# Probabilities Are All You Need: A Probability-Only Approach to Uncertainty Estimation in Large Language Models

**Manh Nguyen, Sunil Gupta, Hung Le**
Applied Artificial Intelligence Initiative
Deakin University
Geelong, Victoria, Australia
{manh.nguyen, sunil.gupta, thai.le}@deakin.edu.au

**Abstract**

Large Language Models (LLMs) exhibit strong performance across various natural language processing (NLP) tasks but remain vulnerable to hallucinations, generating factually incorrect or misleading outputs. Uncertainty estimation, often using predictive entropy estimation, is key to addressing this issue. However, existing methods often require multiple samples or extra computation to assess semantic entropy. This paper proposes an efficient, training-free uncertainty estimation method that approximates predictive entropy using the responses' top-$K$ probabilities. Moreover, we employ an adaptive mechanism to determine $K$ to enhance flexibility and filter out low-confidence probabilities. Experimental results on three free-form question-answering datasets across several LLMs demonstrate that our method outperforms expensive state-of-the-art baselines, contributing to the broader goal of enhancing LLM trustworthiness.

**Code** — https://github.com/manhitv/PRO

## 1 Introduction

Large language models (LLMs) (Achiam et al. 2023; Touvron et al. 2023; Anil et al. 2023) have demonstrated remarkable capabilities in solving NLP tasks. However, they still suffer from hallucinations (Xiao and Wang 2021; Malinin and Gales 2021), where LLMs generate outputs that are factually incorrect or misaligned with the provided context, even when given valid inputs. Therefore, detecting and mitigating hallucinations remains a significant challenge in the development of trustworthy AI systems.

Estimating the uncertainty of model generations is a promising direction to improve the reliability of LLMs (Manakul, Liusie, and Gales 2023). By quantifying the confidence of their outputs, uncertainty measures help identify cases where a model is likely to produce incorrect or misleading information. Existing uncertainty estimation has been widely explored and can be categorized into two main types: training-based and training-free methods. Training-based methods involve training a custom model or leveraging a pretrained model to assess the truthfulness of the output (Azaria and Mitchell 2023; Kuhn, Gal, and Farquhar 2023; Farquhar et al. 2024; Nikitin et al. 2024; Qiu and Miikkulainen 2024). These methods often require access to low-level features of the LLMs, which may not always be available in practice. In contrast, training-free methods are simpler, as they rely only on token probabilities from the original LLM to estimate predictive entropy and log probability without requiring additional model training (Huang et al. 2025). In this work, we focus on training-free approaches due to their simplicity and practical usability.

Early training-free approaches use log-probability to measure the model's uncertainty. Manakul, Liusie, and Gales (2023) propose several metrics based on token log-probabilities, including taking the average and max over them as baselines for uncertainty estimation. More advanced approaches use predictive entropy (Lindley 1956) to characterize the total uncertainty in a model's output distribution. Predictive entropy is reliable because it captures the overall uncertainty in a model's output, considering the entire probability distribution rather than just a single prediction. This measure has demonstrated robustness across diverse tasks, such as data-to-text generation (Xiao and Wang 2021), neural machine translation (Malinin and Gales 2021), and abstractive summarization (Van der Poel, Cotterell, and Meister 2022). More recently, predictive entropy has been extended to question-answering tasks in the context of LLMs (Kuhn, Gal, and Farquhar 2023; Lin, Trivedi, and Sun 2023; Duan et al. 2024). However, it is impossible to compute predictive entropy exactly due to the high sampling cost associated with generating all possible outputs. Several studies have attempted to refine entropy-based uncertainty estimation by incorporating token- or sentence-level weighting. For example, Duan et al. (2024) introduce token and sentence weights based on their importance within specific spans of text, such as a set of tokens within a sentence or a group of responses, and combine these weights into an improved version of predictive entropy. Kuhn, Gal, and Farquhar (2023) and Farquhar et al. (2024) propose semantic entropy (SE), which quantifies uncertainty by computing the predictive entropy of semantically related response clusters. This approach has opened new possibilities for uncertainty estimation by adjusting for semantic similarities. While these semantic methods have shown strong empirical performance, due to the inherent complexity of semantic meaning, accurately clustering responses remains challenging, particularly since a single sentence can belong to

multiple clusters. Additionally, these methods frequently depend on external models, such as natural language inference (NLI) models (He et al. 2020), to assess semantic distances between responses, yet the relationships between clusters in terms of semantic similarity remain unclear.

Recently, Aichberger, Schweighofer, and Hochreiter (2024) introduce G-NLL, which estimates uncertainty using only the probability of the most likely generation (i.e., the most likely complete sentence). This suggests that reliable estimation is possible without modeling response semantics. However, we argue that relying solely on the most likely generation fails to accurately approximate the uncertainty, resulting in suboptimal performance in certain cases. In this work, we propose a novel training-free uncertainty estimation method, dubbed **PRO** (**PR**obability **O**nly), that is based entirely on the top $K$ generations with the highest probabilities. Unlike existing semantics-based methods, our approach does not require additional computations involving semantic meaning or response embeddings, making it both simple and cost-effective. We formulate uncertainty estimation as a predictive entropy (PE) approximation problem and theoretically demonstrate that PE can be estimated using the highest probabilities from sampled generations. Each generation's probability is computed using the Negative Log-Likelihood (NLL). The resulting uncertainty is further refined by applying a probability threshold, which filters out low-probability generations to ensure that only high-confidence responses contribute to the uncertainty measure, ultimately yielding the final top-$K$ probabilities. We hypothesize that while a larger $K$ may improve entropy estimation, incorporating low-probability responses adds noise and contributes negligibly to the final entropy value. In summary, our main contributions are as follows:

- We propose a simple, theoretically motivated entropy approximation for uncertainty estimation that uses only the top probabilities from sampled generations.
- We introduce a probability threshold as a hyperparameter to adaptively quantify uncertainty, enhancing both flexibility and empirical performance.
- We conduct extensive experiments to show that our method outperforms existing baselines for uncertainty estimation across various free-form question-answering tasks and LLMs.

## 2 Related Work

In LLM's uncertainty estimation literature, one common direction is to focus on analyzing the internal states of the LLMs and training external models to predict confidence and uncertainty. For example, Azaria and Mitchell (2023) train a model to assess the truthfulness of the output based on specific hidden layers. Chuang et al. (2024) instead rely on attention layers to compute a lookback ratio, which measures the model's focus on the provided context, arguing that it correlates positively with the correctness of the response. Additionally, Chen et al. (2024) estimate uncertainty by analyzing eigenvalues of the response covariance matrix, capturing the semantic consistency in the dense embedding space. In contrast, our method relies solely on token probabilities rather than internal states, making it significantly simpler and more efficient. This eliminates the need for additional model training or complex computations on hidden layers, reducing computational costs and enabling broader applicability across different models.

Another prevalent approach is using logits or probability scores to estimate uncertainty. Predictive entropy is widely used for this purpose (Malinin and Gales 2021) as it quantifies the total uncertainty in a model's output distribution. Several studies have explored enhancing predictive entropy by incorporating weighting functions. Duan et al. (2024) and Bakman et al. (2024) utilize a transformer-based model to compute weights at token, phrase, and sentence levels, while Zhang et al. (2023) leverage named entity recognition (NER) models to detect and assign greater weights to key terms. These approaches are simple yet only show moderate performance in practice. In recent years, semantic-based uncertainty estimation has gained significant attention. Kuhn, Gal, and Farquhar (2023) and Farquhar et al. (2024) propose semantic entropy (SE), which quantifies uncertainty by considering semantic relationships between generated responses. Inspired by SE, Lin, Trivedi, and Sun (2023) propose uncertainty metrics by extracting semantic matrices, which are constructed by analyzing the relationships among generated responses. Semantic entropy has been further extended in several ways. For example, Nikitin et al. (2024) apply kernel functions applied to a semantic graph, resulting in a generalized version of SE. Qiu and Miikkulainen (2024) advance this approach by introducing semantic density (SD), which is computed from a generalized semantic space and directly measures uncertainty using kernel functions without the need for clustering. To the best of our knowledge, this method represents the state-of-the-art technique in the field of uncertainty estimation. Although promising, these methods typically incur high computational costs and depend on pretrained models to assess semantic distances. In contrast, our method introduces minimal computational cost by relying solely on the token probability provided by the language models, without the need for additional inference or semantic similarity calculations.

Recently, some studies have also used the likelihood of a single output sequence as a simple yet effective uncertainty estimation (Aichberger, Schweighofer, and Hochreiter 2024; Vazhentsev et al. 2024). Our work also aims to quantify uncertainty through the most probable generations. However, unlike prior approaches that rely on a single or a fixed number of responses, we introduce a thresholding mechanism for selecting top-$K$ generations adaptively. More importantly, we derive a general formulation for entropy approximation, offering a more flexible and theoretically grounded measure of uncertainty.

## 3 Preliminaries

**Problem Statement** Information theory (Lindley 1956) provides a systematic approach to measure uncertainty by defining it as the entropy of the output distribution:

$$PE(x) = H(Y|x) = -\int p(y|x)\log p(y|x)dy \quad (1)$$

where $Y$ is the output random variable, $x$ is the input, and $H(Y|x)$ is a conditional entropy which represents average uncertainty about $Y$ when $x$ is given. A low predictive entropy indicates a heavily concentrated output distribution, whereas a high one indicates that many possible outputs are similarly likely.

In the context of LLMs, we can measure the uncertainty of a generation as:

$$U(x) = H(Y|x) = -\sum_{y} p(y|x)\log p(y|x) \quad (2)$$

where, in practice, $y \in \{y_1, y_2, ..., y_N\}$ is the finite set of generated sequences (i.e., the samples of $Y$) given prompt $x$ using a specific LLM.

**Generation Probability** The probability of generating sequence $y$ given a prompt $x$ is typically factorized as the product of conditional probabilities over its individual tokens:

$$p(y|x) = \prod_{t=1}^{T} p(y^t|y^{<t}, x) \quad (3)$$

where $T$ is the length of the generated sequence, and $y^t$ is the token at position $t$. Taking the logarithm, we get:

$$\log p(y|x) = \sum_{t=1}^{T} \log p(y^t|y^{<t}, x). \quad (4)$$

This quantity can be computed empirically using the **Negative Log-Likelihood (NLL)** (Aichberger, Schweighofer, and Hochreiter 2024), which is commonly used as a loss function in training LLMs:

$$\text{NLL}(y|x) = -\sum_{t=1}^{T} \log p(y^t|y^{<t}, x). \quad (5)$$

The token-level probabilities $p(y^t|y^{<t}, x)$ are typically derived from the model's output logits using the softmax function, making this approach a standard method for evaluating the likelihood of generated text in LLMs.

# 4 Methodology

## 4.1 Uncertainty Approximation

Calculating the uncertainty directly from Eq. (2) is computationally expensive, as generating all possible responses $Y$ for an accurate estimate would require significant resources. A typical way to estimate the predictive entropy is sampling $N$ generations given prompt $x$. Based on these samples, we propose selecting top-$K$ ($K < N$) generations $(y_1^*, y_2^*, ..., y_K^*)$ with the highest probabilities to approximate uncertainty, i.e.:

$$p(y_1^*|x) \geq p(y_2^*|x) \geq \cdots \geq p(y_K^*|x), \quad (6)$$

$$p(y_K^*|x) \geq p(y_i|x), \forall i \in \{K+1, \ldots, N\}. \quad (7)$$

For simplicity, we use $p_i^*$ to represent the probability of the top $i$-th generation $p(y_i^*|x)$. Given these notations, we introduce an approximation of PE as a **PR**obability-**O**nly uncertainty score (PRO):

$$PRO(x) = -\log p_K^* - \sum_{i=1}^{K} p_i^* \log \frac{p_i^*}{p_K^*} \quad (8)$$

Here, using Eq. (4) and the standard NLL from Eq. (5), we can write the probability of a particular generation as follows:

$$p(y|x) = e^{-\text{NLL}(y|x)} \quad (9)$$

We theoretically establish that PRO serves as a lower bound for predictive entropy, as formalized in the following proposition:

**Proposition 1.** *Let $\mathbf{y}^* = (y_1^*, y_2^*, \ldots, y_K^*)$ be the top $K$ generations of a LLM given prompt $x$. The predictive entropy approximation using the top $K$ probabilities satisfies the following inequality:*

$$H(Y|x) \geq -\log p_K^* - \sum_{i=1}^{K} p_i^* \log \frac{p_i^*}{p_K^*} \quad (10)$$

*Proof.* See Appendix A.1. □

While this lower bound is computationally efficient, it still captures the essential characteristics of uncertainty. Furthermore, estimating uncertainty with a lower bound helps avoid being overly influenced by outlier responses, i.e., low-probability generations that contribute disproportionately to the entropy due to their high diversity. These rare responses may not meaningfully represent the model's general behavior but can artificially inflate the uncertainty estimate. By concentrating on the top $K$ most likely responses, our method reduces sensitivity to such noise and provides a more stable approximation. In this sense, the lower bound offers a more concentrated view of uncertainty, emphasizing the model's confidence in its most plausible outputs. Notably, when our approximation is high, it still indicates that the true predictive entropy $H(Y \mid x)$ must also be high, indicating more uncertainty in the model's output. This is important for detecting highly uncertain cases, where the model is unsure and generates diverse responses.

When $K = 1$, the right-hand side of Eq. (10) becomes the negative log-probability of the most likely generation, as in Aichberger, Schweighofer, and Hochreiter (2024). Although using the most likely generation demonstrates robustness in performance, it is not always sufficient to capture the distribution of possible responses, especially in cases of high uncertainty or ambiguous prompts. Relying only on the top-1 response may overlook other plausible responses, leading to a less accurate uncertainty estimation. Our method accounts for a broader set of likely responses, yielding a more representative and stable estimate of uncertainty. This benefit is empirically demonstrated in Section 5.3, where using larger $K$ improves uncertainty estimation and leads to better performance on downstream tasks.

### 4.2 On Top-$K$ Selection

To select $K$, one could simply set a fixed number of $K < N$. However, we argue that a fixed $K$ is suboptimal because the probability distributions of LLM generations may include low-probability responses that are less reliable for uncertainty estimation. These low-probability responses can introduce noise, reducing the effectiveness of the uncertainty measure. To address this, we introduce an adaptive constraint that filters out low-probability responses that would otherwise dilute the quality of the approximation. This constraint is essential for ensuring that the uncertainty measure reflects only the most confident responses while avoiding noise from less probable, potentially irrelevant generations. The adaptive constraint is defined as follows:

$$\mathbf{p}_K = \{p_k \mid p_k \geq \alpha, 1 \leq k \leq N\}, \tag{11}$$

where $\mathbf{p}_K$ is the selected set of top $K$ probabilities, and $\alpha$ ($0 \leq \alpha \leq p_1^* \leq 1$) is a tunable hyperparameter that truncates the output probability distribution $p_k$. Larger $\alpha$ entails more aggressive truncation, keeping only high-probability generations, whereas smaller $\alpha$ allows generations with lower probabilities to be included. When $\alpha = 0$, we use all $N$ generations, as no probability threshold is applied. On the other hand, when $\alpha = p_1^*$, only the most likely generation ($K = 1$) is selected. In practice, we observe that the probability gap between the top response and the others can be significant, particularly when the top response is much more probable than the others. By setting $\alpha$ to avoid distortion caused by low-probability responses, this constraint ensures that only high-confidence generations are included in the uncertainty calculation. This approach is crucial for improving the reliability of the results. For a deeper understanding, including qualitative examples where our method captures uncertainty more effectively than fixed-$K$ baselines, and where probability gaps influence the retained generations, please refer to Appendix A.2. The optimal value of $\alpha$ for each setting (model-dataset pair) is determined through a grid search on a small validation set comprising approximately 100 samples, with further details provided in the following section.

## 5 Experiments and Results

### 5.1 Experimental Setup

We follow the same evaluation approach as related work by focusing on free-form question-answering tasks (Kuhn, Gal, and Farquhar 2023; Farquhar et al. 2024; Qiu and Miikkulainen 2024).

**Datasets**. We perform experiments on three free-form question-answering datasets commonly used in the literature: TriviaQA (Joshi et al. 2017), SciQ (Welbl, Liu, and Gardner 2017), and Natural Questions (NQ) (Kwiatkowski et al. 2019).

**Models**. We use five distinct open-source models from different families and sizes: Gemma-2B, Gemma-7B (Mesnard et al. 2024), Llama2-13B (Touvron et al. 2023), Falcon-11B, and Falcon-40B (Almazrouei et al. 2023).

**Baselines**. We compare our method against seven existing LLM uncertainty estimation methods: semantic density (SD) (Qiu and Miikkulainen 2024), semantic entropy (SE) (Kuhn, Gal, and Farquhar 2023), degree (Deg) (Lin, Trivedi, and Sun 2023), length-normalization predictive entropy (NE) (Malinin and Gales 2021), predictive entropy (PE), average log likelihood (ALL) (Guerreiro, Voita, and Martins 2023), and negative log likelihood (NLL) (Aichberger, Schweighofer, and Hochreiter 2024). PE is computed using Eq. (2).

**Evaluation metrics**. Following previous works, we evaluate uncertainty methods by measuring their ability to predict the correctness of model responses using the Area Under the Receiver Operating Characteristic curve (AUROC). Higher AUROC values indicate better performance of the uncertainty estimator, as they reflect a stronger alignment between the estimated uncertainty scores and the actual correctness of the generated answers. Specifically, we define correctness based on the most likely generation (i.e., top-1 response $y_1^*$), in line with prior studies on uncertainty estimation in LLMs (Kuhn, Gal, and Farquhar 2023; Malinin and Gales 2021; Qiu and Miikkulainen 2024).

We note that some works distinguish between *uncertainty estimation* over the input prompt and *confidence scoring* for individual responses (Lin, Trivedi, and Sun 2023). In our setting, we follow recent benchmarks (Kuhn, Gal, and Farquhar 2023; Guerreiro, Voita, and Martins 2023; Qiu and Miikkulainen 2024) that evaluate uncertainty based on how well it predicts the correctness of the model's primary output. For consistency, we adopt the term *uncertainty estimation*, as used in prior work.

To determine whether a generated answer is correct, we follow standard practice by computing the F1 score of the ROUGE-L metric (Lin 2004) between the generation and the ground-truth reference. A generation is labeled as correct if this score exceeds 0.3, as in prior works. We acknowledge that automatic scoring functions such as ROUGE-L may not fully reflect human judgment. To validate the robustness of our evaluation, we provide additional results in Section 5.3, showing how performance varies across different ROUGE-L thresholds. This helps assess the sensitivity of our method and other baselines to the correctness labeling criteria.

**Sampling**. We follow the same sampling process used in baseline method (Qiu and Miikkulainen 2024). We use diverse beam search sampling with temperature $\tau = 1$ to generate the $N = 10$ responses for each question. We utilize the responses to calculate SD, SE, NE, and PE. For Deg baseline, we adopt multinomial sampling as used in the paper (Lin, Trivedi, and Sun 2023). ALL and NLL are computed using the probability of the most likely generation.

While our method relies solely on output probabilities and is therefore less sensitive to semantic variation, we acknowledge that the choice of decoding strategy can influence the resulting probability distribution. Exploring alternative generation methods, such as nucleus sampling or temperature-controlled sampling, may affect uncertainty estimation and is an important direction we leave for future work. We have added this point to the Limitations section for completeness.

**Hyperparameter selection**. For the other methods, we adopt the hyperparameters recommended in their respective papers. For our method, PRO, we perform a grid search for the optimal $\alpha$ for each configuration using separate valida-

| Dataset | Model | SD | SE | Deg | NE | PE | ALL | NLL | PRO (Ours) |
|---|---|---|---|---|---|---|---|---|---|
| TriviaQA | Gemma-2B | 0.799 | 0.668 | 0.746 | 0.692 | 0.624 | 0.789 | 0.806 | **0.819** |
| | Gemma-7B | 0.831 | 0.690 | 0.715 | 0.702 | 0.652 | 0.833 | 0.812 | **0.841** |
| | Llama2-13B | **0.862** | 0.682 | 0.802 | 0.551 | 0.552 | 0.624 | 0.684 | 0.802 |
| | Falcon-11B | 0.706 | 0.592 | **0.710** | 0.555 | 0.604 | 0.577 | 0.668 | 0.668 |
| | Falcon-40B | 0.700 | 0.724 | 0.722 | 0.674 | 0.623 | 0.658 | **0.765** | **0.765** |
| SciQ | Gemma-2B | 0.719 | 0.570 | 0.725 | 0.601 | 0.605 | 0.719 | 0.728 | **0.751** |
| | Gemma-7B | 0.741 | 0.622 | 0.699 | 0.658 | 0.678 | 0.765 | 0.755 | **0.787** |
| | Llama2-13B | 0.706 | 0.574 | **0.720** | 0.481 | 0.543 | 0.515 | 0.600 | 0.716 |
| | Falcon-11B | 0.724 | 0.554 | 0.771 | 0.561 | 0.603 | 0.573 | 0.797 | **0.799** |
| | Falcon-40B | 0.668 | 0.613 | 0.626 | 0.592 | 0.577 | 0.660 | **0.674** | **0.674** |
| NQ | Gemma-2B | 0.618 | 0.599 | 0.620 | 0.600 | 0.613 | 0.607 | 0.694 | **0.696** |
| | Gemma-7B | 0.670 | 0.621 | 0.691 | 0.662 | 0.566 | **0.698** | 0.683 | 0.691 |
| | Llama2-13B | 0.627 | 0.562 | 0.713 | 0.540 | 0.649 | 0.691 | 0.737 | **0.740** |
| | Falcon-11B | 0.636 | 0.591 | 0.580 | 0.515 | 0.522 | 0.512 | 0.684 | **0.685** |
| | Falcon-40B | 0.632 | 0.603 | 0.579 | 0.544 | 0.585 | 0.475 | 0.638 | **0.645** |
| **Average AUC** | | 0.709 | 0.618 | 0.695 | 0.595 | 0.600 | 0.646 | 0.715 | **0.739** |
| **Best Count** | | 1 | 0 | 2 | 0 | 0 | 1 | 2 | **11** |

Table 1: AUC performance comparison across various LLMs and datasets. The best scores for each setting are bolded, while the second-best scores are underlined. The last two rows summarize the average AUC and the total occurrences of the best scores for each setting, respectively.

tion sets, comprising 100 samples. We then apply the best-found $\alpha$ to the standard test sets. We report the test results and analyze the effect of this hyperparameter ($\alpha$) in Section 5.2 and Section 5.3, respectively, where we also list the final values of $\alpha$ used in each experiment.

**Computing Resources**. All experiments are conducted on a single GPU of H100 80GB.

## 5.2 Benchmarking Results

The results are summarized in Table 1. Our method outperforms all other baselines in 11 out of 15 cases, demonstrating its superior performance. On average, it achieves a 2.4% improvement across the three datasets. Notably, while our method performs similarly to SD (Qiu and Miikkulainen 2024) on TriviaQA, it significantly enhances performance on SciQ and NQ, with improvements of 3.3% and 6.5%, respectively. These gains highlight the robustness and scalability of our approach. Furthermore, our method, which can be considered a generalized version of NLL, consistently outperforms NLL by 0.3% on NQ and achieves a notable enhancement of 3.4% over the TriviaQA and SciQ. This demonstrates that our method matches existing approaches' performance and introduces tangible benefits, especially in scenarios where other models struggle. As illustrated in Appendix A.2, NLL may fail to detect uncertainty when the top-1 generation receives a high probability despite the presence of multiple diverse and equally plausible alternatives, leading to overconfident yet incorrect predictions.

Interestingly, the results also indicate that a larger model within the same model family does not always yield better performance in uncertainty estimation. For example, with Gemma, the smaller model (Gemma-2B) outperforms the larger one (Gemma-7B), while the reverse is true for Falcon, where Falcon-11B performs better than Falcon-40B. This suggests that model size alone does not guarantee improved uncertainty estimation, as larger models may overfit or generalize poorly on certain tasks. This trend aligns with recent findings (Wen and Zhang 2024; Lamba, Tiwari, and Gaur 2025), which note that smaller models can appear to perform better because their hallucinations are easier to detect, whereas stronger models hallucinate less but more subtly, leading to lower AUROC.

## 5.3 Model Analysis and Ablation Study

**Performance when adjusting the hyperparameter**

We report the change in performance when changing $\alpha$ on the validation set in Figure 1. While performance differences exist across models, the trends are model- and dataset-specific, highlighting the need for adaptive $\alpha$ selection for each setting. Our findings suggest that the optimal $\alpha$ varies considerably, as it is influenced not only by the model being used but also by the unique properties of each question-answer pair, including complexity, length, and contextual nuances. Note that we only tune $\alpha$ on a small validation set. The best ones are used for much bigger test sets. For example, with Gemma-7B and TriviaQA dataset, we choose $\alpha = 0.3$ (see Figure 1's first plot). The results remain robust on the test set, as shown in Table 1. The detailed $\alpha$ values for all model–dataset pairs can be found in Section 5.3 for further reference.

**Effect of using more generations**

We investigate the impact of using a fixed top-$K$ approach across models of similar sizes but from different families. Specifically, we compare their performance on TriviaQA dataset, where our method demonstrates superior performance over other variants of Eq. (8) with fixed values of

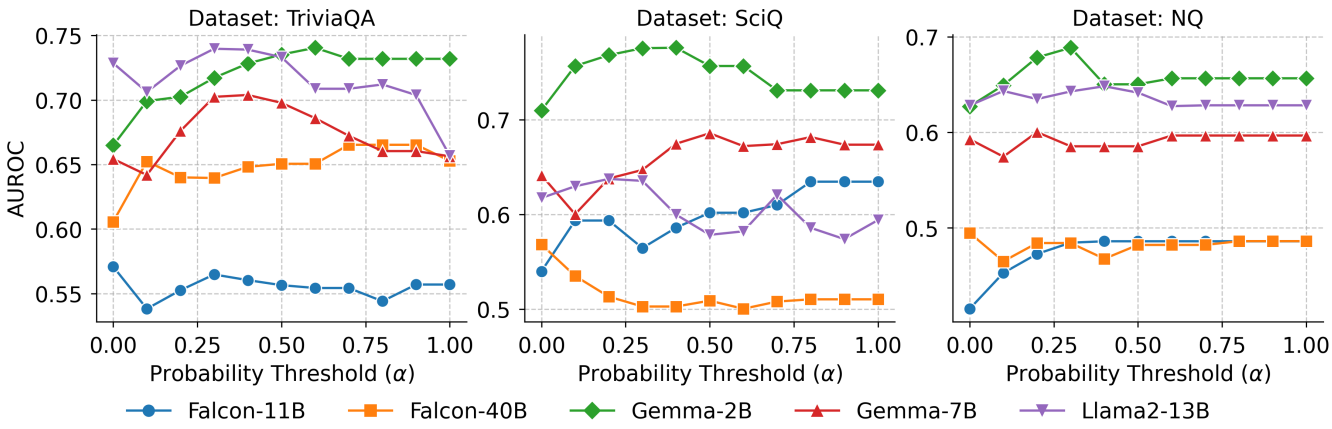


Figure 1: AUC performance when adjusting $\alpha$ for various models and datasets.

| $K$ | 1 | 2 | 3 | 4 | 5 | 6 | 7 | 8 | 9 | 10 | **PRO** |
|---|---|---|---|---|---|---|---|---|---|---|---|
| Gemma-2B | 0.806 | 0.803 | 0.795 | 0.783 | 0.780 | 0.778 | 0.781 | 0.780 | 0.786 | 0.788 | **0.819** |
| Gemma-7B | 0.812 | 0.834 | 0.831 | 0.825 | 0.810 | 0.815 | 0.826 | 0.827 | 0.819 | 0.813 | **0.841** |
| Llama2-13B | 0.684 | 0.795 | 0.801 | 0.800 | 0.797 | 0.799 | 0.802 | **0.809** | 0.803 | 0.806 | 0.802 |

Table 2: AUC performance comparison between approaches using a fixed top-$K$ ($K = 1, 2, ..., 10$) and PRO with adaptive constraint $\alpha$ on TriviaQA dataset. The best scores for each setting are bolded, while the second-best scores are underlined. $K = 1$ represents NLL baseline.

$K$, including NLL ($K = 1$). The results, summarized in Table 2, underscore the effectiveness of applying an adaptive constraint compared to a fixed $K$, reinforcing the importance of dynamically adjusting the number of considered generations. Using multiple generations (e.g., $K = 2$) enables the method to capture a broader range of plausible responses per question, leading to more stable selection and improved performance. While fixed-$K$ methods can outperform our approach in isolated cases, they generally yield lower average performance and may lack robustness when applied across different models or datasets. In contrast, our method employs an adaptive threshold (i.e., $\alpha$) to dynamically determine the number of generations considered *per question*. This adaptability allows the method to retain more diverse responses when the model exhibits uncertainty, while effectively filtering out noise when the model is confident. As illustrated in Appendix A.2, examples with large probability gaps between top-ranked generations further highlight how model confidence varies, emphasizing the benefits of an adaptive approach in handling such scenarios. Additionally, we provide recommended values of $\alpha$ selected via grid search for each model and dataset in Section 5.3. When validation data is unavailable, we suggest a conservative default of $\alpha = 0.4$, which might performs robustly across settings.

**$K$–$\alpha$ relationship**

In Figure 2, we further analyze the relationship between the number of selected generations $K$ and $\alpha$ when varying $\alpha$ across models and datasets. As $\alpha$ increases, the number of generations generally decreases. Notably, $\alpha$ decreases more rapidly than $k$, particularly in the first bucket of values ranging from 0 to 0.1. Moreover, the observed downward trend is also consistent across models within the same family, such as Falcon and Gemma. This reduction in selected generations can be attributed to a significant probability gap between the most likely generation and the others. In certain cases, using only the probability of the most likely generation ($\alpha = p_1^*$) may suffice; however, this approach generally underperforms across diverse datasets and models, lacking consistency and generalization. This is particularly evident in the TriviaQA and SciQ datasets with Falcon models, where our method produces results similar to NLL, as indicated in Table 1.

**Choice of $\alpha$**

| **Model** | **TriviaQA** | **SciQ** | **NQ** |
|---|---|---|---|
| Gemma-2B | 0.40 | 0.40 | 0.70 |
| Gemma-7B | 0.30 | 0.50 | 0.05 |
| Llama2-13B | 0.10 | 0.35 | 0.30 |
| Falcon-11B | 0.85 | 0.80 | 0.35 |
| Falcon-40B | 0.65 | 0.90 | 0.45 |

Table 3: Values of $\alpha$ used for each setting.

Table 3 shows the values of the hyperparameter $\alpha$ selected via grid search for each combination of model and dataset. These values are determined using validation sets drawn from the training data and applied during evaluation. While some larger or well-calibrated models (e.g., Llama2-

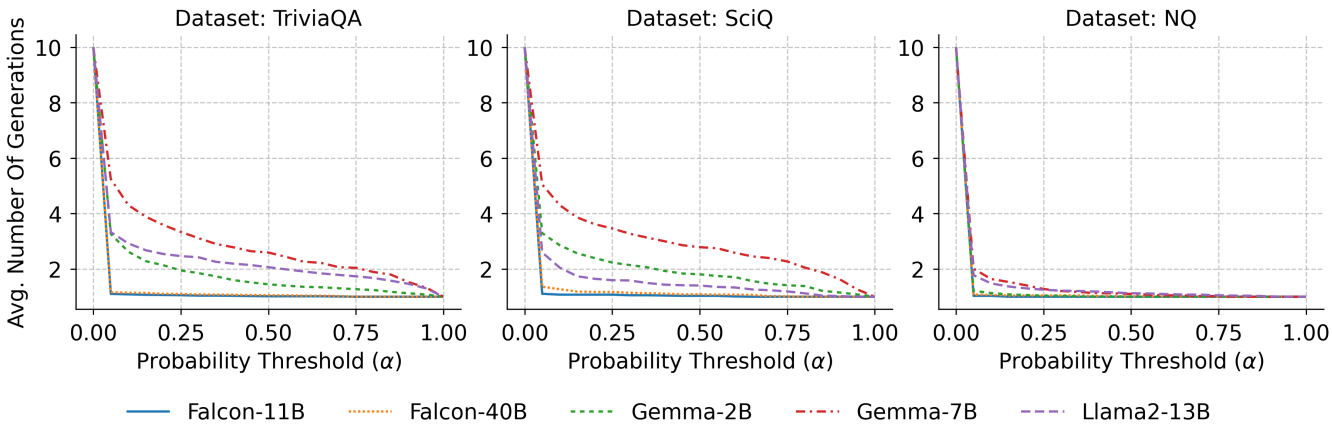


Figure 2: Relationship between number of selected generations and $\alpha$ across different models and datasets.

| **Method** | 0.1 | 0.2 | 0.3 | 0.4 | 0.5 |
|---|---|---|---|---|---|
| SD | 0.741 | 0.742 | 0.741 | 0.739 | 0.750 |
| SE | 0.628 | 0.628 | 0.622 | 0.624 | 0.636 |
| Deg | 0.721 | 0.723 | 0.699 | 0.712 | 0.716 |
| NE | 0.659 | 0.658 | 0.658 | 0.648 | 0.657 |
| PE | 0.649 | 0.659 | 0.678 | 0.691 | 0.708 |
| ALL | 0.751 | 0.755 | 0.765 | 0.784 | 0.818 |
| NLL | 0.726 | 0.736 | 0.755 | 0.792 | 0.823 |
| PRO | **0.757** | **0.763** | **0.787** | **0.802** | **0.832** |

Table 4: AUC performance comparison under varying correctness thresholds (ROUGE-L F1 from 0.1 to 0.5) for Gemma-7B on SciQ. The best scores for each threshold are bolded, and second-best scores are underlined.

13B) perform well with lower $\alpha$ values (e.g., 0.1-0.3), others such as Falcon-11B and Falcon-40B may require higher thresholds (e.g., 0.8-0.9). When validation data is unavailable, we recommend setting $\alpha$ between 0.3 and 0.5, with $\alpha = 0.4$ as a strong default.

**Effect of correctness threshold**

To understand how different correctness criteria affect evaluation, we report the Area Under Curve (AUC) performance across a range of ROUGE-L F1-score thresholds from 0.1 to 0.5 (Table 4). These thresholds define how strictly an answer must match the reference to be counted as correct. We focus on this range as it aligns with standard practice in evaluating free-form generation, where lexical variation is expected and exact matches are not strictly required.

Across all thresholds, our method (PRO) consistently achieves the highest performance, with notable gains in the 0.3–0.5 range, which is commonly used in real-world evaluation. These results underscore PRO's effectiveness under realistic correctness assumptions.

## 6 Conclusion

In this work, we studied the problem of uncertainty estimation in LLMs and introduced a generalized and robust measure of predictive uncertainty based on top $K$ generations. Our method is both simple and effective, relying exclusively on output probabilities without incorporating additional features, such as semantic similarity between responses. Our approach dynamically adjusts the number of generated samples based on probability distributions, providing greater flexibility in uncertainty estimation. These improvements strengthen the interpretability and robustness of uncertainty quantification in LLMs, contributing to more reliable and trustworthy model predictions.

## Limitations

Although our approach demonstrates strong performance in uncertainty estimation, it relies on access to token logits, which may not always be available in fully black-box LLMs. As a result, our method is currently more suitable for grey-box models where token-level probability distributions can be obtained. Additionally, our approach does not consider the meaning of the question or answer, nor their relationship in measuring uncertainty. One possible direction for further research is to integrate semantic-aware features to refine uncertainty estimation and validate the approach across diverse real-world applications. Another limitation lies in our use of beam search for candidate generation. While it ensures consistent top-mode outputs and aligns with prior work, it may introduce bias due to limited generation diversity. In future work, we plan to investigate the effect of alternative decoding strategies, such as sampling-based methods, on the quality and robustness of uncertainty estimation.

## References

Achiam, J.; Adler, S.; Agarwal, S.; Ahmad, L.; Akkaya, I.; Aleman, F. L.; Almeida, D.; Altenschmidt, J.; Altman, S.; Anadkat, S.; et al. 2023. GPT-4 technical report (2023). *arXiv preprint arXiv:2303.08774*.

Aichberger, L.; Schweighofer, K.; and Hochreiter, S. 2024. Rethinking Uncertainty Estimation in Natural Language Generation. *arXiv preprint arXiv:2412.15176*.

Almazrouei, E.; Alobeidli, H.; Alshamsi, A.; Cappelli, A.; Cojocaru, R.; Debbah, M.; Goffinet, É.; Hesslow, D.; Launay, J.; Malartic, Q.; et al. 2023. The falcon series of open language models. *arXiv preprint arXiv:2311.16867*.

Anil, R.; Borgeaud, S.; Alayrac, J.-B.; Yu, J.; Soricut, R.; Schalkwyk, J.; Dai, A. M.; Hauth, A.; Millican, K.; et al. 2023. Gemini: a family of highly capable multimodal models. *arXiv preprint arXiv:2312.11805*.

Azaria, A.; and Mitchell, T. 2023. The Internal State of an LLM Knows When It's Lying. In *The 2023 Conference on Empirical Methods in Natural Language Processing*.

Bakman, Y. F.; Yaldiz, D. N.; Buyukates, B.; Tao, C.; Dimitriadis, D.; and Avestimehr, S. 2024. Mars: Meaning-aware response scoring for uncertainty estimation in generative llms. In *Proceedings of the 62nd Annual Meeting of the Association for Computational Linguistics (Volume 1: Long Papers)*, 7752–7767.

Chen, C.; Liu, K.; Chen, Z.; Gu, Y.; Wu, Y.; Tao, M.; Fu, Z.; and Ye, J. 2024. INSIDE: LLMs' Internal States Retain the Power of Hallucination Detection. *CoRR*.

Chuang, Y.-S.; Qiu, L.; Hsieh, C.-Y.; Krishna, R.; Kim, Y.; and Glass, J. R. 2024. Lookback Lens: Detecting and Mitigating Contextual Hallucinations in Large Language Models Using Only Attention Maps. In *EMNLP*.

Duan, J.; Cheng, H.; Wang, S.; Zavalny, A.; Wang, C.; Xu, R.; Kailkhura, B.; and Xu, K. 2024. Shifting Attention to Relevance: Towards the Predictive Uncertainty Quantification of Free-Form Large Language Models. In *Proceedings of the 62nd Annual Meeting of the Association for Computational Linguistics (Volume 1: Long Papers)*, 5050–5063.

Farquhar, S.; Kossen, J.; Kuhn, L.; and Gal, Y. 2024. Detecting hallucinations in large language models using semantic entropy. *Nature*, 630(8017): 625–630.

Guerreiro, N. M.; Voita, E.; and Martins, A. F. 2023. Looking for a Needle in a Haystack: A Comprehensive Study of Hallucinations in Neural Machine Translation. In *Proceedings of the 17th Conference of the European Chapter of the Association for Computational Linguistics*, 1059–1075.

He, P.; Liu, X.; Gao, J.; and Chen, W. 2020. Deberta: Decoding-enhanced bert with disentangled attention. *arXiv preprint arXiv:2006.03654*.

Huang, L.; Yu, W.; Ma, W.; Zhong, W.; Feng, Z.; Wang, H.; Chen, Q.; Peng, W.; Feng, X.; Qin, B.; et al. 2025. A survey on hallucination in large language models: Principles, taxonomy, challenges, and open questions. *ACM Transactions on Information Systems*, 43(2): 1–55.

Joshi, M.; Choi, E.; Weld, D. S.; and Zettlemoyer, L. 2017. Triviaqa: A large scale distantly supervised challenge dataset for reading comprehension. *arXiv preprint arXiv:1705.03551*.

Kuhn, L.; Gal, Y.; and Farquhar, S. 2023. Semantic Uncertainty: Linguistic Invariances for Uncertainty Estimation in Natural Language Generation. In *The Eleventh International Conference on Learning Representations*.

Kwiatkowski, T.; Palomaki, J.; Redfield, O.; Collins, M.; Parikh, A.; Alberti, C.; Epstein, D.; Polosukhin, I.; Devlin, J.; Lee, K.; et al. 2019. Natural questions: a benchmark for question answering research. *Transactions of the Association for Computational Linguistics*, 7: 453–466.

Lamba, N.; Tiwari, S.; and Gaur, M. 2025. Investigating Symbolic Triggers of Hallucination in Gemma Models Across HaluEval and TruthfulQA. *arXiv preprint arXiv:2509.09715*.

Lin, C.-Y. 2004. Rouge: A package for automatic evaluation of summaries. In *Text summarization branches out*, 74–81.

Lin, Z.; Trivedi, S.; and Sun, J. 2023. Generating with confidence: Uncertainty quantification for black-box large language models. *arXiv preprint arXiv:2305.19187*.

Lindley, D. V. 1956. On a measure of the information provided by an experiment. *The Annals of Mathematical Statistics*, 27(4): 986–1005.

Malinin, A.; and Gales, M. 2021. Uncertainty Estimation in Autoregressive Structured Prediction. In *International Conference on Learning Representations*.

Manakul, P.; Liusie, A.; and Gales, M. 2023. SelfCheckGPT: Zero-Resource Black-Box Hallucination Detection for Generative Large Language Models. In *Proceedings of the 2023 Conference on Empirical Methods in Natural Language Processing*, 9004–9017.

Mesnard, T.; Hardin, C.; Dadashi, R.; Bhupatiraju, S.; Pathak, S.; Sifre, L.; Rivière, M.; Kale, M. S.; Love, J.; et al. 2024. Gemma: Open models based on gemini research and technology. *arXiv preprint arXiv:2403.08295*.

Nikitin, A.; Kossen, J.; Gal, Y.; and Marttinen, P. 2024. Kernel language entropy: Fine-grained uncertainty quantification for LLMs from semantic similarities. *Advances in Neural Information Processing Systems*, 37: 8901–8929.

Qiu, X.; and Miikkulainen, R. 2024. Semantic Density: Uncertainty Quantification for Large Language Models through Confidence Measurement in Semantic Space. In *The Thirty-eighth Annual Conference on Neural Information Processing Systems*.

Touvron, H.; Martin, L.; Stone, K.; Albert, P.; Almahairi, A.; Babaei, Y.; Bashlykov, N.; Batra, S.; Bhargava, P.; Bhosale, S.; et al. 2023. Llama 2: Open foundation and fine-tuned chat models. *arXiv preprint arXiv:2307.09288*.

Van der Poel, L.; Cotterell, R.; and Meister, C. 2022. Mutual information alleviates hallucinations in abstractive summarization. *arXiv preprint arXiv:2210.13210*.

Vazhentsev, A.; Fadeeva, E.; Xing, R.; Panchenko, A.; Nakov, P.; Baldwin, T.; Panov, M.; and Shelmanov, A. 2024. Unconditional Truthfulness: Learning Conditional Dependency for Uncertainty Quantification of Large Language Models. *arXiv preprint arXiv:2408.10692*.

Welbl, J.; Liu, N. F.; and Gardner, M. 2017. Crowdsourcing multiple choice science questions. *arXiv preprint arXiv:1707.06209*.

Wen, B.; and Zhang, X. 2024. Enhancing Reasoning to Adapt Large Language Models for Domain-Specific Applications. In *Annual Conference on Neural Information Processing Systems*.

Xiao, Y.; and Wang, W. Y. 2021. On Hallucination and Predictive Uncertainty in Conditional Language Generation. In *Proceedings of the 16th Conference of the European Chapter of the Association for Computational Linguistics: Main Volume*, 2734–2744.

Zhang, T.; Qiu, L.; Guo, Q.; Deng, C.; Zhang, Y.; Zhang, Z.; Zhou, C.; Wang, X.; and Fu, L. 2023. Enhancing uncertainty-based hallucination detection with stronger focus. *arXiv preprint arXiv:2311.13230*.

# A Appendix

## A.1 Proof of Proposition 1

In this section, we present a detailed proof of Proposition 1 as follows:

$$H(Y|x) = -\sum_{i=1}^{\infty} p(y_i|x) \log p(y_i|x) \tag{12}$$

$$= -\sum_{i=1}^{K} p_i^* \log p_i^* - \sum_{i=K+1}^{\infty} p_i^* \log p_i^* \tag{13}$$

$$\geq -\sum_{i=1}^{K} p_i^* \log p_i^* - \log p_K^* \sum_{i=K+1}^{\infty} p_i^* \tag{14}$$

$$= -\sum_{i=1}^{K} p_i^* \log p_i^* - \log p_K^* (1 - \sum_{i=1}^{K} p_i^*) \tag{15}$$

$$= -\log p_K^* - \sum_{i=1}^{K} p_i^* \log \frac{p_i^*}{p_K^*} \tag{16}$$

Eq. (14) follows directly from our definition of the top $K$ generations. Subsequently, Eq. (15) is derived by applying the property that the total probability sums to 1.

## A.2 Illustrative examples

This section provides illustrative examples demonstrating how our method more effectively estimates uncertainty in practice. These cases highlight instances where a fixed number of generations is insufficient, whereas our approach better captures confidence variations and distinguishes between likely and unlikely outcomes. Additional quantitative results supporting these observations are provided in Table 5.

## A.3 Model And Data Appendix

We list the links to the LLM models and datasets in Table 6.

| Q&A | Output Sequence | $p(y_i^*\|x)$ |
|---|---|---|
| **Question** | **Who is the sixth president of the United States?** | - |
| *Ground Truth* | *John Quincy Adams* | - |
| $y_1^*$ | John Quincy Adams (**correct**) | 0.455 |
| $y_2^*$ | John Quincy Adams | 0.455 |
| $y_3^*$ | John Quincy Adams | 0.455 |
| $y_4^*$ | John Adams | 0.159 |
| $y_5^*$ | Andrew Jackson | 0.065 |
| $y_6^*$ | Andrew Jackson | 0.065 |
| $y_7^*$ | Andrew Jackson | 0.065 |
| $y_8^*$ | James Madison | 0.015 |
| $y_9^*$ | George Washington | 0.005 |
| $y_{10}^*$ | John Adams, President of the United States from 1797 to 1801 | $1e^{-6}$ |
| $K = 1$ | - | 0.788 |
| $K = 2$ | - | 0.788 |
| $K = 3$ | - | 0.788 |
| **PRO** ($\alpha = 0.1$) | - | **0.404** |
| **Question** | **Who plays whitey bulger's girlfriend in black mass?** | - |
| *Ground Truth* | *actress Dakota Johnson* | - |
| $y_1^*$ | Sienna Miller (**wrong**) | 0.144 |
| $y_2^*$ | Sienna Miller | 0.144 |
| $y_3^*$ | Juno Temple | 0.118 |
| $y_4^*$ | Juno Temple | 0.118 |
| $y_5^*$ | Dakota Johnson | 0.103 |
| $y_6^*$ | Dakota Johnson | 0.103 |
| $y_7^*$ | Noomi Rapace | 0.028 |
| $y_8^*$ | Juliette Lewis | 0.020 |
| $y_9^*$ | Julianne Nicholson | 0.019 |
| $y_{10}^*$ | Winona Ryder, 1990, 1991, 1992, 1993, 1994 | $1e^{-10}$ |
| K=1 | - | 1.935 |
| K=2 | - | 1.935 |
| K=3 | - | 2.081 |
| **PRO** ($\alpha = 0.1$) | - | **2.142** |
| **Question** | **Which country has the most coastline in the world?** | - |
| *Ground Truth* | *Canada* | - |
| $y_1^*$ | Russia (**wrong**) | 0.147 |
| $y_2^*$ | Canada | 0.136 |
| $y_3^*$ | Canada | 0.136 |
| $y_4^*$ | Indonesia | 0.114 |
| $y_5^*$ | Indonesia | 0.114 |
| $y_6^*$ | Norway | 0.056 |
| $y_7^*$ | Brazil | 0.044 |
| $y_8^*$ | United States of America | 0.007 |
| $y_9^*$ | Russia, 17, | $2e^{-5}$ |
| $y_{10}^*$ | Australia, with 36,737 km of coastline | $2e^{-8}$ |
| $K = 1$ | - | 1.921 |
| $K = 2$ | - | 1.987 |
| $K = 3$ | - | 1.987 |
| **PRO** ($\alpha = 0.1$) | - | **2.087** |

Table 5: Examples of output sequences for NQ questions using Gemma-7B. The bolded scores indicate that our method better captures uncertainty estimation, whereas using a fixed number of generations is insufficient.

| **Models/Datasets** | **URL** |
|---|---|
| Gemma-2B | https://huggingface.co/google/gemma-2b |
| Gemma-7B | https://huggingface.co/google/gemma-7b |
| Llama2-13B | https://huggingface.co/meta-llama/Llama-2-13b-chat-hf |
| Falcon-11B | https://huggingface.co/tiiuae/falcon-11B |
| Falcon-40B | https://huggingface.co/tiiuae/falcon-40B |
| TriviaQA | https://huggingface.co/datasets/mandarjoshi/trivia_qa |
| SciQ | https://github.com/launchnlp/LitCab/tree/main/sciq |
| NQ | https://github.com/launchnlp/LitCab/tree/main/NQ |

Table 6: Models and Datasets Details.