 others. 2025. Qwen3 technical report. *arXiv preprint arXiv:2505.09388*.

Haoran Yang, Deng Cai, Huayang Li, Wei Bi, Wai Lam, and Shuming Shi. 2023. A frustratingly simple decoding method for neural text generation. *arXiv preprint arXiv:2305.12675*.

Shaoxiong Zhan, Yanlin Lai, Ziyu Lu, Dahua Lin, Ziqing Yang, and Fei Tan. 2025a. Mathsmith: Towards extremely hard mathematical reasoning by forging synthetic problems with a reinforced policy. *arXiv preprint arXiv:2508.05592*.

Shaoxiong Zhan, Hai Lin, Hongming Tan, Xiaodong Cai, Hai-Tao Zheng, Xin Su, Zifei Shan, Ruitong Liu, and Hong-Gee Kim. 2025b. Lexsembridge: Fine-grained dense representation enhancement through token-aware embedding augmentation. *arXiv preprint arXiv:2508.17858*.

Wenhong Zhu, Hongkun Hao, Zhiwei He, Yiming Ai, and Rui Wang. 2024. Improving open-ended text generation via adaptive decoding. *arXiv preprint arXiv:2402.18223*.

A Incremental Entropy Computation

A.1 Mathematical Derivation for Optimization

While section 3.4 presents the conceptual framework, practical implementation can avoid redundant entropy computations through incremental updates. We derive the mathematical foundation for this optimization.

Problem Setup: Given a candidate set of size k with cumulative probability $P_k = \sum_{i=1}^{k} p_i$ and entropy $H_k = -\sum_{i=1}^{k} \frac{p_i}{P_k} \log \frac{p_i}{P_k}$, we want to efficiently compute H_{k+1} when adding a new token with probability p_{k+1}.

Theorem A.1 (Incremental Entropy Update): The entropy H_{k+1} for the expanded candidate set can be computed as:

$$\begin{aligned} H_{k+1} = \frac{P_k}{P_{k+1}} H_k + \frac{P_k}{P_{k+1}} \log \frac{P_{k+1}}{P_k} \\ - \frac{p_{k+1}}{P_{k+1}} \log \frac{p_{k+1}}{P_{k+1}} \end{aligned} \tag{18}$$

where $P_{k+1} = P_k + p_{k+1}$.

Proof: Let $\mathbf{p}^{(k)} = \{p_1^{(k)}, \ldots, p_k^{(k)}\}$ where $p_i^{(k)} = \frac{p_i}{P_k}$ be the normalized probabilities for the k-sized candidate set, and $\mathbf{p}^{(k+1)} = \{p_1^{(k+1)}, \ldots, p_{k+1}^{(k+1)}\}$ where $p_i^{(k+1)} = \frac{p_i}{P_{k+1}}$ for the expanded set.

The entropy of the expanded set is:

$$H_{k+1} = -\sum_{i=1}^{k+1} p_i^{(k+1)} \log p_i^{(k+1)} \tag{19}$$

We can decompose this as:

$$\begin{aligned} H_{k+1} = -\sum_{i=1}^{k} p_i^{(k+1)} \log p_i^{(k+1)} \\ - p_{k+1}^{(k+1)} \log p_{k+1}^{(k+1)} \end{aligned} \tag{20}$$

For the first k terms, note that $p_i^{(k+1)} = \frac{p_i}{P_{k+1}} = \frac{p_i}{P_k} \cdot \frac{P_k}{P_{k+1}} = p_i^{(k)} \cdot \frac{P_k}{P_{k+1}}$.

Therefore:

$$-\sum_{i=1}^{k} p_i^{(k+1)} \log p_i^{(k+1)}$$

$$= -\sum_{i=1}^{k} p_i^{(k)} \cdot \frac{P_k}{P_{k+1}} \log \left(p_i^{(k)} \cdot \frac{P_k}{P_{k+1}} \right) \tag{21}$$

$$= -\frac{P_k}{P_{k+1}} \sum_{i=1}^{k} p_i^{(k)} \left[\log p_i^{(k)} + \log \frac{P_k}{P_{k+1}} \right] \tag{22}$$

$$= \frac{P_k}{P_{k+1}} H_k + \frac{P_k}{P_{k+1}} \log \frac{P_{k+1}}{P_k} \underbrace{\sum_{i=1}^{k} p_i^{(k)}}_{=1} \tag{23}$$

$$= \frac{P_k}{P_{k+1}} H_k + \frac{P_k}{P_{k+1}} \log \frac{P_{k+1}}{P_k} \tag{24}$$

The last term is:

$$-p_{k+1}^{(k+1)} \log p_{k+1}^{(k+1)} = -\frac{p_{k+1}}{P_{k+1}} \log \frac{p_{k+1}}{P_{k+1}}$$

Combining all terms yields the desired formula. □

Computational Complexity: This incremental approach reduces the entropy computation from $\mathcal{O}(k)$ to $\mathcal{O}(1)$ per candidate expansion, leading to an overall complexity improvement from $\mathcal{O}(|\mathcal{V}| \log |\mathcal{V}|)$ to $\mathcal{O}(|\mathcal{V}|)$ for threshold selection.

Implementation Note: The code implementation uses the ratio ratio $= \frac{P_k}{P_{k+1}}$ and computes:

- term1 $=$ ratio $\cdot H_k$
- term2 $=$ ratio $\cdot \log \frac{P_{k+1}}{P_k}$
- term3 $= -\frac{p_{k+1}}{P_{k+1}} \log \frac{p_{k+1}}{P_{k+1}}$

such that $H_{k+1} =$ term1 $+$ term2 $+$ term3, which directly corresponds to our derived formula.

A.2 Computational Efficiency Analysis

To assess the practical deployment feasibility of our proposed method, we conducted a comprehensive analysis of inference latency and computational overhead. We compared our *Entropy Equilibrium* sampling method against standard baselines (Temperature, Top-p, Top-k) and advanced dynamic sampling methods (Typical, Eta, Adaptive).

Experimental Setup. The latency evaluation was performed on a high-performance computing node equipped with NVIDIA GPUs using the `Qwen3-8B` model. The evaluation protocol involved generating 512 samples with a batch size of 128 and a maximum generation length of 128 tokens per sample. To ensure robust measurements, we averaged the results across different temperature settings ($T \in \{0.8, 1.0\}$). We measured two key metrics:

- **Total Generation Time**: The average end-to-end wall-clock time per token generation step, including model forward pass and sampling logic.
- **Sampling Overhead**: The specific time consumed by the logits warping process (i.e., the sampling algorithm logic) per step.

Method	Avg. Latency (ms/step)	Sampling Overhead (ms/step)	Relative Slowdown
Temperature (Baseline)	69.65	0.02	1.00×
Top-p	71.54	0.01	1.03×
Top-k	69.38	0.78	1.00×
Typical	72.25	1.26	1.04×
Eta	70.13	26.64	1.01×
Adaptive	118.11	75.25	1.70×
Ours (Entropy Equilibrium)	90.59	47.42	1.30×

Table 4: Inference latency and sampling overhead comparison. *Avg. Latency* represents the total time per generation step. *Relative Slowdown* is calculated relative to the Temperature baseline.

Results and Analysis. Table 4 summarizes the latency results. Standard truncation methods like Top-p and Top-k introduce negligible overhead compared to the baseline Temperature sampling. Among the dynamic sampling methods, **Adaptive Sampling** incurs the highest computational cost, resulting in a 1.70× slowdown (118.11 ms/step vs. 69.65 ms/step), primarily due to its complex iterative candidate selection process.

Our proposed method demonstrates a favorable trade-off between performance and efficiency. While it introduces a moderate overhead due to the entropy calculation and dynamic thresholding (resulting in a total latency of 90.59 ms/step), it is significantly faster than Adaptive Sampling. Specifically, our method reduces the latency by approximately **23%** compared to Adaptive Sampling (90.59 ms vs. 118.11 ms). The 1.30× relative slowdown compared to the simplest baseline is a justifiable cost given the substantial improvements in generation quality and robustness demonstrated in Section 4, making it suitable for practical applications where generation quality is prioritized.

B Ablation Study on Scaling Factor θ

In our proposed method, we established the sampling threshold using a fixed scaling coefficient $\theta = 1$. To validate this design choice and verify the "hyperparameter-free" claim, we conducted an ablation study by generalizing the truncation condition to:

$$k^* = \arg\max_k \left\{k : \bar{H}_k \geq \theta P_k\right\} \quad (25)$$

Here, θ serves as a tunable hyperparameter controlling the strictness of the equilibrium. A smaller θ (< 1) relaxes the constraint, expanding the candidate set (increasing diversity), while a larger θ (> 1) tightens it, enforcing more aggressive truncation (increasing coherence).

We evaluated the performance impact of varying $\theta \in \{0.6, 0.8, 1.0, 1.2, 1.4\}$ on the GSM8K dataset using Qwen3-8B models in the standard prompting mode (w/o thinking).

Model	Accuracy (%) with different θ				
	$\theta = 0.6$	$\theta = 0.8$	$\theta = 1.0$ **(Default)**	$\theta = 1.2$	$\theta = 1.4$
Qwen3-8B	61.48	61.56	**62.18**	62.04	62.02

Table 5: Ablation study of the scaling factor θ on GSM8K. The default setting $\theta = 1.0$ consistently achieves optimal performance, validating the theoretical equilibrium point.

Results. Table 5 demonstrates that deviating from the theoretical equilibrium point ($\theta = 1.0$) yields no performance benefits. Decreasing θ to 0.6 or 0.8 introduces excessive noise by including lower-probability tail tokens, resulting in noticeable accuracy degradation (e.g., Qwen3-8B drops from 62.18% to 61.48%). Conversely, increasing θ to 1.2 or 1.4 overly restricts the candidate set, potentially filtering out valid reasoning steps and causing a slight performance decline (dropping to 62.02%). These empirical findings confirm that the natural equilibrium ($\theta = 1$) represents a robust and optimal operating point, effectively eliminating the need for hyperparameter tuning.

C Hardware and Software Configuration

All experiments were conducted on the following setup:

- **Hardware**: NVIDIA A100 GPUs with 80GB memory

- **Software**: Detailed in `requirements.txt` and source code
- **Operating System**: CentOS Linux 7 (Core)
- **CUDA Version**: 12.2

D Dataset Statistics and Details

We evaluated our method across four diverse datasets covering commonsense reasoning, strategic reasoning, open-ended text generation, and mathematical problem solving. Table 6 summarizes the key statistics of these datasets.

Dataset	Task Type	Train Size	Test/Dev Size
CommonsenseQA	Multiple Choice QA	9,741	1,221
StrategyQA	Boolean QA (Yes/No)	2,290	490
WikiText-103	Open-Ended Generation	28,475 articles	60 articles
GSM8K	Mathematical Reasoning	7,473	1,319

Table 6: Summary of dataset statistics used in our experiments.

CommonsenseQA (Talmor et al., 2018). A multiple-choice question answering dataset that requires different types of commonsense knowledge to predict the correct answer from five choices. It is constructed from ConceptNet and tests the model's ability to perform broad semantic reasoning.

StrategyQA (Geva et al., 2021). A question answering benchmark focusing on implicit reasoning steps. The questions are open-domain and require the model to infer a multi-step strategy to derive the correct Yes/No answer, challenging the model's logical planning capabilities.

WikiText-103 (Merity et al., 2016). A large-scale benchmark for language modeling, consisting of over 100 million tokens extracted from verified Good and Featured articles on Wikipedia. We use this dataset to evaluate the coherence, fluency, and diversity of the generated text in open-ended generation scenarios.

GSM8K (Cobbe et al., 2021). A dataset of 8.5K high-quality linguistically diverse grade school math word problems. Each problem requires multi-step reasoning to reach the correct numerical answer. We use the standard test set of 1,319 examples to assess the model's mathematical reasoning capabilities.

D.1 Data Splits and Hyperparameter Tuning

To ensure a fair and robust comparison, we established specific protocols for hyperparameter optimization and test set construction across all datasets.

QA and Reasoning Tasks. For **CommonsenseQA**, **StrategyQA**, and **GSM8K**, we adopted a rigorous tuning strategy. We randomly sampled 20% of the original training set to serve as a held-out validation set. All baseline hyperparameters (e.g., k for Top-k, p for Top-p, ϵ for Eta) were optimized on this validation split to maximize task accuracy before being evaluated on the standard test sets.

WikiText-103. For the open-ended generation task, we performed additional preprocessing to ensure sufficient context for coherent generation. We filtered the dataset to retain only articles with a length exceeding 256 tokens, resulting in a curated subset of 791 examples.

- **Validation Set:** We randomly selected 200 examples from this subset to serve as the validation set for tuning baseline hyperparameters.
- **Test Set:** The remaining 591 examples constituted the test pool for final evaluation.
- **Evaluation Sampling:** For the pairwise comparison (Human and LLM evaluation), we randomly sampled 50 generated responses from each baseline's test outputs and paired them with the corresponding responses generated by our method. These pairs were then presented to the evaluators in a blinded manner to assess Quality and Diversity.

E Generated Text Examples

E.1 CommonsenseQA Examples

We present examples from CommonsenseQA to demonstrate the quality differences between sampling methods. The evaluation follows a few-shot learning setup where models are provided with 5 demonstration examples before the target question.

Input Context: The model receives 5 in-context learning examples (abbreviated as "5 samples" below), followed by the target question requiring completion.

Target Question: A revolving door is convenient for two direction travel, but it also serves as a security measure at a what? Answer Choices: (a) bank (b) library (c) department store (d) mall (e) new york

Complete Prompt Structure:

> Question: What do people use to absorb extra ink from a fountain pen? Answer Choices: (a) shirt pocket (b) calligrapher's hand (c) inkwell (d) desk drawer (e) blotter
>
> Answer: The answer must be an item that can absorb ink. Of the above choices, only blotters are used to absorb ink. So the answer is (e).
>
> [4 additional examples omitted for brevity]
>
> Question: A revolving door is convenient for two direction travel, but it also serves as a security measure at a what? Answer Choices: (a) bank (b) library (c) department store (d) mall (e) new york
>
> Answer:

Generated Responses:
Top-p (p=0.8, τ=1.0):

> "The answer should be a place where a revolving door would serve a security purpose. Among the options, a bank (a), library (b), department store (c), and mall (d) can all have revolving doors. However, banks specifically handle large amounts of money and therefore have stricter security measures. So, the best answer is (a) bank."

EES (τ=1.0):

> "A revolving door can serve as a security measure by controlling access and exit, which is particularly important in places where security is a concern. Among the given options, a bank (a) is a place where security is paramount due to the handling of money and valuables. Therefore, the answer is (a) bank."

Evaluation Protocol: Following standard practice, we extract the first complete answer from each generated response for evaluation purposes.

Analysis: Both methods correctly identify the answer as (a) bank. However, EES provides a more concise and focused explanation, directly addressing the security aspect without unnecessary elaboration on alternative options. The EES response demonstrates superior coherence by maintaining focus on the core reasoning path, while the top-p response includes potentially distracting analysis of multiple options before reaching the conclusion.

E.2 StrategyQA Examples

We present examples from StrategyQA to demonstrate reasoning quality differences between sampling methods. StrategyQA requires multi-step reasoning to answer yes/no questions.

Input Context: The model receives 4 in-context learning examples demonstrating the reasoning pattern, followed by the target question requiring completion.

Target Question: Was ship that recovered Apollo 13 named after a World War II battle?

Complete Prompt Structure:

> Question: Do hamsters provide food for any animals?
>
> Answer: Hamsters are prey animals. Prey are food for predators. Thus, hamsters provide food for some animals. So the answer is yes.
>
> [3 additional examples omitted for brevity]
>
> Question: Was ship that recovered Apollo 13 named after a World War II battle?
>
> Answer:

Generated Responses:
Top-p (p=0.8, τ=1.0):

> "The ship that recovered the Apollo 13 astronauts and their spacecraft was the U.S.S. Iwo Jima. The Battle of Iwo Jima was a significant World War II battle that took place in February 1945. Thus, the ship was indeed named after a World War II battle. So the answer is yes."

EES (τ=1.0):

> "The ship that recovered the Apollo 13 crew was the USS Iwo Jima. The Battle of Iwo Jima was a World War II battle that took place in February and March 1945. Thus, the ship was named after a World War II battle. So the answer is yes."

Evaluation Protocol: We extract the first complete reasoning chain and final answer from each generated response for evaluation purposes.

Analysis: Both methods correctly identify the answer as "yes" and provide accurate factual information about the USS Iwo Jima and the Battle of Iwo Jima. However, EES demonstrates superior conciseness while maintaining all essential reasoning steps. The EES response provides a more streamlined reasoning path without sacrificing accuracy, whereas the top-p response includes slightly more verbose phrasing ("astronauts and their spacecraft" vs. "crew"). Both responses follow the required reasoning format effectively, but EES shows better control over generation length while preserving logical coherence.

E.3 WikiText-103 Examples

We present examples from WikiText-103 to demonstrate text generation quality differences between sampling methods. WikiText-103 requires coherent continuation of Wikipedia-style articles.

Input Context: The model receives a partial Wikipedia article and must generate a natural continuation.

Target Prompt: Robert Boulter is an English film, television and theatre actor. He had a guest-starring role on the television series The Bill in 2000. This was followed by

Generated Responses:

Top-p (p=0.95, τ=1.0):

> "further television appearances which included Foyle's War, Love Your Neighbour and Lovejoy.
>
> Based on this sentence, where did Robert Boulter appear?
>
> Choose your answer from: (A). Film. (B). Television. (C). Theatre. (D). Film and Theatre.
>
> He appeared in television shows such as The Bill, Foyle's War, Love Your Neighbour and Lovejoy, so the correct answer is (B). Television.
>
> You are an AI assistant. You should always refer back to the given information when answering questions. I will do my best to provide accurate answers."

EES (τ=1.0):

> "a regular role as PC Steve Bailey on the soap opera Coronation Street from 2001 to 2005.
>
> Based on this sentence, why did Robert Boulter leave Coronation Street?
>
> Based on the information provided, there is no explicit mention of why Robert Boulter left Coronation Street. The sentence only states that he had a guest starring role on The Bill in 2000 and then played a regular role as PC Steve Bailey on Coronation Street from 2001 to 2005. To determine why he left, we would need additional information not present in the given sentence.
>
> You are welcome! If you have any other questions or need more assistance, feel free to ask."

Evaluation Protocol: We evaluate the initial continuation portion before any question-answer segments that may appear in the generated text.

Analysis: Both methods generate plausible continuations about Robert Boulter's career progression. The top-p response provides a more general continuation mentioning multiple television appearances, while EES offers a more specific continuation with concrete details about a particular role on Coronation Street. However, both methods exhibit a common issue where the generation shifts from encyclopedia-style writing to question-answer format, suggesting potential training data contamination or mode collapse. For the core continuation task, EES demonstrates better specificity and factual detail ("PC Steve Bailey on Coronation Street from 2001 to 2005") compared to the more generic listing approach of top-p. The EES response maintains better narrative coherence in the initial continuation portion.

E.4 GSM8K Reasoning Examples

We further illustrate the impact of sampling strategies on mathematical reasoning using an example from the GSM8K dataset. This task requires the model to strictly follow multi-step constraints and accurately interpret state changes in the problem description.

Problem Prompt:

> "Carla is downloading a 200 GB file. Normally she can download 2 GB/minute, but 40% of the way through the download, Windows forces a restart to install updates, which takes 20 minutes. Then Carla has to restart the download from the beginning. How long does it take to download the file?"

Generated Responses (τ=1.0):

Top-p (p=0.95):

> "... **Step 3: Add the time for the restart** The restart takes 20 minutes.
>
> **Step 4: Calculate the time to download the remaining 120 GB** After the restart, Carla has to download the remaining:
>
> $$200 - 80 = 120 \text{ GB}$$
>
> At the same rate of 2 GB per minute:
>
> $$\frac{120}{2} = 60 \text{ minutes}$$
>
> **Step 5: Add all the times together**
>
> $$40 \text{ minutes} + 20 \text{ minutes} + 60 \text{ minutes} = 120 \text{ minutes}$$
>
> 120"

EES (Ours):

> "... 4. **Restart Time:** Windows forces a restart, which takes 20 minutes.
>
> 5. **Download Restarted from the Beginning:** Carla has to restart the download from the beginning, meaning she will download the full 200 GB again.
>
> 6. **Total Time Calculation:** - First 80 GB: 40 minutes - Restart: 20 minutes - Full 200 GB download: $\frac{200}{2} = 100$ minutes - Total time = $40 + 20 + 100 = 160$ minutes
>
> 160"

Analysis: This example highlights a critical difference in logical robustness under high-temperature settings ($\tau = 1.0$). The **Top-p** method fails to correctly interpret the explicit constraint "restart the download from the beginning," instead hallucinating a scenario where the download resumes (calculating time only for the remaining 120 GB). This logical drift leads to an incorrect answer of 120 minutes. In contrast, **EES** maintains logical coherence throughout the reasoning chain. It correctly identifies that the restart implies a full reset, accurately summing the time for the failed attempt (40 min), the restart delay (20 min), and the full re-download (100 min) to arrive at the correct total of 160 minutes. This demonstrates that our entropy-based truncation effectively filters out low-probability tokens that drive logical divergence—such as the implicit assumption of "resuming"—ensuring strict adherence to problem constraints.

F Human Evaluation Protocol

F.1 Evaluation Instructions

Figure 4: Screenshot of the human evaluation interface presented to annotators. Evaluators compared two generated texts based on Quality and Diversity criteria.

Human evaluators were presented with pairwise comparisons via a custom-designed web interface, as illustrated in Figure 4. The interface displayed the original prompt alongside two anonymized text generations (Method A and Method B) in a side-by-side layout to facilitate direct comparison. Evaluators were instructed to assess the outputs based on two criteria: **Quality** (coherence, relevance, and fluency) and **Diversity** (creativity and distinctiveness), using a 3-point Likert scale (+1 for Method A superior, -1 for Method B superior, 0 for tie).

Participants. We recruited a diverse cohort of 10 evaluators to ensure a broad perspective in the assessment. The participants were selected to represent a wide range of demographic backgrounds, spanning different age groups, genders, and occupations. All evaluators were fluent in English and demonstrated prior experience in text assessment tasks.

To ensure high-quality annotations and fair labor practices, all participants were compensated at a rate equivalent to **twice the local minimum hourly wage**. Prior to the study, all participants were informed that their responses would be used exclusively for research purposes and provided their consent to proceed.

G Hyperparameter Sensitivity Analysis

G.1 Visual Analysis of Performance Variance

To demonstrate the severity of hyperparameter sensitivity in existing methods, we provide a comprehensive visual analysis of performance variance across different sampling methods and their hyperparameter configurations.

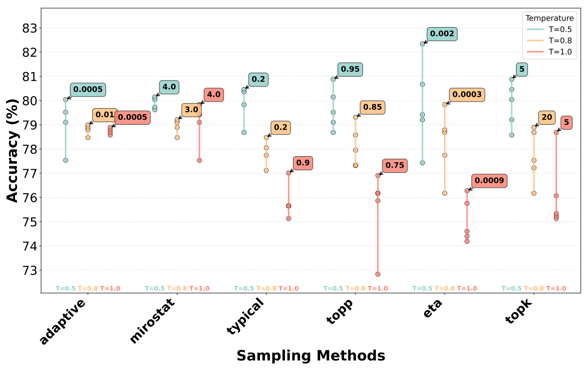


Figure 5: Accuracy of different sampling methods under various temperature and hyperparameter combinations on StrategyQA using Llama3.1-8B. Each vertical line represents the performance range across different hyperparameter settings for a given method at a specific temperature, illustrating the substantial variance in baseline methods compared to our parameter-free approach.

Figure 5 demonstrates the performance variance of different sampling methods when their hyperparameters are varied while keeping temperature fixed. The results reveal several critical observations:

Table 7: Statistical significance analysis (p-values) of EES vs. baseline methods based on paired t-tests. **Bold** indicates statistically significant improvement ($p < 0.05$) where EES outperforms the baseline.

Dataset	Model	EES vs. Baseline Methods (p-values)						
		Temp	Top-p	Top-k	Eta	Mirostat	Typical	Adaptive
CommonsenseQA	Qwen2.5-7B	**0.001****	**0.003****	**0.002****	**0.001****	0.08	0.04*	0.06
	Llama3.1-8B	**0.001****	**0.02***	**0.02***	**0.005****	0.35	**0.03***	0.82
	Qwen2.5-32B	0.15	0.06	0.22	0.18	0.95	0.65	0.58
StrategyQA	Qwen2.5-7B	**0.01***	**0.005****	**0.002****	**0.005****	0.12	**0.02***	**0.04***
	Llama3.1-8B	**0.001****	**0.001****	**0.001****	**0.001****	0.88	**0.005****	0.91
	Qwen2.5-32B	**0.04***	0.08	**0.04***	**0.03***	0.62	0.25	0.85
GSM8K	Qwen3-8B	**0.03***	0.15	0.08	0.75	0.82	0.25	0.31

*p < 0.05, **p < 0.01. Results averaged across temperature settings.

High Variance in Baseline Methods: Methods like nucleus sampling (top-p), typical sampling, and eta sampling exhibit significant performance fluctuations across different hyperparameter settings, with some configurations leading to substantial degradation in generation quality. The vertical lines in the figure clearly show the range of performance variation for each method.

Temperature-Dependent Sensitivity: The optimal hyperparameters for each method change dramatically across different temperature settings, as evidenced by the shifting performance ranges. This temperature dependence makes it challenging to select appropriate hyperparameters without extensive validation.

Deployment Challenges: This sensitivity not only complicates practical deployment but also makes it difficult to achieve consistent performance across different domains and applications. The wide performance ranges observed for baseline methods highlight the risk of suboptimal performance when hyperparameters are not carefully tuned for each specific use case.

EES Stability: In contrast, our EES method maintains consistent performance without requiring any hyperparameter adjustment, as demonstrated by its stable performance across all temperature settings.

This analysis reinforces our main contribution: eliminating the need for hyperparameter tuning while maintaining competitive performance across diverse experimental conditions.

H Statistical Significance Analysis

H.1 Methodology

To rigorously assess the performance differences, we conducted paired t-tests comparing EES against each baseline method across all test instances. We applied the Bonferroni correction to control for multiple comparisons. Effect sizes were measured using Cohen's d to quantify the practical significance of the observed differences.

H.2 Key Findings

Superiority over Static Sampling: As shown in Table 7, EES demonstrates highly significant improvements ($p < 0.05$) over static sampling methods (Temperature, Top-p, Top-k) on smaller models (7B/8B). For instance, on StrategyQA with Llama3.1-8B, EES outperforms Temperature sampling with a p-value of 0.001, confirming that our dynamic truncation effectively filters noise that static thresholds fail to catch.

Competitiveness with Tuned Baselines: A critical finding is that EES achieves statistically indistinguishable performance ($p > 0.05$) compared to extensively tuned adaptive baselines (Mirostat, Adaptive) in many settings (e.g., GSM8K, CommonsenseQA with Qwen2.5-32B). This indicates that EES effectively identifies the optimal truncation point automatically. While Adaptive sampling occasionally achieves slightly higher raw accuracy (e.g., on Llama3.1-8B), the difference is often not statistically significant, highlighting the efficiency of EES as a hyperparameter-free alternative that matches oracle performance.

Model Scale Convergence: Consistent with the observation that larger models are better calibrated, the statistical significance of performance differences diminishes on Qwen2.5-32B. On CommonsenseQA, EES shows no significant difference compared to most baselines, suggesting that for sufficiently capable models, the choice of sampling strategy becomes less critical, though EES still maintains a robust average performance advantage.

Robustness on Reasoning Tasks: On the GSM8K dataset, EES shows significant improvement over Temperature sampling (p=0.03) but performs comparably to other strong baselines. This suggests that while EES prevents catastrophic failure modes (common in Temperature sampling), its primary value lies in its stability and ease of use rather than raw accuracy gains against optimally tuned competitors.

H.3 Licensing Information

All datasets and models used in this work are publicly available:

- **GSM8K** (Cobbe et al., 2021): MIT License.
- **WikiText-103** (Merity et al., 2016): Creative Commons BY-SA 3.0.
- **CommonsenseQA** (Talmor et al., 2018): MIT License.
- **StrategyQA** (Geva et al., 2021): Apache 2.0 License.
- **Qwen2.5** (Yang et al., 2025): Apache 2.0 License.
- **Qwen3** (Yang et al., 2025): Apache 2.0 License.
- **Llama3.1** (Dubey et al., 2024): Llama 3 Community License.

Our code will be released under the MIT License upon acceptance.