Top-H Decoding: Adapting the Creativity and Coherence with Bounded Entropy in Text Generation

Erfan Baghaei Potraghloo[u]†, **Seyedarmin Azizi**[u]†, **Souvik Kundu**[i], **and Massoud Pedram**[u]

[u]University of Southern California, Los Angeles, USA
[i]Intel AI, USA
† Equal contribution authors
{baghaeip, seyedarm, pedram}@usc.edu, souvikk.kundu@intel.com

Abstract

Large language models (LLMs), despite their impressive performance across a wide range of tasks, often struggle to balance two competing objectives in open-ended text generation: fostering diversity and creativity while preserving logical coherence. Existing truncated sampling techniques, including temperature scaling, top-p (nucleus) sampling, and min-p sampling, aim to manage this trade-off. However, they exhibit limitations, particularly in the effective incorporation of the confidence of the model into the corresponding sampling strategy. For example, min-p sampling relies on a single top token as a heuristic for confidence, eventually underutilizing the information of the probability distribution. To effectively incorporate the model confidence, this paper presents ***top-H*** decoding. We first establish the theoretical foundation of the interplay between creativity and coherence in truncated sampling by formulating an **entropy-constrained minimum divergence** problem. We then prove this minimization problem to be equivalent to an **entropy-constrained mass maximization** (ECMM) problem, which is NP-hard. Finally, we present top-H decoding, a computationally efficient greedy algorithm to solve the ECMM problem. Extensive empirical evaluations demonstrate that top-H outperforms the state-of-the-art (SoTA) alternative of min-p sampling by up to $\mathbf{25.63}\%$ on creative writing benchmarks, while maintaining robustness on question-answering datasets such as GPQA, GSM8K, and MT-Bench. Additionally, an *LLM-as-judge* evaluation confirms that top-H indeed produces coherent outputs even at higher temperatures, where creativity is especially critical. In summary, top-H advances SoTA in open-ended text generation and can be *easily integrated* into creative writing applications. The code is available at https://github.com/ErfanBaghaei/Top-H-Decoding.

1 Introduction

Large language models (LLMs) have exhibited impressive abilities in open-ended generation tasks, including creative writing and multi-turn dialogue (Lee et al., 2022). However, these models often need to deal with the challenge of *balancing creativity and coherence*, accepting less likely and more imaginative token choices while avoiding incoherent or nonsensical output. This trade-off is complex, as indiscriminate broadening of the sampling pool can lead to fragmented or disjoint text (Holtzman et al., 2019).

To navigate this balance, various sampling strategies have emerged, including temperature scaling (Ackley et al., 1985), top-k (Fan et al., 2018), top-p (nucleus) (Holtzman et al., 2019), η (Hewitt et al., 2022), and min-p sampling (Nguyen et al., 2024). They generally apply heuristics to control diversity and risk. Specifically, min-p sampling (Nguyen et al., 2024) stands out for its dynamic

39th Conference on Neural Information Processing Systems (NeurIPS 2025).

truncation of low-probability tokens using a threshold tied to the probability of the top token. Although this method performs well at high temperatures (T), its exclusive reliance on the maximum probability token to estimate confidence disregards the potential distribution of the probability mass over the remaining vocabulary. As a result, min-p remains vulnerable to *over-truncation* in sparse (low-entropy) distributions and *under-truncation* in dense (high-entropy) distributions.

The above limitation motivates the need for a more methodical *confidence-aware* sampling framework that accounts for the overall shape of the distribution, rather than only its peak. In addition, the proliferation of heuristic methods highlights a deeper issue, namely *the lack of a theory-based foundation to analyze the interplay between creativity and coherence in autoregressive generation*.

Our Contributions. Towards effective incorporation of the confidence of the model, in this work, we present **top-H** decoding. In particular, top-H maintains the creativity and coherence balance guided by bounded entropy in text generation. Unlike most earlier approaches that rely on a fixed threshold, top-H dynamically selects a subset of tokens such that the resulting truncated distribution over the selected subset has an upper-bounded uncertainty while maintaining minimal divergence from the original distribution predicted by the model.

To formally ground top-H, we first introduce a constrained optimization problem that characterizes the trade-off between creativity and coherence in language generation, namely, ***entropy-constrained minimum divergence*** (ECMD). We show that this minimization is equivalent to an ***entropy-constrained mass maximization*** (ECMM) problem. We then prove that ECMM is NP-hard. Thus, in top-H, we offer a greedy solution that is both efficient and practically effective in approximating the solution of the ECMM while bounding the entropy of the selected distribution. During autoregressive generation, as the token distribution evolves at each step, top-H adjusts its entropy threshold based on the entropy of the token distribution, thereby ***dynamically*** adapting to the model's varying confidence over time.

We validate the effectiveness of top-H through extensive experiments in a diverse set of tasks, including creative writing (Alpaca-Eval (Li et al., 2023) and MT-Bench (Zheng et al., 2023)), reasoning (GSM8k (Cobbe et al., 2021) and GPQA (Rein et al., 2024)), and human-aligned evaluations using LLM as a judge framework. Specifically, top-H consistently outperforms existing sampling methods in accuracy while maintaining a robust balance between expressiveness and fluency. For example, compared to min-p, top-H demonstrates an accuracy improvement of up to $\mathbf{25.63}\%$.

2 Related Work

2.1 Stochastic Sampling Strategies for Autoregressive Models

Temperature scaling (Ackley et al., 1985) multiplies the logits by a scalar, encouraging the exploration of less likely tokens. However, it can get too indiscriminate at high Ts, generating incoherent or contradictory texts. Top-k (Fan et al., 2018) includes only the k highest probability tokens. Although simple, this *hard cutoff* is insensitive to context, sometimes excluding large swaths of moderately plausible tokens. Top-p (nucleus) sampling (Holtzman et al., 2019) chooses the smallest subset of tokens whose cumulative probability exceeds p. This alleviates some of the rigidity of top-k. Unfortunately, at high T, the distribution can be so flat that the top-p may inadvertently include very low-probability tokens, harming coherence. This incoherence in top-p sampling is demonstrated in the experimental results Table 4, where the coherence score on text drops significantly at higher T.

Min-p (Nguyen et al., 2024) sampling dynamically scales a base probability score threshold p_{base} by the probability of the top-1 token. This effectively restricts the sample space more aggressively when the model is confident. Min-p has been shown to outperform top-p in tasks requiring both diversity and correctness at higher temperatures. However, *its reliance on only the highest-probability token* can overlook broader features of the distribution. Two different probability mass functions might share a top-1 token probability; however, they differ widely in their overall confidence.

2.2 Entropy-Based Sampling Strategies for Autoregressive Models

Several methods attempt to exploit *entropy* or related uncertainty measures when sampling. η-sampling (Hewitt et al., 2022) dynamically adjusts the sampling threshold based on the entropy of the distribution of the next token. However, this method often requires carefully tuned hyperparameters and can introduce significant runtime overhead at higher Ts. Mirostat (Basu et al., 2020) aims to

maintain a target perplexity (related to entropy) via feedback control. Although it can yield steady perplexities, it adds complexity to parameter tuning and integration into generation pipelines.

Despite their *entropy-aware* intentions, these approaches do not strictly limit the randomness of the sampling distribution; instead, they often aim to achieve a perplexity target or modify the sampling heuristics in real-time. As a result, controlling the maximum allowed randomness in the final distribution, thus ensuring both coherence and flexibility, can be challenging.

3 Motivational Case Study

This section presents a key motivation to develop a new sampling method, despite the widespread use of nucleus and min-p sampling within the community. Specifically, we try to pose the following question.

Why do we need a more ***distribution-aware*** *sampling technique if min-p already considers the model's confidence?*

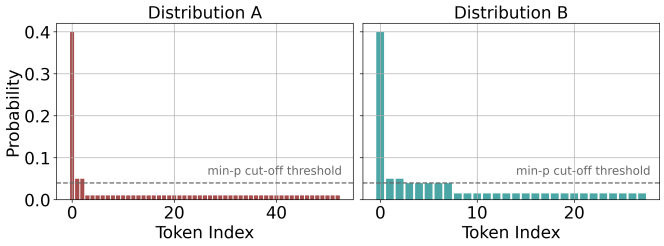

Figure 1: Probability distribution of two different types with associated min-p threshold.

Min-p employs a dynamic truncation threshold by modulating the maximum probability of the next token probability distribution with a base factor. Although this approach accounts for the confidence of the model to some extent, it is insufficient to select an optimal sampling pool.

Consider one scenario where min-p may yield low efficacy as illustrated in Fig. 1. Distributions A and B represent two token probability distributions over the vocabulary, where tokens are sorted by their probability values, and tokens not shown have a probability of zero. Since both distributions have the same maximum probability, min-p applies a similar cut-off threshold. However, the two distributions are distinct in terms of confidence. Distribution A exhibits greater randomness, as it contains numerous low-probability tokens, while distribution B includes some high-probability tokens discarded due to min-p's truncation threshold. This example demonstrates that the min-p approach does not accurately capture the underlying distributional characteristics. Consequently, we are motivated to adopt a sampling method that considers the overall shape of the probability distribution rather than solely relying on a maximum probability threshold. In Appendix C.4, we demonstrate how our proposed sampling strategy, top-H, addresses this issue using the exact same example.

4 Theoretical Foundation for Entropy-Based Truncation Sampling

This section establishes the theoretical foundations of top-H sampling. Given a language model $\mathcal{M}$ and a preceding context window $x_{1:t-1}$, the probability distribution over the vocabulary $\mathcal{V}$ for the next token x_t can be written as,

$$p(x_t) = \mathcal{M}(x_{1:t-1}). \tag{1}$$

Our objective is to determine a subset $S \subseteq \mathcal{V}$ from which the next token will be sampled, ensuring that the resulting probability distribution over the subset S, denoted $q(x_t) : S \to [0, 1]$ with $\sum_{x_t \in S} q(x_t) = 1$, satisfies the following desired characteristics:

1. **Minimum divergence from the original probability mass function:** The subset S should be constructed such that the distribution over the tokens in the subset, $q(x_t)$, has minimal divergence from the original distribution $p(x_t)$, thereby "maximally matches" $p(x_t)$.

2. **Reduced randomness for enhanced coherence**: The probability mass function $q(x_t)$ should exhibit lower randomness compared to $p(x_t)$ in the sense that $H(q) \leq H(p)$ where $H(.)$ denotes the Shannon entropy, effectively upper-bounding uncertainty.

These criteria form the basis of top-H, which seeks to construct S and calculate q so that q maximally matches p while exhibiting lower randomness compared to it[1]. To regulate diversity in a controllable manner, we introduce a parametric randomness bound, parameterized by α (see Eq. 2). We formalize this objective as a minimization of the Jensen–Shannon divergence (JSD) between p and q under the parametric entropy constraint. Formally, we intend to solve the following.

$$\min_{S} \text{JSD}(q \,\|\, p) \quad \text{subject to} \quad H(q) \;\leq\; \alpha\, H(p), \tag{2}$$

where $\alpha \in (0, 1)$ is a tunable hyperparameter. We refer to this problem as ***entropy-constrained minimum divergence*** (ECMD). By upper-bounding $H(q)$ in proportion to $H(p)$, ECMD encourages the sampling of *more* tokens in the case of higher uncertainty (higher $H(p)$) and the *less* token in the case of lower uncertainty (lower $H(p)$). This approach preserves coherent tokens in contexts where the model "knows" likely the next token, yet encourages exploration when multiple candidates plausibly fit the context, precisely where creativity is more beneficial. Therefore, with an appropriate choice of α, solving the ECMD problem can *ideally balance creativity and coherence* in autoregressive text generation. In the rest of this section, we prove the following statements. I) Minimizing JSD under an entropy bound is equivalent to maximizing the sum of probabilities of the tokens in S (subject to $H(q) \leq \alpha H(p)$). II) The ECMD problem is, in general, NP-hard.

4.1 Formulation of the JSD Minimization Problem

We first start by defining the values of each element in the probability distribution of p and q, respectively. Assuming v_i denotes the i^{th} token in the dictionary $\mathcal{V}$, the conditional probability p_i of selecting v_i as the t^{th} generated token given $x_{1:t-1}$ is,

$$p_i = \texttt{Prob}(x_t = v_i | x_{1:t-1}) \quad \text{for} \quad i = 1, 2. \ldots, n$$

where $n = |\mathcal{V}|$, $|.|$ identifies the cardinality of a set. Similarly, the conditional probability q_i of selecting v_i as the t^{th} generated token given $x_{1:t-1}$ is

$$q_i = \begin{cases} \frac{p_i}{\Gamma_S} & v_i \in S \\ 0 & \text{otherwise.} \end{cases}, \quad \text{where} \quad \Gamma_S = \sum_i p_i \mathbb{1}_{\{v_i \in S\}} \tag{3}$$

Having defined the distributions, the Jensen-Shannon divergence between p and q is calculated as

$$\text{JSD}(p||q) = \frac{1}{2} D_{KL}(p||M) + \frac{1}{2} D_{KL}(q||M), \quad \text{where} \quad M = \frac{1}{2}(p + q) \tag{4}$$

Next, without loss of generality, we use the properties of JSD and re-formulate the ECMD problem as a maximization problem of the probability mass function for ease of analysis.

4.2 Equivalence to Entropy-Constrained Mass Maximization

The ECMD problem in Equation 2 is challenging to analyze directly due to the complexity associated with the expansion of JSD. We thus reformulate the problem using the Γ_S metric to facilitate analysis and interpretation. The following theorem formalizes the necessary condition for achieving the optimal solution to the original optimization problem in terms of Γ_S.

Theorem 1. *The Jensen-Shannon divergence between the distributions p and q is only dependent on the Γ_S and can be minimized by maximizing Γ_S.*

Proof. Refer to Appendix A.1 for the proof. □

As a result, ECMD can be rewritten as the following,

$$\max_{S} \Gamma_S \quad s.t. \quad H(q) \leq \alpha H(p) \rightarrow \max_{S} \sum_i p_i \mathbb{1}_{\{v_i \in S\}} \quad s.t. \quad H(q) \leq \alpha H(p) \tag{5}$$

[1]From now on, we will use p and q to refer the distributions of the next token defined over the original set ($\mathcal{V}$) and the selected subset (S) of tokens.

We name the above formulation as ***entropy-constrained mass maximization*** (ECMM). This reformulated version of the problem is easier to reason about. Next, we prove that given $0 < \alpha < 1$, the problem remains NP-hard. Finally, we propose a greedy approach as a solution to this. Unless otherwise specified, we empirically set $\alpha = 0.4$ and use it throughout our analysis.

4.3 NP-Hardness Proof of the ECMM Problem

Theorem 2. *The entropy-constrained mass maximization problem is NP-hard.*

Proof. In Appendix A.2, we present a detailed polynomial-time reduction from the well-known cardinality-constrained subset-sum (CCSS) problem (Garey & Johnson, 1979). As CCSS is a popular NP-complete problem, our formulation establishes the NP-hardness of ECMM. □

5 Top-H Decoding Method

Having established the NP-hardness of the ECMM problem, we recognize that it cannot be solved efficiently in the general cases. Thus, to produce a practical, efficient, and yet competitive solution, we now present a greedy approximation algorithm, namely **top-H**. Top-H incrementally maximizes the objective of Eq. 5, while adhering to the imposed entropy constraint.

Algorithm 1 Top-H: proposed greedy token selection algorithm

Require: Probability mass function $p = (p_1, p_2, \ldots, p_n)$, entropy threshold coefficient $\alpha \in (0, 1)$
Ensure: Selected token set S

```
Sort tokens in descending order of probability: p_1 ≥ p_2 ≥ ... ≥ p_n
Initialize S ← ∅, H(q) ← 0
for each token i in sorted order do
    Add token i to S
    Compute updated distribution q over S
    Compute entropy H(q)
    if H(q) > α · H(p) then
        Remove token i from S
        break
    end if
end for
return S
```

Algorithm 1 outlines the token selection strategy of top-H. The objective is to maximize the probability mass of the tokens $\sum_i p_i \mathbb{1}_{\{v_i \in S\}}$, where the tokens v_i are selected into the sampling set S. To achieve this, the algorithm begins by sorting all candidate tokens in the *descending* order of their probabilities. It then iteratively adds tokens to the sampling set in this order. After each addition, a distribution q is constructed over the selected tokens, and its entropy is calculated. Top-H continues this process until the entropy of q reaches the dynamic[2] threshold $\alpha \cdot H(p)$, ensuring that the selected subset respects the global entropy constraint.

Unlike prior truncation-based sampling methods, top-H explicitly controls the randomness of the distribution it samples from, $H(q)$, by adapting it to the entropy of the original next token probability distribution $H(p)$. As a result, the allowed randomness dynamically adjusts throughout the steps of autoregressive generation as p evolves. In Section 6.2, we provide empirical evidence on the competitiveness of the top-H's greedy approach in solving the ECMM.

We now present a theorem that guarantees the termination of the algorithm, with an *early* convergence governed by the entropy scaling coefficient α.

Termination Guarantee. Entropy is a *non-linear* and *non-monotonic* function. Thus, the entropy of the distribution q over a set S is not predictable as tokens are added. Specifically, adding a token

[2]At each step of auto-regressive token generation, the model produces a new probability distribution p, causing the entropy threshold $\alpha H(p)$ to vary dynamically across generation steps.

to S can increase or decrease the entropy, depending on the underlying probabilities. However, under a greedy selection strategy, it can be shown that each additional token strictly increases the entropy of q. Consequently, the entropy constraint is not a vacuous bound, and the growth of S is inherently bounded; the set cannot expand indefinitely without eventually violating the entropy constraint. This intuition is formalized in the following theorem.

Theorem 3. *Consider a greedy algorithm that selects tokens in descending order of their probabilities. Let q be the probability mass function over the selected tokens. Then, the entropy of q increases strictly at each selection step and is maximized only when all tokens are selected. Therefore, if the entropy threshold coefficient α is chosen such that $0 < \alpha < 1$, the algorithm is guaranteed to terminate before all tokens are selected.*

Proof. Refer to Appendix A.3 for the proof. □

The termination guarantee uses the monotonic growth of entropy under the greedy selection procedure. Each token added to the set contributes positively to the entropy, regardless of its probability, thus ensuring that the entropy $H(q)$ approaches the threshold $\alpha\, H(p)$. The algorithm stops adding tokens to the set S at the moment when any further addition of tokens would violate the constraint. This ensures that the ECMM objective avoids the trivial solution of selecting all tokens while still satisfying the entropy constraint.

6 Experiments

6.1 Experimental Setup

Models, sampling methods, and datasets. We evaluate top-H on three recent instruction-tuned language models, namely, LLaMA3.1–8B–Instruct (Grattafiori et al., 2024), Qwen2.5–3B (Yang et al., 2024), and Phi-3-Mini-4K-Instruct (Abdin et al., 2024). As baselines, we compare with several widely used truncation-based sampling methods, namely, top-k, top-p (nucleus sampling), min-p, and η-sampling. Our evaluations span multiple benchmarks designed to test creative generation, reasoning ability, and evaluative judgment. Specifically, we also used the Alpaca-Eval dataset (Li et al., 2023), GSM8K (Cobbe et al., 2021), GPQA (Rein et al., 2024), MT-Bench (Zheng et al., 2023), and an LLM-as-a-judge evaluation setting.

Experimental settings. For decoding hyperparameters, we follow the configuration in (Nguyen et al., 2024), using `min_p` = 0.1, `top_p` = 0.9, and η = 0.0002 for min-p, top-p, and η-sampling methods, respectively. We choose the best result out of the `k` = 10, 20, and 50 for top-k method. Regarding the evaluation, we use the `lm-eval-harness` framework (EleutherAI, 2023) and report exact match accuracy with the flexible extract filter on the GPQA and GSM8K datasets, length-controlled win rate on Alpaca-Eval, and judge scores (on a scale from 1 to 10) on MT-Bench. For Alpaca-Eval and MT-Bench, we used GPT-4o (OpenAI, 2024) as the judge LLM. All experiments were conducted on a single NVIDIA A6000 GPU, and algorithms were implemented using PyTorch version `2.5.1+cu124` and the Hugging Face Transformers library version `4.50.1`.

6.1.1 Performance on Creative Writing: Alpaca-Eval and MT-Bench

Fig. 2 presents compelling evidence for the superiority of top-H sampling compared to alternative SoTA approaches. In Fig. 2(a-c) (Alpaca-Eval), top-H shows remarkable improvements over the state-of-the-art min-p method, and also conventional sampling methods. For example, for LLaMA3.1-8B across different T, top-H demonstrates an win-rate (%) improvement of up to $\mathbf{17.11}\%$ compared to SoTA min-p sampling. A critical finding from Fig. 2 is the resilience of top-H to temperature scaling. **While traditional sampling methods exhibit severe performance degradation at higher T, top-H preserves much of its effectiveness**. For instance, for LLaMA3.1-8B-Instruct in Fig. 2(a), top-p sampling shows a catastrophic 34.06% decline in win rate from T=1 to T=2. In contrast, top-H experiences only a 3.78% reduction over the same temperature range. This robustness is particularly significant given that higher T settings are essential for generating diverse, creative texts. The MT-Bench results (Fig. 2(d-f)) further validate the capability of top-H. For example, for LLaMA3.1-8B, similar to that on Alpaca-Eval, the advantage becomes more pronounced at higher T, with top-H achieving a higher score value of up to $\mathbf{3.78}$.

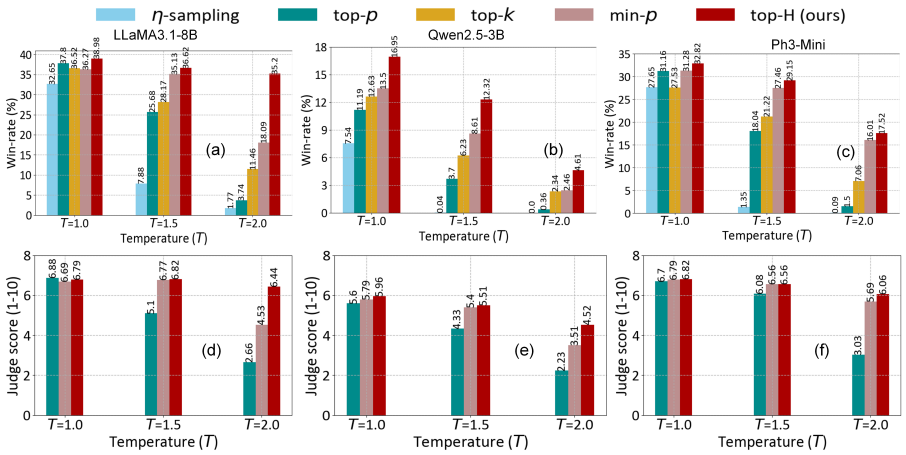

Figure 2: (a)-(c): Length-controlled win rates (%) comparison of different SoTA sampling with top-H on Alpaca-Eval benchmark. (d)-(f): Judge scores (on a scale of 1 to 10) on MT-Bench.

6.1.2 Performance on Reasoning and CoT Tasks

Following the setup in (Nguyen et al., 2024) we use the `gsm_cot` (8-shot) and `gpqa_main_generative_n_shot` (8-shot) tasks for GSM8k and GPQA, respectively.

The experimental results in Tables 1 and 2 demonstrate the effectiveness of top-H compared to min-p and top-p sampling across language models and temperature settings. Temperature is a key hyperparameter in generation, striking a balance between creativity and factual accuracy.

On the GSM8k benchmark (Table 1), top-H consistently outperforms the alternatives. At T=1, it leads for all the models, outperforming both min-p and top-p by a significant margin. Top-H maintains strong accuracy as temperature increases, while the baselines degrade significantly. At T=2, the contrast becomes even more pronounced, with top-p showing near-total collapse (declining by up to 73.62% (Phi-3-Mini) in accuracy), while top-H experiences a very modest degradation, showing an accuracy improvement of up to $\mathbf{25.63}\%$ compared to min-p (on LLaMA3.1-8B).

A similar trend is observed in GPQA benchmark (Table 2). At T=1, top-H remains competitive, outperforming both top-p and min-p on Qwen2.5 and Phi-3-Mini. At T=2, it exhibits notable robustness, maintaining performance levels substantially higher than those of top-p, which experiences significant deterioration. Compared to min-p, top-H an accuracy improvement of up to 3.12%, 2.67%, and 7.36% on Qwen2.5, LLaMA3.1, and Phi-3-Mini, respectively. In summary, **top-H demonstrates competitive performance even at low temperatures and significantly superior performance at higher temperatures, marking it as a reliable sampling strategy for diverse generation needs**.

Temperature	Qwen2.5 3B			LLaMA3.1-8B-Instruct			Phi-3-Mini		
	Min-p	Top-p	Top-H	Min-p	Top-p	Top-H	Min-p	Top-p	Top-H
1.0	72.40	71.27	**75.97**	48.90	67.93	**76.35**	81.96	81.35	**83.24**
1.5	66.79	55.57	**72.55**	58.00	23.81	**70.51**	77.10	67.25	**77.86**
2.0	49.43	9.10	**55.57**	13.72	2.65	**39.35**	**60.88**	7.73	60.20

Table 1: Accuracy (%) for top-H, min-p, and top-p on GSM8K.

Temperature	Qwen2.5 3B			LLaMA3.1-8B-Instruct			Phi-3-Mini		
	Min-p	Top-p	Top-H	Min-p	Top-p	Top-H	Min-p	Top-p	Top-H
1.0	28.35	27.68	**28.79**	26.34	**32.81**	29.24	31.92	30.58	**32.37**
1.5	**30.13**	27.23	27.90	28.35	28.57	**30.58**	29.91	28.57	**30.80**
2.0	25.00	22.32	**28.12**	26.12	23.88	**28.79**	23.44	18.53	**30.80**

Table 2: Accuracy (%) for top-H, min-p, and top-p on GPQA.

6.1.3 Performance Analysis with LLM-as-a-Judge

In this section, we employ the LLM-as-a-Judge framework to directly evaluate the creativity and coherence of texts generated using min-p, top-p, and top-H sampling strategies. Following the

evaluation setup proposed in (Nguyen et al., 2024), we use three open-ended prompts designed to elicit creative storytelling on diverse topics. We generate responses using three different models: LLaMA3.1-8B-Instruct, Qwen2.5-3B, and Phi3-Mini-4k-Instruct, each evaluated across three different temperature settings. The top-p and min-p sampling methods serve as baselines for comparison. The prompts used are closely aligned with those in (Nguyen et al., 2024) and are listed in Appendix B. We use GPT-4o (OpenAI, 2024) as the judge model to assess the outputs, which scores the responses based on creativity and coherence using the evaluation prompt detailed in Appendix B.

For each evaluation, the outputs of the three sampling strategies are randomly shuffled to mitigate positional bias. The scores are then extracted from the GPT-4o evaluation responses. To reduce the impact of randomness and noise, each experimental configuration, defined by model, temperature, prompt, and sampling strategy, is **repeated five times**, and the average score is reported. The results for LLaMA3.1-8B-Instruct are presented in Table 3. The results in Table 3 reveal a consistent trend: At lower temperatures, top-H produces outputs with significantly higher creativity, originality, and coherence compared to min-p and top-p sampling methods in all three prompts. As T increases, the top-p sampling suffers a marked decline in coherence, often generating fragmented and incoherent text. This degradation stems from top-p's lack of awareness of the model's confidence, as it truncates the distribution based purely on cumulative probability without accounting for distributional entropy.

In contrast, min-p and top-H maintain stronger coherence at higher temperatures by adaptively limiting their sampling pools based on model confidence. Among the two, top-H consistently outperforms min-p in both creativity and coherence. This is attributed to top-H's direct control over the entropy of the selected token set, allowing it to modulate randomness in alignment with the model's uncertainty. Additional LLM-as-a-Judge results supporting these conclusions are provided in Table 5 in the Appendix, which covers the evaluations on the Qwen2.5 and Phi-3-Mini models.

Temperature	Prompt	Sampling	M1	M2	M3	M4	M5	Average
1.0	Prompt 1	Top-p	7.45 $\pm$0.20	6.35 $\pm$0.22	**8.75** $\pm$0.26	6.60 $\pm$0.30	7.55 $\pm$0.15	7.45 $\pm$0.30
		Min-p	8.25 $\pm$0.22	7.60 $\pm$0.26	8.25 $\pm$0.15	7.65 $\pm$0.20	7.60 $\pm$0.15	7.85 $\pm$0.26
		Top-H	**8.80** $\pm$0.15	**8.65** $\pm$0.20	8.40 $\pm$0.22	**8.05** $\pm$0.15	**8.55** $\pm$0.22	**8.45** $\pm$0.26
	Prompt 2	Top-p	7.85 $\pm$0.15	7.20 $\pm$0.15	**8.35** $\pm$0.22	7.05 $\pm$0.35	7.50 $\pm$0.15	7.60 $\pm$0.30
		Min-p	7.25 $\pm$0.24	7.20 $\pm$0.15	8.05 $\pm$0.26	7.55 $\pm$0.20	6.75 $\pm$0.35	7.40 $\pm$0.22
		Top-H	**8.10** $\pm$0.2	**8.10** $\pm$0.3	8.25 $\pm$0.15	**7.90** $\pm$0.15	**8.35** $\pm$0.26	**8.25** $\pm$0.30
	Prompt 3	Top-p	6.80 $\pm$0.26	6.10 $\pm$0.22	**8.90** $\pm$0.22	7.05 $\pm$0.20	7.65 $\pm$0.35	7.20 $\pm$0.15
		Min-p	6.7 $\pm$0.26	6.65 $\pm$0.25	7.65 $\pm$0.26	6.77 $\pm$0.25	7.00 $\pm$0.20	6.90 $\pm$0.30
		Top-H	**8.15** $\pm$0.26	**8.05** $\pm$0.26	8.00 $\pm$0.15	**8.30** $\pm$0.31	**8.25** $\pm$0.20	**8.05** $\pm$0.30
1.5	Prompt 1	Top-p	7.45 $\pm$0.15	7.10 $\pm$0.26	8.20 $\pm$0.22	7.55 $\pm$0.30	7.35 $\pm$0.22	7.55 $\pm$0.31
		Min-p	7.95 $\pm$0.22	7.55 $\pm$0.35	8.25 $\pm$0.20	7.55 $\pm$0.26	7.60 $\pm$0.22	7.80 $\pm$0.26
		Top-H	**8.75** $\pm$0.22	**9.05** $\pm$0.26	**8.50** $\pm$0.15	**8.40** $\pm$0.15	**8.80** $\pm$0.20	**8.80** $\pm$0.22
	Prompt 2	Top-p	7.80 $\pm$0.30	7.75 $\pm$0.22	**8.65** $\pm$0.35	7.00 $\pm$0.22	7.65 $\pm$0.26	7.75 $\pm$0.22
		Min-p	7.30 $\pm$0.20	7.10 $\pm$0.31	7.87 $\pm$0.30	6.80 $\pm$0.26	6.75 $\pm$0.26	7.10 $\pm$0.15
		Top-H	**8.10** $\pm$0.20	**8.10** $\pm$0.26	8.05 $\pm$0.15	**7.70** $\pm$0.20	**8.05** $\pm$0.22	**8.10** $\pm$0.22
	Prompt 3	Top-p	7.40 $\pm$0.20	6.85 $\pm$0.26	7.70 $\pm$0.15	7.20 $\pm$0.22	8.35 $\pm$0.31	7.45 $\pm$0.30
		Min-p	6.35 $\pm$0.20	6.05 $\pm$0.22	**7.85** $\pm$0.22	6.55 $\pm$0.15	7.20 $\pm$0.26	6.80 $\pm$0.26
		Top-H	**8.35** $\pm$0.30	**7.80** $\pm$0.31	7.80 $\pm$0.20	**8.05** $\pm$0.22	**8.10** $\pm$0.30	**8.05** $\pm$0.26
2.0	Prompt 1	Top-p	7.00 $\pm$0.26	6.45 $\pm$0.30	5.35 $\pm$0.26	5.40 $\pm$0.26	5.60 $\pm$0.24	5.95 $\pm$0.22
		Min-p	8.05 $\pm$0.31	8.35 $\pm$0.31	7.65 $\pm$0.24	7.15 $\pm$0.22	7.65 $\pm$0.31	7.70 $\pm$0.20
		Top-H	**8.80** $\pm$0.22	**9.05** $\pm$0.24	**8.75** $\pm$0.26	**8.70** $\pm$0.15	**8.80** $\pm$0.22	**8.85** $\pm$0.20
	Prompt 2	Top-p	8.25 $\pm$0.20	7.65 $\pm$0.30	3.85 $\pm$0.20	5.20 $\pm$0.24	6.30 $\pm$0.22	6.25 $\pm$0.15
		Min-p	7.60 $\pm$0.15	7.45 $\pm$0.30	7.15 $\pm$0.31	7.55 $\pm$0.20	7.40 $\pm$0.35	7.40 $\pm$0.26
		Top-H	**8.85** $\pm$0.20	**8.35** $\pm$0.31	**8.60** $\pm$0.26	**8.60** $\pm$0.30	**8.70** $\pm$0.22	**8.60** $\pm$0.22
	Prompt 3	Top-p	7.15 $\pm$0.25	**8.05** $\pm$0.24	4.20 $\pm$0.24	5.65 $\pm$0.20	**7.80** $\pm$0.31	6.55 $\pm$0.30
		Min-p	6.80 $\pm$0.31	6.75 $\pm$0.31	7.20 $\pm$0.30	6.30 $\pm$0.15	6.35 $\pm$0.20	6.65 $\pm$0.20
		Top-H	**8.0** $\pm$0.31	7.1 $\pm$0.22	**9.0** $\pm$0.15	**8.05** $\pm$0.20	7.05 $\pm$0.20	**7.65** $\pm$0.24

Table 3: Evaluation metrics and the judge scores (on a scale of 1.0 to 10.0) for different temperatures, prompts, and sampling methods on **LLaMA3.1-8B-Instruct**. M1-M5 denote creativity, originality, narrative flow, imagery, and vitality, respectively.

In Appendix C, we present additional results and discussions on top-H, including **evaluations with a larger 70B model**, **human evaluation** of creativity and coherence across different sampling techniques, and **comparison of top-H to Mirostat method**.

6.1.4 Computational Overhead and Timing Comparisons

We compare per-token decode latency (ms/token) of top-H against top-p and min-p on three models: **LLaMA3.1-8B-Instruct**, **Phi-3-Mini-3.8B**, and **LLaMA3.3-70B-Instruct** in the Table 4. For each

configuration, we evaluate on 100 prompts from AlpacaEval, generating 128 tokens per prompt, and report the mean ms/token over prompts. Specifically, we observe a negligible overhead of **as low as 0.8%** compared to min-p and top-p.

Computational complexity. Let n denote the vocabulary size, and let $p_1 \geq p_2 \geq \cdots \geq p_n$ be the sorted probabilities. Sorting the logits dominates the computational cost for all cumulative decoding methods, requiring $O(n \log n)$ time. Subsequent operations such as partial selection or cumulative thresholding in top-p and min-p decoding only involve a single linear pass, adding $O(n)$ additional work but not changing the overall asymptotic complexity.

For top-H decoding, define the *partial entropy* $h_j = \sum_{i=1}^{j} p_i \log p_i$. According to the proof of Theorem A.3, the entropy of the distribution q^j is given by

$$H(q^j) = \log \Gamma_j - \frac{h_j}{\Gamma_j},$$

where the cumulative mass satisfies $\Gamma_j = \Gamma_{j-1} + p_j$, and the partial entropy follows $h_j = h_{j-1} + p_j \log p_j$. These recurrences enable *incremental entropy accumulation* (Alg. 2), which updates $H(q^j)$ in $O(1)$ time per step, or $O(n)$ in total given sorted inputs. Therefore, the overall complexity of top-H decoding is also bounded by the sorting step, i.e., $O(n \log n)$. In practice, $\log p_j$ values are directly available from the model's log-probabilities.

Algorithm 2 Incremental entropy accumulation

Initialize $\Gamma \leftarrow 0$, $h \leftarrow 0$, $H \leftarrow 0$
for each step j **do**
  $\Gamma \leftarrow \Gamma + p_j$
  $h \leftarrow h + p_j \log p_j$
  $H \leftarrow \log(\Gamma) - \frac{h}{\Gamma}$
end for

Temperature	LLaMA3.1-8B-Instruct			Phi-3-Mini			LLaMA3.3-70B-Instruct		
	Top-H	Min-p	Top-p	Top-H	Min-p	Top-p	Top-H	Min-p	Top-p
1.0	28.3951	27.3396	27.4275	24.3847	23.6499	23.7809	219.3837	219.1391	218.4900
2.0	28.4671	27.3840	27.4389	24.5929	23.9397	23.5844	219.3428	218.3609	217.7083

Table 4: Average runtime per token (ms/token) across sampling strategies and models.

6.2 Discussions and Ablations

Sensitivity of the text to the temperature scaling. In this section, we present a quantitative analysis of how the coherence of generated text varies with changes in sampling temperature. We conducted experiments using the Qwen2.5-3B and LLaMA3.1-8B-Instruct models on prompts from the Alpaca-Eval dataset. To operationalize coherence, we use the total log-probability (log-likelihood) of the generated sequence as a proxy: higher total log-probability suggests that the model is more confident in the output, which we interpret as a signal of greater coherence.

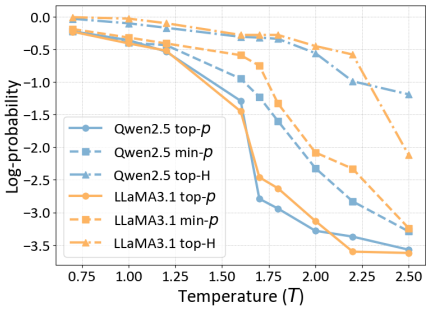

Figure 3: Effect of T scaling on generation coherence in min-p, top-p vs top-H.

Specifically, we compute the log-likelihood of each generated token during autoregressive generation, average these values across the entire sequence. This process is repeated in multiple temperature settings —0.7, 1.2, 1.6, 2.0, and 2.5—and for three different sampling strategies: top-p, min-p, and top-H. The result is portrayed in Fig. 3. As the temperature increases, the log-likelihood of the text generated under min-p and top-p sampling declines sharply. This suggests that the coherence of these methods is highly sensitive to temperature and that at higher temperatures, where increased creativity is encouraged, the generated

text tends to become less coherent. In contrast, top-H adjusts adaptively to the entropy of the distribution of the next token $H(p)$, effectively constraining randomness. As a result, it maintains more consistent and coherent output even in high-temperature settings.

Impact of α parameter. The only hyperparameter in top-H sampling is α, which directly controls the maximum allowable entropy for the distribution q. As such, a careful tuning of α is essential. To determine an appropriate value, we randomly select 50 development samples from the Alpaca-Eval dataset and use LLaMA3.1–8B–Instruct to generate responses. We explore values of α in the range [0.1, 0.9], with increments of 0.05. For each candidate value, we run the model on the development set and evaluate the outputs using our LLM-as-a-judge prompt (the same as in Section 6.1.3) to assess both creativity and coherence. The optimal value of α is selected based on its ability to best balance these two objectives. The results of the creativity and coherence evaluation, averaged over 50 development samples, are presented in Figure 4. As α increases, the entropy threshold becomes more permissive, allowing greater randomness in token selection. Consequently, creativity tends to increase, while coherence tends to decline. The optimal value of α is the point at which these two metrics are best balanced. Based on the figure, we observe that $\alpha = 0.4$ produces the highest average in the creativity and coherence scores, indicating it as the most suitable choice. Additional quantitative results are provided in Appendix C.5.

Empirical optimality of the top-H decoding strategy. We now empirically evaluate the competitiveness of the greedy algorithm of the top-H relative to the optimal solution of the ECMM problem, found by exhaustive search. We randomly sample 20 prompts from the Alpaca-Eval dataset and generate responses using the top-H method. At each generation step, the candidate set of tokens is restricted to the top-15 tokens of the probability distribution predicted by the model. To obtain the optimal solution, we exhaustively enumerate all possible 2^{15} subsets of the feasible token set and identify the subset S^* that maximizes the objective Γ_{S^*}, subject to the entropy constraint $H(q) \leq 0.4\, H(p)$, with q denoting the distribution over selected subset.

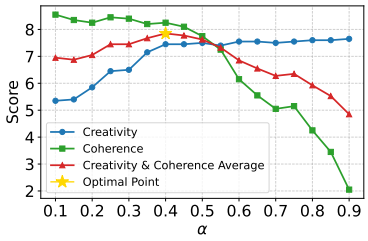

Figure 4: Effect of the parameter α on creativity and coherence.

For comparison, we also compute the greedy solution S^g using Algorithm 1. At each generation step, we calculate the ratio $\Gamma_{S^g}/\Gamma_{S^*}$, and report the mean and variance of this ratio (across different generation steps) for 20 different evaluation prompts, as visualized in Figure 5. As shown in the figure, the mean of the ratio remains consistently close to 1.0 across randomly sampled instances from the dataset, with only minor variance. Although deriving a formal approximation guarantee is beyond the scope of this work, our empirical results indicate that the solution obtained by top-H for the ECMM problem closely approximates the optimal solution in practice.

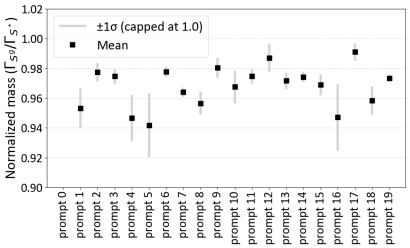

Figure 5: Empirical evaluation of top-H performance relative to the optimal solution of the ECMM problem.

While the primary goal of this work is to empirically demonstrate the efficacy of top-H decoding in addressing the ECMM, we defer a detailed theoretical analysis of the associated error bounds to future research. Nevertheless, in Appendix A.4, we provide a preliminary worst-case error bound for the greedy top-H solution under a specific assumption about the next-token probability distribution.

7 Conclusions

This paper addresses the challenge of balancing creativity and coherence in LLMs, particularly under high-temperature settings, where coherence often deteriorates. We introduce the entropy-constrained mass maximization (ECMM) problem, which formalizes the objective of balancing creativity and coherence by imposing an entropy constraint on the distribution of tokens in the sampling set. After proving the NP-hardness of ECMM, we propose top-H, a computationally efficient greedy algorithm that effectively approximates the solution of ECMM problem. Extensive empirical evaluation across various tasks demonstrates that top-H consistently outperforms established sampling strategies such as top-p and min-p, achieving up to **25.6**% higher accuracy. These results establish top-H as a new state-of-the-art method for creative writing in LLMs.

Acknowledgments

This work was partially supported by a grant from the Directorate for Computer and Information Science and Engineering (CISE) of the National Science Foundation.

References

Marah Abdin, Jyoti Aneja, Hany Awadalla, Ahmed Awadallah, Ammar Ahmad Awan, Nguyen Bach, Amit Bahree, Arash Bakhtiari, Jianmin Bao, Harkirat Behl, et al. Phi-3 technical report: A highly capable language model locally on your phone. *arXiv preprint arXiv:2404.14219*, 2024.

David H Ackley, Geoffrey E Hinton, and Terrence J Sejnowski. A learning algorithm for boltzmann machines. *Cognitive science*, 9(1):147–169, 1985.

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

Richard P Brent and Paul Zimmermann. *Modern computer arithmetic*, volume 18. Cambridge University Press, 2010.

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

EleutherAI. Lm evaluation harness, 2023. URL `https://github.com/EleutherAI/lm-evaluation-harness`. Version 0.4.5.

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

Michael R. Garey and David S. Johnson. Computers and intractability: A guide to the theory of np-completeness. In *W.H. Freeman*. W. H. Freeman and Company, 1979. Problem SP13: Cardinality-Constrained Subset Sum is NP-complete.

Martin Gerlach and Eduardo G Altmann. Stochastic model for the vocabulary growth in natural languages. *Physical Review X*, 3(2):021006, 2013.

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

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

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

Mina Lee, Percy Liang, and Qian Yang. Coauthor: Designing a human-ai collaborative writing dataset for exploring language model capabilities. In *Proceedings of the 2022 CHI conference on human factors in computing systems*, pp. 1–19, 2022.

Xuechen Li, Tianyi Zhang, Yann Dubois, Rohan Taori, Ishaan Gulrajani, Carlos Guestrin, Percy Liang, and Tatsunori B. Hashimoto. Alpacaeval: An automatic evaluator of instruction-following models. `https://github.com/tatsu-lab/alpaca_eval`, 5 2023.

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

OpenAI. Gpt-4o system card. *arXiv preprint arXiv:2410.21276*, 2024. URL `https://arxiv.org/abs/2410.21276`.

Christos H. Papadimitriou. *Computational Complexity*. Addison-Wesley, Reading, Massachusetts, 1994. ISBN 978-0-201-53082-7.

David Rein, Betty Li Hou, Asa Cooper Stickland, Jackson Petty, Richard Yuanzhe Pang, Julien Dirani, Julian Michael, and Samuel R Bowman. Gpqa: A graduate-level google-proof q&a benchmark. In *First Conference on Language Modeling*, 2024.

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

Lianmin Zheng, Wei-Lin Chiang, Ying Sheng, Siyuan Zhuang, Zhanghao Wu, Yonghao Zhuang, Zi Lin, Zhuohan Li, Dacheng Li, Eric Xing, et al. Judging llm-as-a-judge with mt-bench and chatbot arena. *Advances in Neural Information Processing Systems*, 36:46595–46623, 2023.

NeurIPS Paper Checklist

1. **Claims**

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

   Answer: [Yes]

   Justification: The abstract and introduction accurately reflect the paper's contributions and scope.

   Guidelines:

   - The answer NA means that the abstract and introduction do not include the claims made in the paper.
   - The abstract and/or introduction should clearly state the claims made, including the contributions made in the paper and important assumptions and limitations. A No or NA answer to this question will not be perceived well by the reviewers.
   - The claims made should match theoretical and experimental results, and reflect how much the results can be expected to generalize to other settings.
   - It is fine to include aspirational goals as motivation as long as it is clear that these goals are not attained by the paper.

2. **Limitations**

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

   Answer: [Yes]

   Justification: We provide a discussion of the limitations in Appendix D.

   Guidelines:

   - The answer NA means that the paper has no limitation while the answer No means that the paper has limitations, but those are not discussed in the paper.
   - The authors are encouraged to create a separate "Limitations" section in their paper.
   - The paper should point out any strong assumptions and how robust the results are to violations of these assumptions (e.g., independence assumptions, noiseless settings, model well-specification, asymptotic approximations only holding locally). The authors should reflect on how these assumptions might be violated in practice and what the implications would be.
   - The authors should reflect on the scope of the claims made, e.g., if the approach was only tested on a few datasets or with a few runs. In general, empirical results often depend on implicit assumptions, which should be articulated.
   - The authors should reflect on the factors that influence the performance of the approach. For example, a facial recognition algorithm may perform poorly when image resolution is low or images are taken in low lighting. Or a speech-to-text system might not be used reliably to provide closed captions for online lectures because it fails to handle technical jargon.
   - The authors should discuss the computational efficiency of the proposed algorithms and how they scale with dataset size.
   - If applicable, the authors should discuss possible limitations of their approach to address problems of privacy and fairness.
   - While the authors might fear that complete honesty about limitations might be used by reviewers as grounds for rejection, a worse outcome might be that reviewers discover limitations that aren't acknowledged in the paper. The authors should use their best judgment and recognize that individual actions in favor of transparency play an important role in developing norms that preserve the integrity of the community. Reviewers will be specifically instructed to not penalize honesty concerning limitations.

3. **Theory assumptions and proofs**

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

   Answer: [Yes]

Justification: The full set of assumptions and complete proofs are provided in Appendix A. The main intuitions for the results are provided in the main paper and complete proofs are provided in the appendix.

Guidelines:

- The answer NA means that the paper does not include theoretical results.
- All the theorems, formulas, and proofs in the paper should be numbered and cross-referenced.
- All assumptions should be clearly stated or referenced in the statement of any theorems.
- The proofs can either appear in the main paper or the supplemental material, but if they appear in the supplemental material, the authors are encouraged to provide a short proof sketch to provide intuition.
- Inversely, any informal proof provided in the core of the paper should be complemented by formal proofs provided in appendix or supplemental material.
- Theorems and Lemmas that the proof relies upon should be properly referenced.

4. **Experimental result reproducibility**

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

Answer: [Yes]

Justification: All the necessary information required to reproduce the experiments are discussed in section 6 in combination with the actual implementation in https://github.com/ErfanBaghaei/Top-H-Decoding.

Guidelines:

- The answer NA means that the paper does not include experiments.
- If the paper includes experiments, a No answer to this question will not be perceived well by the reviewers: Making the paper reproducible is important, regardless of whether the code and data are provided or not.
- If the contribution is a dataset and/or model, the authors should describe the steps taken to make their results reproducible or verifiable.
- Depending on the contribution, reproducibility can be accomplished in various ways. For example, if the contribution is a novel architecture, describing the architecture fully might suffice, or if the contribution is a specific model and empirical evaluation, it may be necessary to either make it possible for others to replicate the model with the same dataset, or provide access to the model. In general. releasing code and data is often one good way to accomplish this, but reproducibility can also be provided via detailed instructions for how to replicate the results, access to a hosted model (e.g., in the case of a large language model), releasing of a model checkpoint, or other means that are appropriate to the research performed.
- While NeurIPS does not require releasing code, the conference does require all submissions to provide some reasonable avenue for reproducibility, which may depend on the nature of the contribution. For example
  (a) If the contribution is primarily a new algorithm, the paper should make it clear how to reproduce that algorithm.
  (b) If the contribution is primarily a new model architecture, the paper should describe the architecture clearly and fully.
  (c) If the contribution is a new model (e.g., a large language model), then there should either be a way to access this model for reproducing the results or a way to reproduce the model (e.g., with an open-source dataset or instructions for how to construct the dataset).
  (d) We recognize that reproducibility may be tricky in some cases, in which case authors are welcome to describe the particular way they provide for reproducibility. In the case of closed-source models, it may be that access to the model is limited in some way (e.g., to registered users), but it should be possible for other researchers to have some path to reproducing or verifying the results.

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

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

   Answer: [Yes]

   Justification: Implementation of this work, including the full code repository is available at https://github.com/ErfanBaghaei/Top-H-Decoding.

   Guidelines:

   - The answer NA means that paper does not include experiments requiring code.
   - Please see the NeurIPS code and data submission guidelines (`https://nips.cc/public/guides/CodeSubmissionPolicy`) for more details.
   - While we encourage the release of code and data, we understand that this might not be possible, so "No" is an acceptable answer. Papers cannot be rejected simply for not including code, unless this is central to the contribution (e.g., for a new open-source benchmark).
   - The instructions should contain the exact command and environment needed to run to reproduce the results. See the NeurIPS code and data submission guidelines (`https://nips.cc/public/guides/CodeSubmissionPolicy`) for more details.
   - The authors should provide instructions on data access and preparation, including how to access the raw data, preprocessed data, intermediate data, and generated data, etc.
   - The authors should provide scripts to reproduce all experimental results for the new proposed method and baselines. If only a subset of experiments are reproducible, they should state which ones are omitted from the script and why.
   - At submission time, to preserve anonymity, the authors should release anonymized versions (if applicable).
   - Providing as much information as possible in supplemental material (appended to the paper) is recommended, but including URLs to data and code is permitted.

6. **Experimental setting/details**

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

   Answer: [Yes]

   Justification: This work does not involve model training; however, evaluation and implementation details are provided in Section 6.1.

   Guidelines:

   - The answer NA means that the paper does not include experiments.
   - The experimental setting should be presented in the core of the paper to a level of detail that is necessary to appreciate the results and make sense of them.
   - The full details can be provided either with the code, in appendix, or as supplemental material.

7. **Experiment statistical significance**

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

   Answer: [Yes]

   Justification: We conduct all experiments multiple times (e.g., five runs in Section 6.1.3) and report the average performance across runs. The standard deviation of the error is also reported when relevant, as in Figure 5.

   Guidelines:

   - The answer NA means that the paper does not include experiments.
   - The authors should answer "Yes" if the results are accompanied by error bars, confidence intervals, or statistical significance tests, at least for the experiments that support the main claims of the paper.

- The factors of variability that the error bars are capturing should be clearly stated (for example, train/test split, initialization, random drawing of some parameter, or overall run with given experimental conditions).
- The method for calculating the error bars should be explained (closed form formula, call to a library function, bootstrap, etc.)
- The assumptions made should be given (e.g., Normally distributed errors).
- It should be clear whether the error bar is the standard deviation or the standard error of the mean.
- It is OK to report 1-sigma error bars, but one should state it. The authors should preferably report a 2-sigma error bar than state that they have a 96% CI, if the hypothesis of Normality of errors is not verified.
- For asymmetric distributions, the authors should be careful not to show in tables or figures symmetric error bars that would yield results that are out of range (e.g. negative error rates).
- If error bars are reported in tables or plots, The authors should explain in the text how they were calculated and reference the corresponding figures or tables in the text.

8. **Experiments compute resources**

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

   Answer: [Yes]

   Justification: This information is provided in section 6.1.

   Guidelines:

   - The answer NA means that the paper does not include experiments.
   - The paper should indicate the type of compute workers CPU or GPU, internal cluster, or cloud provider, including relevant memory and storage.
   - The paper should provide the amount of compute required for each of the individual experimental runs as well as estimate the total compute.
   - The paper should disclose whether the full research project required more compute than the experiments reported in the paper (e.g., preliminary or failed experiments that didn't make it into the paper).

9. **Code of ethics**

   Question: Does the research conducted in the paper conform, in every respect, with the NeurIPS Code of Ethics `https://neurips.cc/public/EthicsGuidelines`?

   Answer: [Yes]

   Justification: This paper conforms, in every aspect, with the NeurIPS Code of Ethics.

   Guidelines:

   - The answer NA means that the authors have not reviewed the NeurIPS Code of Ethics.
   - If the authors answer No, they should explain the special circumstances that require a deviation from the Code of Ethics.
   - The authors should make sure to preserve anonymity (e.g., if there is a special consideration due to laws or regulations in their jurisdiction).

10. **Broader impacts**

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

    Answer: [Yes]

    Justification: This is provided in the Appendix E.

    Guidelines:

    - The answer NA means that there is no societal impact of the work performed.
    - If the authors answer NA or No, they should explain why their work has no societal impact or why the paper does not address societal impact.

- Examples of negative societal impacts include potential malicious or unintended uses (e.g., disinformation, generating fake profiles, surveillance), fairness considerations (e.g., deployment of technologies that could make decisions that unfairly impact specific groups), privacy considerations, and security considerations.
- The conference expects that many papers will be foundational research and not tied to particular applications, let alone deployments. However, if there is a direct path to any negative applications, the authors should point it out. For example, it is legitimate to point out that an improvement in the quality of generative models could be used to generate deepfakes for disinformation. On the other hand, it is not needed to point out that a generic algorithm for optimizing neural networks could enable people to train models that generate Deepfakes faster.
- The authors should consider possible harms that could arise when the technology is being used as intended and functioning correctly, harms that could arise when the technology is being used as intended but gives incorrect results, and harms following from (intentional or unintentional) misuse of the technology.
- If there are negative societal impacts, the authors could also discuss possible mitigation strategies (e.g., gated release of models, providing defenses in addition to attacks, mechanisms for monitoring misuse, mechanisms to monitor how a system learns from feedback over time, improving the efficiency and accessibility of ML).

11. **Safeguards**

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

    Answer: [NA]

    Justification: This paper poses no such risks.

    Guidelines:

    - The answer NA means that the paper poses no such risks.
    - Released models that have a high risk for misuse or dual-use should be released with necessary safeguards to allow for controlled use of the model, for example by requiring that users adhere to usage guidelines or restrictions to access the model or implementing safety filters.
    - Datasets that have been scraped from the Internet could pose safety risks. The authors should describe how they avoided releasing unsafe images.
    - We recognize that providing effective safeguards is challenging, and many papers do not require this, but we encourage authors to take this into account and make a best faith effort.

12. **Licenses for existing assets**

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

    Answer: [Yes]

    Justification: The assets used in this paper are properly cited.

    Guidelines:

    - The answer NA means that the paper does not use existing assets.
    - The authors should cite the original paper that produced the code package or dataset.
    - The authors should state which version of the asset is used and, if possible, include a URL.
    - The name of the license (e.g., CC-BY 4.0) should be included for each asset.
    - For scraped data from a particular source (e.g., website), the copyright and terms of service of that source should be provided.
    - If assets are released, the license, copyright information, and terms of use in the package should be provided. For popular datasets, `paperswithcode.com/datasets` has curated licenses for some datasets. Their licensing guide can help determine the license of a dataset.

- For existing datasets that are re-packaged, both the original license and the license of the derived asset (if it has changed) should be provided.
- If this information is not available online, the authors are encouraged to reach out to the asset's creators.

13. **New assets**

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

    Answer: [NA]

    Justification: No new assets are created.

    Guidelines:

    - The answer NA means that the paper does not release new assets.
    - Researchers should communicate the details of the dataset/code/model as part of their submissions via structured templates. This includes details about training, license, limitations, etc.
    - The paper should discuss whether and how consent was obtained from people whose asset is used.
    - At submission time, remember to anonymize your assets (if applicable). You can either create an anonymized URL or include an anonymized zip file.

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

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

    Answer: [NA]

    Justification: This paper does not involve crowdsourcing nor research with human subjects.

    Guidelines:

    - The answer NA means that the paper does not involve crowdsourcing nor research with human subjects.
    - Including this information in the supplemental material is fine, but if the main contribution of the paper involves human subjects, then as much detail as possible should be included in the main paper.
    - According to the NeurIPS Code of Ethics, workers involved in data collection, curation, or other labor should be paid at least the minimum wage in the country of the data collector.

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

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

    Answer: [NA]

    Justification: This paper does not involve crowdsourcing nor research with human subjects.

    Guidelines:

    - The answer NA means that the paper does not involve crowdsourcing nor research with human subjects.
    - Depending on the country in which research is conducted, IRB approval (or equivalent) may be required for any human subjects research. If you obtained IRB approval, you should clearly state this in the paper.
    - We recognize that the procedures for this may vary significantly between institutions and locations, and we expect authors to adhere to the NeurIPS Code of Ethics and the guidelines for their institution.
    - For initial submissions, do not include any information that would break anonymity (if applicable), such as the institution conducting the review.

16. **Declaration of LLM usage**

Question: Does the paper describe the usage of LLMs if it is an important, original, or non-standard component of the core methods in this research? Note that if the LLM is used only for writing, editing, or formatting purposes and does not impact the core methodology, scientific rigorousness, or originality of the research, declaration is not required.

Answer: [NA]

Justification: The core method development in this research does not involve LLMs as any important, original, or non-standard components.

Guidelines:

- The answer NA means that the core method development in this research does not involve LLMs as any important, original, or non-standard components.
- Please refer to our LLM policy (`https://neurips.cc/Conferences/2025/LLM`) for what should or should not be described.

A Proofs of Theorems

Unless otherwise specified, all logarithms are natural log.

A.1 Proof of Theorem 1

Proof of Theorem 1. The probability of the selected tokens needs to be divided by their sum, to make sure that the sum of the distribution q is 1. Given:

$$\sum_i p_i = 1 \quad \Rightarrow \quad \sum_i p_i \mathbb{1}_{\{v_i \notin S\}} = 1 - \Gamma_S \tag{6}$$

According to 4:

$$M = \begin{cases} \frac{p_i + \frac{p_i}{\Gamma_S}}{2} & i \leq m \\ \frac{p_i}{2} & i > m \end{cases}$$

$$D_{KL}(p||M) = \sum_{i=1}^{m} p_i \log\left(\frac{p_i}{\frac{1}{2}(p_i + \frac{p_i}{\Gamma_S})}\right) + \sum_{i=m+1}^{n} p_i \log\left(\frac{p_i}{\frac{1}{2}p_i}\right) = \sum_{i=1}^{m} p_i \log\left(\frac{2\Gamma_S}{1+\Gamma_S}\right) + \sum_{i=m+1}^{n} p_i \log(2)$$

$$= \log\left(\frac{2\Gamma_S}{1+\Gamma_S}\right) \sum_{i=1}^{m} p_i + \log(2) \sum_{i=m+1}^{n} p_i$$

Using (3) and (6):

$$= \Gamma_S \log\left(\frac{2\Gamma_S}{1+\Gamma_S}\right) + \log(2)(1-\Gamma_S) = \Gamma_S \left[\log\left(\frac{\Gamma_S}{1+\Gamma_S}\right) + \log(2)\right] + \log(2)(1-\Gamma_S)$$

$$= \Gamma_S \log\left(\frac{\Gamma_S}{1+\Gamma_S}\right) + \log(2)\Gamma_S + \log(2) - \log(2)\Gamma_S = \Gamma_S \log\left(\frac{\Gamma_S}{1+\Gamma_S}\right) + \log(2) \tag{7}$$

$$D_{KL}(q||M) = \sum_{i=1}^{m} \frac{p_i}{\Gamma_S} \log\left(\frac{\frac{p_i}{\Gamma_S}}{\frac{1}{2}(p_i + \frac{p_i}{\Gamma_S})}\right) = \frac{1}{\Gamma_S} \sum_{i=1}^{m} p_i \log\left(\frac{2}{1+\Gamma_S}\right) = \frac{1}{\Gamma_S} \log\left(\frac{2}{1+\Gamma_S}\right) \sum_{i=1}^{m} p_i$$

Using (3):

$$= \log\left(\frac{2}{1+\Gamma_S}\right) = \log(2) - \log(1+\Gamma_S) \tag{8}$$

Rewriting Jensen-Shannon Divergence using (7) and (8):

$$\text{JSD}(p||q) = \frac{1}{2}\left(\log(2) + \Gamma_S \log\left(\frac{\Gamma_S}{1+\Gamma_S}\right)\right) + \frac{1}{2}\left(\log(2) - \log(1+\Gamma_S)\right)$$

$$= \frac{1}{2}\left(2\log(2) + \Gamma_S \log(\Gamma_S) - \Gamma_S \log(1+\Gamma_S) - \log(1+\Gamma_S)\right) = \boxed{\log(2) + \frac{1}{2}\left(\Gamma_S \log(\Gamma_S) - (1+\Gamma_S)\log(1+\Gamma_S)\right)}$$

Therefore, JSD is only dependent on Γ_S.
Now we want to show that the distance between the distributions is decreasing with respect to Γ_S:

$$\frac{d}{d\Gamma_S}\text{JSD}(p||q) = \frac{1}{2}\left(\log(\Gamma_S) + 1 - \log(1+\Gamma_S) - 1\right)$$

$$= \frac{1}{2}\left(\log(\Gamma_S) - \log(1+\Gamma_S)\right) \quad \rightarrow \quad \text{Always negative.}$$

Therefore, $\text{JSD}(p||q)$ is decreasing with respect to Γ_S and to minimize $\text{JSD}(p||q)$, one needs to maximize Γ_S. □

A.2 NP-Hardness of Entropy-Constrained Mass Maximization

All logarithms are natural (ln). Arithmetic is performed on a *unitcost RAM with binary encodings*; an integer $x \geq 1$ occupies $\lfloor \log_2 x \rfloor + 1$ bits.

A.2.1 Problem definition

For a probability vector $\mathbf{p} = (p_1, \ldots, p_n)$ with $\sum_i p_i = 1$, define

$$H(\mathbf{p}) := -\sum_{i=1}^{n} p_i \ln p_i.$$

The fixed–budget maximization problem ECMM is

$$\max_{S \subseteq [n]} \Gamma_S := \sum_{i \in S} p_i \quad \text{s.t.} \quad H(S) \leq 0.4 \cdot H(\mathbf{p}), \tag{ECMM}$$

where $H(S)$ denotes the entropy of the renormalized vector $(p_i)_{i \in S}$.

The corresponding decision version, ECME, is:

> *Input:* A probability vector $\mathbf{p} = (p_1, \ldots, p_n)$, a mass target $\beta = \frac{2}{3}$, and the fixed budget $\alpha = 0.4 \cdot H(\mathbf{p})$.
> *Question:* Does there exist $S \subseteq [n]$ such that
>
> $$\sum_{i \in S} p_i = \beta \quad \text{and} \quad H(S) \leq \alpha?$$

We show that the decision variant is NP–complete; the optimization variant is NP–hard.

A.2.2 Source problem: Cardinality–Constrained Subset Sum

Definition 1 (CCSS)**.** *Given positive integers $w_1, \ldots, w_m$, a target τ, and an integer K with $3 \leq K \leq m$, decide whether some subset of exactly K weights sums to τ.*

CCSS is NP–complete: reduce from the classic SUBSET–SUM; see, e.g., Papadimitriou (Papadimitriou, 1994, Exercise 8.14).

Our reduction from CCSS to (ECMM) needs the following *narrow–range* condition.

Assumption 1 (Narrow range)**.**

$$\frac{\tau}{K+1} < w_i < \frac{\tau}{K-1} \quad (1 \leq i \leq m).$$

The next lemma shows that we may enforce Assumption 1 by a *polynomial–time padding* step.

Lemma 1 (Padding to narrow range)**.** *There is a polynomial–time transformation that maps an arbitrary* CCSS *instance $(w_1, \ldots, w_m; \tau; K)$ to an equivalent instance $(w'_1, \ldots, w'_m; \tau'; K)$ that satisfies Assumption 1. The new weights and target have binary lengths polynomial in the original instance size.*

Proof. Set

$$M := (K+1)\,\tau, \qquad w'_i := w_i + M, \qquad \tau' := \tau + K\,M.$$

Because the same constant M is added to every weight, a subset of exactly K items sums to τ iff it sums to τ':

$$\sum_{i \in S} w_i = \tau \iff \sum_{i \in S} w'_i = \tau'.$$

Lower bound. Each new weight satisfies $w'_i > M$. Moreover

$$\frac{\tau'}{K+1} = \frac{\tau + K(K+1)\tau}{K+1} = KT + \frac{\tau}{K+1} < K\tau + \tau = M,$$

so $w'_i > \tau'/(K+1)$.

Upper bound. We have $w_i' \le w_{\max} + M \le \tau + (K+1)\tau = (K+2)\tau$. On the other hand,

$$\frac{\tau'}{K-1} \;=\; \frac{(K(K+1)+1)\tau}{K-1} \;=\; \frac{K^2+K+1}{K-1}\,\tau.$$

Since $(K+2)(K-1) = K^2+K-2 < K^2+K+1$ for every $K \ge 3$, it follows that $w_i' < \tau'/(K-1)$. Therefore Assumption 1 holds for the padded instance.

Encoding size. The multiplier $M = (K+1)\tau$ increases the bit–length of the largest weight by at most $\log_2(K+1)$ bits, and the same holds for τ'. Hence the transformation is polynomial in the input length. □

Henceforth we assume without loss of generality that every CCSS instance meets Assumption 1; if it does not, we first apply the padding from Lemma 1.

We also stipulate $K \ge 20$ (duplicate the instance as below if necessary).

Scaling step (making $K \ge 20$ while keeping the narrow range). If the given instance has $K < 20$, put

$$d \;:=\; \lceil 20/K \rceil,$$

duplicate *every* weight d times and set

$$K_1 := dK, \qquad \tau_1 := d\tau.$$

Then

$$\exists\, S \subseteq [m] : |S| = K,\ \sum_{i\in S} w_i = \tau \iff \exists\, S \subseteq [m] : |S| = K_1,\ \sum_{i\in S} w_i = \tau_1,$$

so feasibility is preserved and $K_1 \ge 20$.

The plain duplication, however, may violate Assumption 1 because the new lower bound $\tau_1/(K_1+1)$ can exceed some of the duplicated weights. To restore the assumption we now *re-apply* the padding of Lemma 1 to the instance $(w_1, \dots, w_m;\, \tau_1;\, K_1)$. This yields an equivalent instance

$$(w_1', \dots, w_m';\, \tau';\, K_1)$$

that *satisfies* Assumption 1 and still has $K_1 \ge 20$.

All numbers grow by at most $\lceil \log_2 d \rceil + O(\log K)$ bits, so the whole transformation remains polynomial-time.

Henceforth we may—and do—assume that $K \ge 20$ and that the narrow-range condition holds.

A.2.3 Reduction to ECME

Let $(w_1, \dots, w_m; \tau; K)$ be a CCSS instance that already satisfies Assumption 1. Define

$$\gamma_K := \frac{1}{16K^2}, \quad \theta_K < \frac{1}{2K^2}, \quad \delta_K := \frac{5\theta_K}{2\ln K}, \quad \varepsilon_K := \frac{0.0384 + \gamma_K}{\ln K},$$

$$\lambda(K) := \left\lceil \frac{0.7333 - \varepsilon_K + \delta_K}{0.133} \right\rceil, \quad B := \lceil K^{\lambda(K)} \rceil,$$

$$w_b := \frac{\tau}{2B}, \quad \tau_b := Bw_b = \tfrac{1}{2}\tau, \quad W := \tau + \tau_b = \tfrac{3}{2}\tau.$$

Set

$$p_i := \frac{w_i}{W} \quad (1 \le i \le m), \quad p_b := \frac{w_b}{W} = \frac{1}{3B}, \quad \beta := \frac{\tau}{W} = \tfrac{2}{3}.$$

The resulting ECME instance contains $n = m + B$ items. The numbers above are representable with $O(\log K)$ bits, hence the reduction runs in polynomial time.

A.2.4 Entropy budget window

Lemma 2 (Budget window)**.** *For the constructed instance,*

$$\ln K - \gamma_K < 0.4\, H(\mathbf{p}) < \ln(K+1).$$

Proof. Split $H(\mathbf{p}) = H_{\mathrm{h}} + H_{\mathrm{b}}$, where

$$H_{\mathrm{h}} = \tfrac{2}{3}(H(\mathbf{q}) + \ln \tfrac{2}{3}), \quad q_i := \frac{w_i}{\tau},$$
$$H_{\mathrm{b}} = \tfrac{1}{3}(\ln 3 + \lambda(K)\, \ln K).$$

By Assumption 1 and a second–order Taylor bound, $H(\mathbf{q}) = \ln K - \theta_K$ with $0 < \theta_K < \frac{1}{2K^2}$. Substitution gives

$$0.4\, H(\mathbf{p}) = (1-\varepsilon_K)\ln K + 0.0384 - 0.2667\,\theta_K + 0.133\,\delta_K\, \ln K = \ln K - \gamma_K + 0.0666\,\theta_K,$$

and since $0 < 0.0666\,\theta_K < \ln(1 + 1/K)$ for $K \geq 20$, the window follows. □

A.2.5 Structural lemmas

Lemma 3 (Booster blow-up)**.** *Let S be any subset with $\Gamma_S = \beta$. If S contains at least one booster item, then $H(S) > 0.4\, H(\mathbf{p})$.*

Proof. Write $S = H \cup B$ where H (resp. B) is the set of heavy (resp. booster) indices selected. Let $b := |B| \geq 1$ and $L := |H|$.

Step 1 – how many boosters are needed. Each booster weighs $w_b = \tau/(2B)$, whereas every heavy weight is at least $w_{\min} = \tau/(K+1)$ by Assumption 1. Total weight has to be exactly τ, so each heavy item that is *removed* must be replaced by at least

$$\frac{w_{\min}}{w_b} \;=\; \frac{\tau/(K+1)}{\tau/(2B)} \;=\; \frac{2B}{K+1} \;>\; \frac{2B}{K}$$

boosters. Therefore $L \leq K - 1$ and

$$b \;\geq\; \frac{2B}{K}, \qquad \delta := \frac{b}{2B} \;\geq\; \frac{1}{K}. \tag{9}$$

(The quantity δ equals the total probability mass of the boosters after renormalisation because each has probability $w_b/\tau = 1/(2B)$.)

Step 2 – a lower bound on the entropy of S. The booster probabilities are all $1/(2B)$, so their contribution is $\delta \ln(2B)$. For the heavy part we use the crude bound $\ln(K-1) \leq \ln K - 1/K$ together with the fact that the L heavy probabilities add up to $1 - \delta$:

$$H(S) \;\geq\; (1-\delta)\ln(K-1) + \delta \ln(2B) \;\geq\; \ln K - \frac{1}{K} + \delta\big((\lambda(K)-1)\ln K + \ln 2\big),$$

because $\ln(2B) = \ln 2 + \lambda(K)\ln K$ by definition of B. With (9) this gives

$$H(S) \;\geq\; \ln K \;+\; \frac{(\lambda(K)-1)\ln K + \ln 2 - 1}{K}.$$

Since $\lambda(K) \geq 2$ for every $K \geq 20$, the numerator is positive and we conclude

$$H(S) > \ln K. \tag{10}$$

Step 3 – compare with the budget. Lemma 2 states $0.4\, H(\mathbf{p}) < \ln K$. Combining this with (10) proves $H(S) > 0.4\, H(\mathbf{p})$, as required. □

Lemma 4 (Cardinality lock)**.** *If S contains no boosters and $\Gamma_S = \beta = 2/3$, then $|S| = K$ and $\sum_{i \in S} w_i = \tau$.*

Proof. Since S has no boosters, its total weight is

$$\sum_{i \in S} w_i = W \, \Gamma_S = \tau.$$

Let $|S| = L$. By the narrow–range assumption,

$$L \cdot \frac{\tau}{K+1} < \sum_{i \in S} w_i = \tau < L \cdot \frac{\tau}{K-1}.$$

Dividing through by τ gives

$$\frac{L}{K+1} < 1 < \frac{L}{K-1},$$

which simplifies to $K - 1 < L < K + 1$. Since L is an integer, $L = K$. Having $|S| = K$ and total weight τ establishes the claim. □

Lemma 5 (Entropy gap for a K–heavy subset)**.** *Every K–element subset S of heavy items summing to τ satisfies*

$$H(S) \leq \ln K - \gamma_K.$$

Proof. After selecting K heavy items summing to τ, their renormalised probabilities are $r_i = w_i/\tau$. By the narrow–range assumption,

$$\frac{1}{K+1} < r_i < \frac{1}{K-1},$$

so we may write $r_i = \frac{1}{K} + x_i$ with $|x_i| < \frac{1}{K(K-1)}$ and $\sum_i x_i = 0$.

Then

$$H(S) = -\sum_{i=1}^{K} r_i \ln r_i = -\sum_{i=1}^{K} \Big(\tfrac{1}{K} + x_i\Big) \ln\Big(\tfrac{1}{K} + x_i\Big) = \ln K - \sum_{i=1}^{K} \Big(\tfrac{1}{K} + x_i\Big) \ln\big(1 + K x_i\big).$$

Using the Taylor approximation $\ln(1+u) \geq u - \frac{u^2}{2}$ for $|u| < 1$,

$$\big(\tfrac{1}{K} + x_i\big) \ln(1 + K x_i) \;\geq\; \big(\tfrac{1}{K} + x_i\big)\Big(K x_i - \tfrac{(K x_i)^2}{2}\Big) = x_i + \tfrac{K x_i^2}{2} - K^2 x_i^3/2.$$

Summing over i and using $\sum_i x_i = 0$ and $|x_i| < 1/(K(K-1))$ gives

$$\sum_{i=1}^{K} \Big(\tfrac{1}{K} + x_i\Big) \ln(1 + K x_i) \;\geq\; \frac{K}{2} \sum_{i=1}^{K} x_i^2 - \frac{K^2}{2} \sum_{i=1}^{K} |x_i|^3 > \frac{1}{2(K-1)^2} - \frac{1}{2(K-1)^3} = \frac{K-2}{2(K-1)^3}.$$

For $K \geq 20$, one checks

$$\frac{K-2}{2(K-1)^3} \;>\; \frac{1}{16K^2} = \gamma_K.$$

Hence

$$H(S) = \ln K - \sum_{i} \Big(\tfrac{1}{K} + x_i\Big) \ln(1 + K x_i) \leq \ln K - \gamma_K,$$

as claimed. □

A.2.6 Equivalence theorem

Theorem 4. *The constructed* ECME *instance admits a subset of mass β* iff *the original* CCSS *instance is a YES instance.*

Proof. ($\Rightarrow$) Any feasible S must exclude boosters by Lemma 3, so Lemma 4 gives a K–subset summing to τ.
($\Leftarrow$) Conversely, let S be any K–element subset summing to τ. By Lemma 5, $H(S) \leq \ln K - \gamma_K$, and by Lemma 2, $\ln K - \gamma_K < 0.4\, H(\mathbf{p})$. Hence $H(S) < 0.4\, H(\mathbf{p})$ and $\Gamma_S = \beta$, so S is feasible. □

A.2.7 Complexity consequence

Theorem 5. *The decision variant* ECME *is* **NP–complete***, and the corresponding optimization problem* ECMM *is* **NP–hard***.*

Proof. **Membership in NP** Given a candidate subset S, we can verify both constraints in polynomial time. For the mass we simply add the p_i's. For the entropy, we approximate each $\ln$ to $O(\log K)$ bits; the required precision is well below the separating gap $\gamma_K - 0.0666\,\theta_K = \Theta(K^{-2})$, so rounding cannot flip the inequality. Classical results on transcendental evaluation on a unit–cost RAM (Brent & Zimmermann, 2010) show that such an approximation takes $\tilde{O}((\log K)^2)$ time—polynomial in the input size. Hence the verifier runs in polynomial time.

NP–hardness of the decision problem. Apply the polynomial–time padding from Lemma 1 and then the reduction of Section A.2.3. By Theorem 4, the resulting instance is a *YES* instance of ECME iff the original CCSS instance is a *YES* instance. Therefore the decision problem is NP–hard.

Optimization hardness. Assume, for contradiction, that we had a polynomial–time algorithm that returns

$$\max_{S\subseteq[n]} \Gamma_S \quad \text{s.t.} \quad H(S) \le 0.4 \cdot H(\mathbf{p}).$$

On the same input we could decide the ECME instance by a single comparison of that maximum with the fixed target value $\beta = \frac{2}{3}$. This would solve an NP–complete problem in polynomial time, contradicting $\mathsf{P} \neq \mathsf{NP}$. Hence the optimization version is **NP–hard**. □

A.3 Proof of Early Termination

Proof of Theorem 3. Assume that the distribution of the selected tokens after j steps is q^j, therefore:

$$H(q^{j-1}) = -\sum_{i=1}^{j-1} \frac{p_i}{\Gamma_{j-1}} \log\left(\frac{p_i}{\Gamma_{j-1}}\right) = -\frac{1}{\Gamma_{j-1}} \sum_{i=1}^{j-1} p_i \log p_i + \frac{\log(\Gamma_{j-1})}{\Gamma_{j-1}} \sum_{i=1}^{j-1} p_i, \quad \text{where } \Gamma_{j-1} = \sum_{i=1}^{j-1} p_i$$

Since $\sum_{i=1}^{j-1} p_i = \Gamma_{j-1}$:

$$H(q^{j-1}) = \log(\Gamma_{j-1}) - \frac{1}{\Gamma_{j-1}} \sum_{i=1}^{j-1} p_i \log p_i$$

$$\therefore -\sum_{i=1}^{j-1} p_i \log p_i = \Gamma_{j-1}(H(q^{j-1}) - \log(\Gamma_{j-1})) \tag{11}$$

Now, calculating $H(q^j)$:

$$H(q^j) = -\sum_{i=1}^{j} \frac{p_i}{\Gamma_{j-1}+p_j} \log\left(\frac{p_i}{\Gamma_{j-1}+p_j}\right) = -\sum_{i=1}^{j} \frac{p_i}{\Gamma_{j-1}+p_j} \log p_i + \sum_{i=1}^{j} \frac{p_i}{\Gamma_{j-1}+p_j} \log(\Gamma_{j-1}+p_j)$$

$$= -\frac{1}{\Gamma_{j-1}+p_j} \sum_{i=1}^{j} p_i \log p_i + \frac{\log(\Gamma_{j-1}+p_j)}{\Gamma_{j-1}+p_j} \sum_{i=1}^{j} p_i$$

Since $\sum_{i=1}^{j} p_i = \sum_{i=1}^{j-1} p_i + p_j = \Gamma_{j-1} + p_j$:

$$H(q^j) = \log(\Gamma_{j-1}+p_j) - \frac{1}{\Gamma_{j-1}+p_j} \sum_{i=1}^{j} p_i \log p_i = \log(\Gamma_{j-1}+p_j) + \frac{1}{\Gamma_{j-1}+p_j} \left(-\sum_{i=1}^{j-1} p_i \log p_i - p_j \log p_j\right)$$

Using (11):

$$= \log(\Gamma_{j-1} + p_j) + \frac{1}{\Gamma_{j-1} + p_j} \left(\Gamma_{j-1} H(q^{j-1}) - \Gamma_{j-1} \log(\Gamma_{j-1}) - p_j \log p_j \right)$$

$$\therefore \Delta H = H(q^j) - H(q^{j-1})$$

$$= \log(\Gamma_{j-1} + p_j) + \frac{1}{\Gamma_{j-1} + p_j} \left(\Gamma_{j-1} H(q^{j-1}) - \Gamma_{j-1} \log(\Gamma_{j-1}) - p_j \log p_j \right)) - H(q^{j-1})$$

$$= \log(\Gamma_{j-1}+p_j)+\frac{1}{\Gamma_{j-1} + p_j} \left(\Gamma_{j-1} H(q^{j-1}) - (\Gamma_{j-1} + p_j) H(q^{j-1}) - \Gamma_{j-1} \log(\Gamma_{j-1}) - p_j \log(p_j) \right)$$

$$= \log(\Gamma_{j-1} + p_j) - \frac{1}{\Gamma_{j-1} + p_j} \left(p_j H(q^{j-1}) + \Gamma_{j-1} \log(\Gamma_{j-1}) + p_j \log(p_j) \right)$$

The probabilities are sorted in descending order, therefore:

$$\forall i \leq j : p_j \leq p_i \Rightarrow \sum_{i=1}^{j-1} p_j \leq \sum_{i=1}^{j-1} p_i \Rightarrow (j-1) p_j \leq \Gamma_{j-1} \Rightarrow j - 1 \leq \frac{\Gamma_{j-1}}{p_j}$$

$$\therefore \quad H(q^{j-1}) \leq \log(j-1) \leq \log \left(\frac{\Gamma_{j-1}}{p_j} \right)$$

$$\therefore \quad \Delta H \geq \log(\Gamma_{j-1} + p_j) - \frac{p_j \log \left(\frac{\Gamma_{j-1}}{p_j} \right)}{\Gamma_{j-1} + p_j} - \frac{\Gamma_{j-1} \log(\Gamma_{j-1}) + p_j \log(p_j)}{\Gamma_{j-1} + p_j}$$

$$= \log(\Gamma_{j-1} + p_j) - \frac{1}{\Gamma_{j-1} + p_j} \left[p_j \log(\Gamma_{j-1}) - p_j \log(p_j) + \Gamma_{j-1} \log(\Gamma_{j-1}) + p_j \log(p_j) \right]$$

$$= \log(\Gamma_{j-1} + p_j) - \frac{1}{\Gamma_{j-1} + p_j} \left(p_j \log(\Gamma_{j-1}) + \Gamma_{j-1} \log(\Gamma_{j-1}) \right)$$

$$= \log(\Gamma_{j-1} + p_j) - \log(\Gamma_{j-1}) = \log \left(\frac{\Gamma_{j-1} + p_j}{\Gamma_{j-1}} \right) = \log \left(1 + \frac{p_j}{\Gamma_{j-1}} \right)$$

$$\therefore \quad \Delta H \geq \log \left(1 + \frac{p_j}{\Gamma_{j-1}} \right) > 0 \Rightarrow \Delta H > 0$$

□

A.4 Formal approximation bound for the top-H

Zipf model.

Fix a vocabulary of size n and exponent $s > 1$. We assume the sorted next-token probabilities obey the classical Zipf / regularly varying law:

$$p_i := \frac{i^{-s}}{H_{n,s}}, \quad H_{n,s} = \sum_{j=1}^{n} j^{-s}. \tag{12}$$

Empirically, language-model logits are well approximated by $s \in [1.05, 1.20]$ (Gerlach & Altmann, 2013); hence the assumption captures current practice.

The Zipf assumption matters, because ECMM is NP-hard (Theorem 2), exact polynomial-time solutions are unlikely. In the absence of structural assumptions, a constant-factor approximation guarantee for ECMM is highly unlikely unless $P = NP$, as our NP-hardness proof relies on a gap-preserving reduction from Cardinality-Constrained Subset Sum. This structure is known to be fundamentally difficult to approximate, a standard result in computational complexity theory (Papadimitriou, 1994). However, LLM logits consistently follow heavy-tailed (Zipf / regularly-varying) laws, so analysing this regime yields practically relevant bounds.

Notation. We write $M(k) = \sum_{i \leq k} p_i$ for the prefix mass, $T(k) = 1 - M(k)$ for the tail mass, and $H(k)$ for the entropy of the normalised prefix distribution $q_i^{(k)} = p_i / M(k)$.

Preliminaries

Lemma 6 (Monotonicity of prefix entropy)**.** *For $1 \le k < n$ one has $H(k) < H(k+1)$. Moreover, for any subset S of size k, $H(q_S) \ge H(k)$.*

Proof. The first claim is classical: adding the $(k+1)$-st symbol strictly increases entropy because $p_{k+1} < p_k$ and entropy is Schur-concave. For the second claim we note that $(p_1, \dots, p_k)$ majorises any other k-subset of the sorted vector; Schur-concavity again yields the desired inequality. □

Lemma 7 (Tail mass bound)**.** *For $k < n$,*

$$\frac{(k+1)^{1-s} - n^{1-s}}{(s-1)H_{n,s}} \le T_n(k) \le \frac{k^{1-s}}{(s-1)H_{n,s}}$$

When $n \gg k$, the numerator difference is $o(k^{1-s})$, so the upper bound is asymptotically tight.

Proof. Apply upper and lower Riemann sums to $\int_k^n x^{-s}\,dx$.

□

Lemma 8 (Prefix entropy asymptotics)**.** *There exist constants $c_s, C_s > 0$ such that*

$$c_s + \frac{s}{s-1}\log(\frac{k}{2}) \le H(k) \le C_s + \frac{s}{s-1}\log k \quad \textit{for all } k \ge 2.$$

Proof. Combine Lemma 7 with integral bounds for $\sum i^{-s}\log i$. □

Depth of the greedy prefix

Let $k_g := \max\{k : H(k) \le \alpha H(n)\}$. Lemma 6 implies k_g is well defined.

Lemma 9 (Growth rate of k_g)**.** *There exist constants $a_s, b_s > 0$ such that*

$$a_s n^\alpha \le k_g \le b_s n^\alpha.$$

Proof. Insert Lemma 8 into the defining inequality and solve for k_g. □

Mass captured by top-H

Write $\Gamma_g = M(k_g)$. This is the mass captured by the greedy solution. By construction, this solution is valid as it satisfies the entropy constraint $H(k_g) \le \alpha H(n)$.

Tight upper bound for ECMM

Let $S^\star$ be any subset satisfying the entropy constraint, and set $\Gamma^\star = \sum_{i \in S^\star} p_i$. We want to bound the maximum possible value of $\Gamma^\star$.

The greedy algorithm selects the prefix $[k_g]$ and captures a mass of $\Gamma_g = M(k_g) = 1 - T(k_g)$. The total mass of all tokens not in the greedy solution is, by definition, the tail mass $T(k_g)$.

Any other valid solution $S^\star$ can, at best, capture the mass of the greedy solution *plus* some portion of the remaining tail mass. The absolute maximum mass any solution can capture is 1 (the entire vocabulary). Therefore, the maximum possible improvement any optimal solution $\Gamma^\star$ can have over the greedy solution Γ_g is bounded by the tail mass that the greedy algorithm left behind:

$$\Gamma^\star - \Gamma_g \le 1 - \Gamma_g = 1 - M(k_g) = T(k_g). \tag{13}$$

This gives us a direct upper bound on the additive gap between the optimal and greedy solutions.

Theorem 6 (Distribution-dependent additive guarantee)**.** *Under equation 12, for every $n \ge 4$,*

$$\Gamma^\star - \Gamma_g \le T_n(k_g) \le \frac{k_g^{s-1}}{(s-1)H_{n,s}} = \mathcal{O}(n^{-\alpha(s-1)})$$

Proof. From 13, we have the additive gap $\Gamma^\star - \Gamma_g \le T(k_g) = T_n(k_g)$. Lemma 7 bounds $T_n(k_g)$ by $\frac{k_g^{1-s}}{(s-1)H_{n,s}}$. Finally, lemma 9 gives $k_g = \Theta(n^\alpha)$, so the gap decays as $\mathcal{O}(n^{-\alpha(s-1)})$. □

Discussion on the effectiveness of greedy

We understand that the approach of dropping high-probability tokens could in principle beat the greedy prefix by allowing more low-mass tokens from the tail. However, this problem (ECMM) is NP hard, and we approximate the solution via a practically feasible greedy approach. Notably, we empirically demonstrate the effectiveness of this greedy based top-H approach to be superior to the existing SoTA. Additionally, under the constrained distribution of the Zipfian regime, the greedy prefix is constructed to generally maximize mass under minimal entropy growth. Because entropy increases monotonically with each added token, and the most probable tokens usually contribute less to entropy per unit mass, the greedy approach reaches the entropy threshold more efficiently than any alternative.

B LLM-as-a-Judge Evaluation Prompts

Following (Nguyen et al., 2024), we adopt the following judge evaluation prompt and three open-ended prompts designed to elicit creative responses and facilitate creativity–coherence trade-off analysis:

Judge Evaluation Prompt

You are an expert judge evaluating AI-generated creative writing. I am testing the diversity and coherent writing capabilities of three different models. I will paste three different responses that were generated here. Rate responses based on the following metrics:
1. Diversity: Novelty and uniqueness of ideas **2. Originality:** Innovative approach to the prompt **3. Narrative Flow:** Coherence of the text **4. Emotional Impact:** Ability to evoke feelings **5. Imagery:** Vividness of descriptions.
Rate each metric from 1 to 10. Also, suggest the overall winner: the response that best maintains high coherence while demonstrating high diversity.

Prompt 1

Write a story about an alien civilization's first contact with Earth from their perspective.

Prompt 2

Write a story about a world where time suddenly starts moving backwards.

Prompt 3

Write a story about a mysterious door that appears in an unexpected place.

C More Results

In this section we provide more experimental evaluations and analysis on top-H sampling.

C.1 LLM-as-a-Judge for creativity and coherence evaluation

Table 5 presents creativity and coherence evaluations under the LLM-as-a-Judge setup for the Qwen2.5–3B and Phi-3-Mini–4k–Instruct models. The observed trends are consistent with those of LLaMA3.1–8B–Instruct: as the decoding temperature increases, coherence degrades notably for both min-p and top-p sampling methods. In contrast, top-H effectively maintains coherence while producing more creative outputs.

C.2 Validation with large model

We replicated the MT-Bench (Table 6) and GPQA (Table 7) experiments using **LLaMA3.3-70B-Instruct** with the setup of Section 6.1 in the paper. Specifically, top-H outperforms min-p by up to 6.0% and 6.47% on MT-Bench and GPQA, respectively.

LLM	Temperature	Prompt	Sampling	M1	M2	M3	M4	M5
Phi-3-Mini	1.0	Prompt 1	Top-p	7.4	7.4	**8.0**	8.0	7.2
			Min-p	6.6	6.5	7.8	6.7	7.2
			Top-H	**8.4**	**8.4**	**8.0**	**8.1**	**8.6**
		Prompt 2	Top-p	7.2	7.0	**8.4**	6.6	7.4
			Min-p	6.8	7.0	7.6	7.2	6.4
			Top-H	**7.4**	**7.4**	8.0	**7.8**	**8.4**
		Prompt 3	Top-p	7.0	6.4	**8.0**	7.6	8.0
			Min-p	6.4	5.5	**8.0**	6.2	7.2
			Top-H	**8.1**	**9.0**	**8.0**	**8.0**	**9.0**
	1.5	Prompt 1	Top-p	8.0	7.5	**9.0**	7.6	8.4
			Min-p	7.9	7.8	8.4	7.5	8.4
			Top-H	**8.5**	**8.3**	8.8	**8.4**	**9.2**
		Prompt 2	Top-p	7.2	6.8	7.2	6.6	7.6
			Min-p	7.4	7.4	**8.2**	**7.8**	**8.0**
			Top-H	**8.2**	**8.0**	7.4	7.6	7.8
		Prompt 3	Top-p	7.4	7.1	**8.4**	7.1	7.8
			Min-p	6.9	6.6	7.7	7.2	7.2
			Top-H	**8.2**	**8.0**	8.0	**8.0**	**8.3**
	2.0	Prompt 1	Top-p	7.2	7.0	5.6	6.0	7.0
			Min-p	**7.6**	**7.8**	**8.6**	**8.6**	7.8
			Top-H	**7.6**	**7.8**	**8.6**	8.0	**8.6**
		Prompt 2	Top-p	**8.6**	**8.8**	5.5	7.0	**8.7**
			Min-p	7.4	7.5	**8.8**	7.8	7.8
			Top-H	**8.6**	8.2	8.4	**8.3**	8.3
		Prompt 3	Top-p	**7.4**	7.6	5.0	5.8	7.4
			Min-p	6.6	6.6	8.0	7.0	7.4
			Top-H	**7.4**	**7.8**	**8.4**	**7.6**	**8.2**
Qwen2.5	1.0	Prompt 1	Top-p	5.0	4.8	**7.2**	5.4	5.2
			Min-p	4.8	4.0	6.2	4.2	4.2
			Top-H	**8.2**	**7.8**	**7.2**	**7.6**	**7.8**
		Prompt 2	Top-p	7.4	6.8	**8.0**	6.8	7.1
			Min-p	6.4	6.4	7.0	6.2	6.4
			Top-H	**7.6**	**7.6**	7.2	**7.6**	**7.8**
		Prompt 3	Top-p	6.0	5.3	**8.4**	6.4	6.8
			Min-p	6.4	6.0	7.0	5.8	6.6
			Top-H	**7.9**	**7.6**	8.0	**7.3**	**8.4**
	1.5	Prompt 1	Top-p	6.0	5.3	7.8	5.6	5.6
			Min-p	6.1	5.4	6.0	5.1	4.9
			Top-H	**7.9**	**8.0**	**8.1**	**7.3**	**7.8**
		Prompt 2	Top-p	7.2	6.8	7.6	6.8	7.0
			Min-p	6.8	6.2	7.4	6.8	6.8
			Top-H	**8.2**	**8.2**	**8.0**	**8.0**	**8.6**
		Prompt 3	Top-p	7.2	6.9	7.3	6.2	6.7
			Min-p	6.7	6.5	7.2	7.1	7.0
			Top-H	**7.6**	**7.5**	**7.6**	**7.2**	**8.1**
	2.0	Prompt 1	Top-p	6.0	5.4	3.8	3.6	4.4
			Min-p	6.8	6.6	7.0	6.0	6.8
			Top-H	**7.4**	**7.4**	**8.0**	**6.8**	**7.2**
		Prompt 2	Top-p	**7.6**	**8.0**	4.6	5.8	7.0
			Min-p	6.2	6.3	7.8	5.8	5.9
			Top-H	7.5	7.8	**8.1**	**7.0**	**7.4**
		Prompt 3	Top-p	**7.4**	7.1	4.3	5.5	6.3
			Min-p	6.6	6.5	5.5	6.4	6.9
			Top-H	7.3	**7.4**	**8.4**	**8.2**	**8.3**

Table 5: Evaluation metrics and the judge scores (on a scale of 1.0 to 10.0) for different LLMs, temperatures, prompts, and sampling methods. M1-M5 denote creativity, originality, narrative flow, imagery, and vitality, respectively.

Method	Temperature = 1.0	Temperature = 1.5	Temperature = 2.0
Top-p	7.06	6.75	3.86
Min-p	7.08	7.11	6.44
Top-H	**7.08**	**7.14**	**7.04**

Table 6: MT-Bench results with LLaMA3.3-70B-Instruct.

Method	Temperature = 1.0	Temperature = 1.5	Temperature = 2.0
Top-p	43.75	41.74	34.15
Min-p	45.76	42.41	39.29
Top-H	**51.12**	**48.88**	**45.31**

Table 7: GPQA results with LLaMA3.3-70B-Instruct.

Additionally, with the large model, we produced results with the LLM-as-judge setup described in Section 6.1.3, reported in Table 8. This demonstrates top-H's consistent improvement trend over alternatives.

Temperature	Sampling Method	M1	M2	M3	M4	M5
1.0	Top-p	6.05 ± 0.24	6.10 ± 0.22	8.80 ± 0.20	7.15 ± 0.26	6.95 ± 0.22
	Min-p	7.05 ± 0.22	7.10 ± 0.22	$\mathbf{8.85} \pm 0.20$	$\mathbf{7.95} \pm 0.15$	8.00 ± 0.26
	Top-H	$\mathbf{8.10} \pm 0.26$	$\mathbf{8.85} \pm 0.22$	8.05 ± 0.22	7.60 ± 0.20	$\mathbf{8.10} \pm 0.24$
1.5	Top-p	6.95 ± 0.22	7.80 ± 0.20	$\mathbf{8.65} \pm 0.15$	8.35 ± 0.22	8.40 ± 0.20
	Min-p	7.10 ± 0.30	7.15 ± 0.22	8.05 ± 0.15	7.25 ± 0.26	7.15 ± 0.22
	Top-H	$\mathbf{8.95} \pm 0.20$	$\mathbf{8.90} \pm 0.15$	8.10 ± 0.20	$\mathbf{9.00} \pm 0.22$	$\mathbf{8.95} \pm 0.22$
2.0	Top-p	7.75 ± 0.22	8.15 ± 0.26	5.55 ± 0.22	7.60 ± 0.20	7.25 ± 0.26
	Min-p	8.15 ± 0.20	7.30 ± 0.35	6.25 ± 0.22	$\mathbf{8.05} \pm 0.24$	6.80 ± 0.22
	Top-H	$\mathbf{8.80} \pm 0.20$	$\mathbf{8.20} \pm 0.30$	$\mathbf{7.10} \pm 0.26$	8.00 ± 0.15	$\mathbf{7.85} \pm 0.20$

Table 8: LLM-as-judge results with LLaMA3.3-70B-Instruct.

C.3 Human Eval

We have conducted Human Evaluation of LLM-generated texts using a setup similar to that of min-p, and compared with top-p and min-p. We recruited 14 PhD students for this. We used LLaMA3.1-8B-Instruct with texts generated using a prompt adapted from the min-p framework: "Write me a creative story." For each configuration, we generated three outputs, capped at 512 tokens. Participants were asked to evaluate them along quality and diversity with rating on a scale of 1–10. Results are shown in Table 9.

Sampling Method	Creativity (T=0.7)	Coherence (T=0.7)	Creativity (T=1.0)	Coherence (T=1.0)	Creativity (T=2.0)	Coherence (T=2.0)
Top-p	7.57 ± 0.72	4.78 ± 0.93	6.28 ± 0.69	5.78 ± 0.79	3.57 ± 0.72	7.57 ± 0.62
Min-p	6.92 ± 0.59	5.28 ± 1.03	6.21 ± 0.77	5.92 ± 0.79	6.00 ± 0.75	6.57 ± 0.72
Top-H	7.35 ± 0.71	5.42 ± 0.62	6.57 ± 0.90	6.42 ± 0.91	6.42 ± 0.62	7.07 ± 0.70

Table 9: Human evaluation results on creativity and coherence ratings.

C.4 Top-H truncation threshold

In this section, we demonstrate how top-H addresses the limitations of min-p sampling, as illustrated in Fig. 1, which served as our motivational case study. Comparing the two distributions, we observe that distribution A exhibits greater randomness, with a higher proportion of low-probability tokens relative to distribution B. This observation is supported by their entropy values: distribution A has an entropy of 4.28, while distribution B has a lower entropy of 3.71. Consequently, an optimal

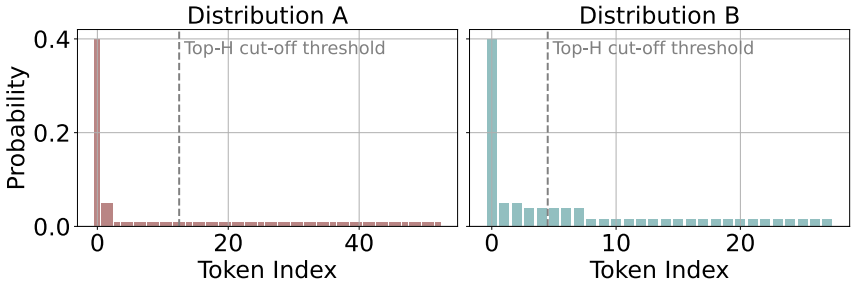

Figure 6: Probability distribution of two different types with associated top-H thresholds.

decoding strategy would be expected to allocate a larger sampling pool in scenario A, reflecting the model's lower confidence. This is precisely how the top-H decoding method operates. When applied with $\alpha = 0.4$, the resulting entropy thresholds are shown in Fig. 6. As illustrated, top-H assigns a significantly larger token set to distribution A to accommodate its higher uncertainty—an adjustment that min-p fails to make. Moreover, in scenario B, top-H retains several high-probability tokens that min-p erroneously excludes. Therefore, top-H effectively addresses both key shortcomings of min-p sampling in such settings.

C.5 Impact of the α Parameter

Table 10 reports GPQA accuracy and the average sampling pool size across different α values. The experiment is done using the `LLaMA3.1-8B-Instruct` model with temperature $T = 1.5$. These results show that:

1. Larger α values slightly reduce accuracy, which aligns with the nature of GPQA's graduate-level questions that benefit from more confident (less diverse) answers.
2. Sampling pool size increases with α, providing more generative options and supporting the creativity aspect observed in Figure 4.

α	0.10	0.15	0.20	0.25	0.30	0.35	0.40	0.45	0.50	0.55	0.60	0.65	0.70	0.75	0.80	0.85	0.90
Accuracy on GPQA	0.3085	0.3085	0.3095	0.3085	0.3125	0.3103	0.3058	0.3050	0.2869	0.2937	0.2879	0.2790	0.2655	0.2879	0.2612	0.2656	0.2701
Average sampling pool size	1.01	1.03	1.20	1.48	1.90	2.53	3.48	4.74	6.94	9.11	11.79	15.77	21.35	27.99	36.28	47.06	59.28

Table 10: GPQA accuracy and average sampling pool size across different α

C.6 Comparison of top-H to Mirostat method

In addition to the results obtained with η-sampling, we include further comparisons with Mirostat (Basu et al., 2020) Table 11 for MTBench and Table 12 for GPQA, serving as an additional entropy-aware baseline. Unless otherwise specified, all decoding and evaluation configurations follow those in the paper. For Mirostat, we set the target entropy parameter to $\tau = 3$.

Temperature	**LLaMA3.1-8B-Instruct**		**Phi-3-Mini**		**Qwen2.5 3B**	
	Top-H	**Mirostat**	**Top-H**	**Mirostat**	**Top-H**	**Mirostat**
1.0	**6.788**	6.375	**6.819**	6.600	**5.956**	5.369
1.5	**6.819**	5.594	**6.556**	5.500	**5.513**	4.469
2.0	**6.438**	5.519	**6.056**	5.269	**4.519**	4.256

Table 11: MTBench results comparing Top-H and Mirostat. Top-H wins in all 9 settings. Averaged over all models and temperatures, Top-H achieves 6.163 vs. Mirostat 5.439 (+0.724 absolute, +13.3%).

D Limitations

In this paper, we introduced a novel sampling method—top-H—as a greedy solution to the NP-hard *entropy-constrained mass maximization* (ECMM) problem. While top-H demonstrates strong

Temperature	LLaMA3.1-8B-Instruct		Phi-3-Mini		Qwen2.5 3B	
	Top-H	Mirostat	Top-H	Mirostat	Top-H	Mirostat
1.0	29.24	**30.36**	**32.37**	30.13	**28.79**	28.35
1.5	**30.58**	25.67	**30.80**	29.02	**27.90**	26.34
2.0	**28.79**	28.79	**30.80**	29.91	**28.12**	27.79

Table 12: GPQA results comparing Top-H and Mirostat. Top-H outperforms Mirostat in 7 of 9 settings, with one Mirostat win at LLaMA-8B (T = 1.0). Averaged over all models and temperatures, Top-H achieves 29.71 vs. Mirostat 28.48 (+1.23 absolute, +4.3%).

empirical performance, it does not provide general competitive guarantees that apply across a broad range of distributions. Moreover, the hyperparameter α was tuned manually, even though the method exhibits robustness to its variation. Designing an algorithm that offers a provable approximation ratio and can dynamically adapt the entropy threshold α remains an important direction for future work.

E Broader Impact

Top-H sampling enhances the coherence and creativity of text generated by large language models, especially at high temperatures. This can positively impact applications such as creative writing, education, and human-AI interaction by making outputs more diverse and engaging. Its efficiency and ease of integration also support broader accessibility in open-source settings. However, the same improvements in fluency could be misused to generate more persuasive disinformation or evade content moderation. While top-H is a general-purpose sampling method, we recommend pairing it with safety mechanisms and monitoring in sensitive deployments. Open-sourcing our implementation and providing clear usage guidelines will support responsible adoption and further research.