Rethinking Uncertainty Estimation in Natural Language Generation

Lukas Aichberger *1 **Kajetan Schweighofer** *1 **Sepp Hochreiter** 1,2

[1] ELLIS Unit Linz and LIT AI Lab, Institute for Machine Learning,
Johannes Kepler University Linz, Austria
[2] NXAI GmbH, Linz, Austria
* Joint first authors
{aichberger, schweighofer, hochreit}@ml.jku.at

Abstract

Large Language Models (LLMs) are increasingly employed in real-world applications, driving the need to evaluate the trustworthiness of their generated text. To this end, reliable uncertainty estimation is essential. Leading uncertainty estimation methods generate and analyze multiple output sequences, which is computationally expensive and impractical at scale. In this work, we inspect the theoretical foundations of these methods and explore new directions to enhance computational efficiency. Building on the framework of proper scoring rules, we find that the negative log-likelihood of the most likely output sequence constitutes a theoretically grounded uncertainty measure. To approximate this alternative measure, we propose `G-NLL`, obtained using a single output sequence from greedy decoding. This approach streamlines uncertainty estimation while preserving theoretical rigor. Empirical results demonstrate that `G-NLL` achieves state-of-the-art performance across various LLMs and tasks. Our work lays the foundation for efficient and reliable uncertainty estimation in natural language generation, challenging the necessity of the prevalent methods that are more complex and resource-intensive.

1 Introduction

Despite advances in natural language generation (NLG), determining the trustworthiness of generated text remains challenging. Addressing this requires reliably estimating the uncertainty a language model has regarding its generated text. Although a low level of uncertainty does not guarantee factual correctness, particularly when the generated text is based on consistent but inaccurate training data, uncertainty estimates remain a reliable indicator of errors at present (Farquhar et al., 2024).

Assessing predictive uncertainty in language models is challenging due to their stochastic, autoregressive nature. Each token is selected probabilistically, leading to diverse outputs for the same input. Furthermore, the vast space of possible sequences is computationally intractable. Common uncertainty estimation methods thus rely on expectations over output distributions, such as sequence entropy (Malinin & Gales, 2021; Kuhn et al., 2023; Duan et al., 2024; Farquhar et al., 2024), which in turn requires sampling multiple output sequences. However, this is computationally costly due to the large number of model parameters. As a result, only a small subset of outputs is sampled in practice. However, differences between sampled sequences do not always indicate uncertainty, as they may vary lexically while remaining semantically similar. Some methods use inference models to assess semantics (Kuhn et al., 2023; Aichberger et al., 2024; Farquhar et al., 2024), improving uncertainty estimates but adding complexity and additional computation. These challenges make large-scale uncertainty estimation impractical for real-world applications.

Efficient uncertainty estimation methods are needed to ensure the trustworthiness of the language model's answer without imposing excessive computational demands. To address this need, we introduce `G-NLL`, an uncertainty measure computed with a single output sequence. We theoretically motivate our measure by building on insights from the framework of proper scoring rules (Gneiting & Raftery, 2007) that has recently been investigated for uncertainty estimation in the standard classification setting (Kotelevskii & Panov, 2024; Hofman et al., 2024). Specifically, we extend proper scoring rules for uncertainty estimation to NLG and explore the zero-one score as an alternative to the prevalent logarithmic score. The resulting uncertainty measure is straightforward: it is the negative log-likelihood of the most likely output sequence, which we approximate using greedy decoding to obtain G-NLL. By eliminating the need to generate and analyze multiple output sequences, our measure significantly reduces computational costs and algorithmic complexity.

Noteworthy, some recent works have considered the sequence likelihood for uncertainty estimation in NLG (Fadeeva et al., 2023; Bakman et al., 2024; Yaldiz et al., 2024; Fadeeva et al., 2024; Vazhentsev et al., 2024; Plaut et al., 2024; Abbasi-Yadkori et al., 2024), as detailed in Apx. A. However, these works introduce the approach as a heuristic without providing theoretical justification, and thus often utilize arbitrary output sequences rather than focusing explicitly on obtaining the most likely one. Furthermore, many prominent works on uncertainty estimation in NLG completely overlook using the sequence likelihood as a baseline for comparison (Kuhn et al., 2023; Duan et al., 2024; Manakul et al., 2023; Farquhar et al., 2024). To close this gap, our work derives the maximum sequence likelihood as proper uncertainty measure and proposes the efficient approximation G-NLL.

Our experiments on question answering tasks demonstrate that G-NLL matches and even exceeds the performance of current state-of-the-art uncertainty estimation measures across various model classes, model sizes, training stages, tasks, datasets, and evaluation metrics. While maintaining theoretical rigor, our measure offers an effective and scalable approach to uncertainty estimation in NLG. Therefore, G-NLL serves not only as a strong baseline for future methods, but also as a highly practical solution for widespread adoption in real-world applications. Our main contributions are:

- We derive the negative log-likelihood of the most likely output sequence as an alternative uncertainty measure in NLG and introduce G-NLL, an efficient approximation using greedy decoding.
- We provide a rigorous theoretical foundation for this alternative measure, based on established principles in uncertainty estimation theory and proper scoring rules.
- We conduct extensive experiments showing that G-NLL is both efficient and reliable, matching or outperforming state-of-the-art methods while significantly reducing computational costs.

2 Predictive Uncertainty in NLG

To introduce predictive uncertainty in NLG, we first provide background on language models. In Sec.2.1, we discuss proper scoring rules and their link to measuring predictive uncertainty in NLG. Sec.2.2 interprets established measures within this framework. Finally, we introduce the maximum sequence probability and its approximation G-NLL in Sec. 2.3 by considering a different scoring rule.

Preliminaries. We assume a fixed training dataset $\mathcal{D} = \{\boldsymbol{s}_i\}_{i=1}^{N}$ consisting of N token sequences $\boldsymbol{s} = (s_1, ..., s_T)$ where individual tokens $s_t \in \mathcal{V}$ are from a given vocabulary $\mathcal{V}$. Each token at step t is assumed to be sampled according to the predictive distribution $p(s_t \mid \boldsymbol{s}_{<t}, \boldsymbol{w}^*)$, conditioned on the sequence of preceding tokens $\boldsymbol{s}_{<t}$ and the true (but unknown) language model parameters $\boldsymbol{w}^*$. We assume that the given model class can theoretically represent the true predictive distribution, a common and usually necessary assumption (Hüllermeier & Waegeman, 2021). The likelihood of some model parameters $\tilde{\boldsymbol{w}}$ matching $\boldsymbol{w}^*$ is given by the posterior $p(\tilde{\boldsymbol{w}} \mid \mathcal{D}) = p(\mathcal{D} \mid \tilde{\boldsymbol{w}})p(\tilde{\boldsymbol{w}})/p(\mathcal{D})$.

The input to a given language model parameterized by $\boldsymbol{w}$ is a sequence $\boldsymbol{x} = (x_1, ..., x_M)$ and the output is a sequence $\boldsymbol{y} = (y_1, ..., y_T) \in \mathcal{Y}_T$, with $x, y \in \mathcal{V}$ and $\mathcal{Y}_T$ being the set of all possible output sequences with sequence length T. The likelihood of a token $y_t \in \boldsymbol{y}$ being generated by the language model is conditioned on both the input sequence and all previously generated tokens, denoted as $p(y_t \mid \boldsymbol{x}, \boldsymbol{y}_{<t}, \boldsymbol{w})$. The likelihood of output sequences $\boldsymbol{y} \in \mathcal{Y}_T$ being generated by the language model is then the product of the individual token probabilities, denoted as $p(\boldsymbol{y} \mid \boldsymbol{x}, \boldsymbol{w}) = \prod_{t=1}^{T} p(y_t \mid \boldsymbol{x}, \boldsymbol{y}_{<t}, \boldsymbol{w})$ (Sutskever et al., 2014), while the heuristic length-normalized variant is $\bar{p}(\boldsymbol{y} \mid \boldsymbol{x}, \boldsymbol{w}) = \exp\left\{\frac{1}{T}\sum_{t=1}^{T} \log p(y_t \mid \boldsymbol{x}, \boldsymbol{y}_{<t}, \boldsymbol{w})\right\}$ (Malinin & Gales, 2021).

Calculating the likelihood that a specific output sequence $\boldsymbol{y}'$ is generated by the language model parameterized by $\boldsymbol{w}$ is straightforward. The language model directly provides the token likelihoods for a given input sequence. However, determining the full probability distribution on all possible output sequences is considerably more challenging, since the size of $\mathcal{Y}_T$ increases exponentially with the sequence length. The computational complexity of evaluating all possible sequences increases with

$\mathcal{O}(|\mathcal{V}|^T)$. Since the vocabulary sizes $|\mathcal{V}|$ of modern language models are well over a hundred thousand tokens, this distribution becomes intractable to determine, even for relatively short maximal sequence lengths T (Dubey et al., 2024).

2.1 Proper Scoring Rules and the Relation to Uncertainty Measures in NLG

We next give an introduction to proper scoring rules and discuss how they give rise to uncertainty measures. For more details, in the standard classification setting, we refer to Hofman et al. (2024); Kotelevskii & Panov (2024). Proper scoring rules are a class of functions that evaluate the quality of probabilistic predictions by assigning a numerical score based on the predictive distribution and actual observations (Gneiting & Raftery, 2007). In particular, a proper scoring rule is an extended real-valued function $\mathrm{S} : \mathcal{P} \times \mathcal{Y} \to [-\infty, \infty]$, such that $\mathrm{S}(p, \boldsymbol{y})$ is $\mathcal{P}$-quasi-integrable over a convex class of probability measures $\mathcal{P}$. A scoring rule is called proper relative to $\mathcal{P}$ if the expected score is minimized when the distribution from which the outcomes $\boldsymbol{y} \in \mathcal{Y}$ are sampled matches the evaluated distribution $p \in \mathcal{P}$, and it is called strictly proper if this minimum is unique. In the context of uncertainty estimation in NLG, the general notion of proper scoring rules assigns a numerical score reflecting the degree to which an observed output sequence $\boldsymbol{y}'$ aligns with the predictive distribution of the true model $p(\boldsymbol{y} \mid \boldsymbol{x}, \boldsymbol{w}^*)$, denoted as

$$\mathrm{S}(p(\boldsymbol{y} \mid \boldsymbol{x}, \boldsymbol{w}^*), \boldsymbol{y}') \; . \tag{1}$$

To obtain concrete uncertainty measures, we need to make two specific assumptions (Schweighofer et al., 2024). First, we have to define the predictive distribution used to sample output sequences. Following Aichberger et al. (2024), we assume that we use a single, given “off-the-shelf” language model with parameters $\boldsymbol{w}$ to sample output sequences $\boldsymbol{y}' \sim p(\boldsymbol{y}' \mid \boldsymbol{x}, \boldsymbol{w})$. This assumption is also implicitly used in other works (Kuhn et al., 2023; Fadeeva et al., 2024; Farquhar et al., 2024) and is intuitively reasonable, since our main concern is the uncertainty of the output of a specific language model. Thus, we consider the expected score for possible output sequences under the predictive distribution of the given language model, denoted as $\mathrm{E}_{p(\boldsymbol{y}' \mid \boldsymbol{x}, \boldsymbol{w})} \left[\mathrm{S}(p(\boldsymbol{y} \mid \boldsymbol{x}, \boldsymbol{w}^*), \boldsymbol{y}') \right]$. This quantifies how well the predictive distribution of the given language model aligns with the true predictive distribution, capturing predictive uncertainty. Second, we have to define how the true model is approximated. We consider a Bayesian approximation of the true model, i.e., we consider each possible language model $\tilde{\boldsymbol{w}}$ according to its posterior probability $p(\tilde{\boldsymbol{w}} \mid \mathcal{D})$ (Schweighofer et al., 2023b;a). Thus, we perform a posterior expectation over the expected score, denoted as $\mathrm{E}_{p(\tilde{\boldsymbol{w}} \mid \mathcal{D})} \left[\mathrm{E}_{p(\boldsymbol{y}' \mid \boldsymbol{x}, \boldsymbol{w})} \left[\mathrm{S}(p(\boldsymbol{y} \mid \boldsymbol{x}, \tilde{\boldsymbol{w}}), \boldsymbol{y}') \right] \right]$, which can be additively decomposed into an entropy term and a divergence term (Gneiting & Raftery, 2007; Kull & Flach, 2015):

$$\underbrace{\mathrm{E}_{p(\tilde{\boldsymbol{w}} \mid \mathcal{D})} \left[\mathrm{E}_{p(\boldsymbol{y}' \mid \boldsymbol{x}, \boldsymbol{w})} \left[\mathrm{S}(p(\boldsymbol{y} \mid \boldsymbol{x}, \tilde{\boldsymbol{w}}), \boldsymbol{y}') \right] \right]}_{\text{expected score}} = \tag{2}$$

$$\underbrace{\mathrm{E}_{p(\boldsymbol{y}' \mid \boldsymbol{x}, \boldsymbol{w})} \left[\mathrm{S}(p(\boldsymbol{y} \mid \boldsymbol{x}, \boldsymbol{w}), \boldsymbol{y}') \right]}_{\text{entropy term}} + \underbrace{\mathrm{E}_{p(\tilde{\boldsymbol{w}} \mid \mathcal{D})} \left[\mathrm{E}_{p(\boldsymbol{y}' \mid \boldsymbol{x}, \boldsymbol{w})} \left[\mathrm{S}(p(\boldsymbol{y} \mid \boldsymbol{x}, \tilde{\boldsymbol{w}}), \boldsymbol{y}') - \mathrm{S}(p(\boldsymbol{y} \mid \boldsymbol{x}, \boldsymbol{w}), \boldsymbol{y}') \right] \right]}_{\text{divergence term}}$$

The expected score over possible output sequences $\boldsymbol{y}'$ and language models $\tilde{\boldsymbol{w}}$ captures the *total* uncertainty of the given language model. The entropy term reflects *aleatoric* uncertainty, which quantifies the inherent stochasticity of generating output sequences with a given language model (Aichberger et al., 2024). The divergence term reflects *epistemic* uncertainty, which quantifies the uncertainty due to lack of knowledge about the true language model parameters arising from limited data (Houlsby et al., 2011; Gal, 2016; Malinin, 2019; Hüllermeier & Waegeman, 2021). Finally, we have not yet specified the proper scoring rule S to derive concrete measures of uncertainty, which we will address in the following sections.

2.2 Established Uncertainty Measures in NLG based on the Logarithmic Score

The logarithmic score is typically assumed implicitly in both the standard classification (Houlsby et al., 2011; Gal, 2016) and the NLG setting (Malinin & Gales, 2021; Kuhn et al., 2023) to derive uncertainty measures. This is due to the grounding of the resulting measures in information theory (Lahlou et al., 2023; Gruber & Buettner, 2023; Hofman et al., 2024; Kotelevskii & Panov, 2024). In the context of NLG, the logarithmic score considers the negative log-likelihood of a generated output sequence $\boldsymbol{y}'$:

$$\mathrm{S}_{\log}(p(\boldsymbol{y} \mid \boldsymbol{x}, \, \cdot \,), \boldsymbol{y}') \; = \; -\log p(\boldsymbol{y} = \boldsymbol{y}' \mid \boldsymbol{x}, \, \cdot \,) \tag{3}$$

Substituting the logarithmic score into Eq. (2) results in the cross-entropy $\mathrm{CE}(\,\cdot\,;\,\cdot\,)$ between the output sequence distribution of the given language model and that of every possible language model according to their posterior probability $p(\tilde{\boldsymbol{w}} \mid \mathcal{D})$ (Aichberger et al., 2024):

$$\underbrace{\mathrm{E}_{p(\tilde{\boldsymbol{w}} \mid \mathcal{D})}\big[\mathrm{CE}(p(\boldsymbol{y} \mid \boldsymbol{x}, \boldsymbol{w}); p(\boldsymbol{y} \mid \boldsymbol{x}, \tilde{\boldsymbol{w}}))\big]}_{\text{total}} \;=\; \underbrace{\mathrm{H}(p(\boldsymbol{y} \mid \boldsymbol{x}, \boldsymbol{w}))}_{\text{aleatoric}} \;+\; \underbrace{\mathrm{E}_{p(\tilde{\boldsymbol{w}} \mid \mathcal{D})}\big[\mathrm{KL}(p(\boldsymbol{y} \mid \boldsymbol{x}, \boldsymbol{w}) \,\|\, p(\boldsymbol{y} \mid \boldsymbol{x}, \tilde{\boldsymbol{w}}))\big]}_{\text{epistemic}} \tag{4}$$

The epistemic uncertainty is captured by a posterior expectation of the Kullback-Leibler divergence $\mathrm{KL}(\,\cdot\,\|\,\cdot\,)$ between the output sequence distribution of the given model and that of all possible models. This requires considering every possible parameterization of the model. Since modern language models have billions of parameters (Radford et al., 2018; Zhang et al., 2022; Touvron et al., 2023; Zuo et al., 2024; Dubey et al., 2024), the epistemic uncertainty is challenging to estimate.

Current work usually focuses on the aleatoric uncertainty, captured by the Shannon entropy $\mathrm{H}(\,\cdot\,)$ of the output sequence distribution of the given language model (Kuhn et al., 2023; Aichberger et al., 2024; Farquhar et al., 2024). Measures based on the logarithmic score (predictive entropy and semantic entropy) take into account the entire distribution of possible output sequences to calculate the uncertainty estimate.

Predictive Entropy. The aleatoric uncertainty under a given language model is the entropy of the output sequence distribution, commonly referred to as Predictive Entropy (PE) (Malinin & Gales, 2021). Intuitively, high PE implies that the language model is likely to generate different output sequences from the same input sequence, indicating high uncertainty of the language model. PE is generally estimated by Monte Carlo (MC) sampling of output sequences (Malinin & Gales, 2021):

$$\begin{aligned} \mathrm{H}(p(\boldsymbol{y} \mid \boldsymbol{x}, \boldsymbol{w})) \;&=\; \mathrm{E}_{p(\boldsymbol{y} \mid \boldsymbol{x}, \boldsymbol{w})}\left[-\log p(\boldsymbol{y} \mid \boldsymbol{x}, \boldsymbol{w})\right] \\ &\approx\; \frac{1}{N}\sum_{n=1}^{N} -\log p(\boldsymbol{y}^n \mid \boldsymbol{x}, \boldsymbol{w})\;, \qquad \boldsymbol{y}^n \sim p(\boldsymbol{y} \mid \boldsymbol{x}, \boldsymbol{w})\;. \end{aligned} \tag{5}$$

Semantic Entropy. Semantic Entropy (SE) (Kuhn et al., 2023; Farquhar et al., 2024) is based on the fact that output sequences may be different on a token level but equivalent on a semantic level. In such cases, the PE can be misleading, as it indicates high uncertainty even when different output sequences have the same semantic meaning. Thus, instead of the entropy of the output sequence distribution, the entropy of the semantic cluster distribution is considered, denoted as $p(c \mid \boldsymbol{x}, \boldsymbol{w}) = \sum_{\mathcal{Y}} p(c \mid \boldsymbol{x}, \boldsymbol{y}, \boldsymbol{w})\, p(\boldsymbol{y} \mid \boldsymbol{x}, \boldsymbol{w})$. The probability of an output sequence belonging to a semantic cluster is usually approximated with a separate natural language inference model. SE thus measures uncertainty about the semantics of output sequences and is defined as

$$\begin{aligned} \mathrm{H}(p(c \mid \boldsymbol{x}, \boldsymbol{w})) \;&=\; \mathrm{E}_{p(c \mid \boldsymbol{x}, \boldsymbol{w})}\left[-\log p(c \mid \boldsymbol{x}, \boldsymbol{w})\right] \\ &\approx\; \frac{1}{N}\sum_{n=1}^{N} -\log p(c^n \mid \boldsymbol{x}, \boldsymbol{w})\;, \qquad c^n \sim p(c \mid \boldsymbol{x}, \boldsymbol{w})\;. \end{aligned} \tag{6}$$

For more details on how to construct a tractable MC approximation of SE, we refer to Aichberger et al. (2024). Each of these uncertainty measures based on the logarithmic score considers the distribution over all possible output sequences $p(\boldsymbol{y} \mid \boldsymbol{x}, \boldsymbol{w})$, which is defined over the entire set of possible output sequences $\mathcal{Y}_T$. Approximating expectations over this distribution requires sampling multiple output sequences from $\mathcal{Y}_T$, which is computationally expensive. In the following, we eliminate this requirement by considering an alternative proper scoring rule.

2.3 New Uncertainty Measures in NLG based on the Zero-One Score

We propose to measure predictive uncertainty in NLG using measures based on the zero-one score instead of the logarithmic score. Although it has been considered in the standard classification setting (Hofman et al., 2024; Kotelevskii & Panov, 2024), to the best of our knowledge, the zero-one score has not yet been considered as a proper scoring rule for deriving uncertainty measures in NLG. The zero-one score considers the predictive distribution for the most likely output sequence:

$$\mathrm{S}_{\text{0-1}}(p(\boldsymbol{y} \mid \boldsymbol{x}, \,\cdot\,), \boldsymbol{y}') \;=\; \big(1 - p(\boldsymbol{y} = \boldsymbol{y}' \mid \boldsymbol{x}, \,\cdot\,)\big)\; \mathbb{1}\{\boldsymbol{y}' = \underset{\boldsymbol{y}}{\operatorname{argmax}}\; p(\boldsymbol{y} \mid \boldsymbol{x}, \,\cdot\,)\} \tag{7}$$

Substituting the zero-one score into Eq. (2) results in the total uncertainty being the expected confidence of the given language model about the most likely output sequences generated by every possible language model according to their posterior probability $p(\tilde{\boldsymbol{w}} \mid \mathcal{D})$:

$$\underbrace{\mathrm{E}_{p(\tilde{\boldsymbol{w}}\mid\mathcal{D})}\left[1 - p(\boldsymbol{y} = \tilde{\boldsymbol{y}}^* \mid \boldsymbol{x}, \boldsymbol{w})\right]}_{\text{total}} = \tag{8}$$
$$\underbrace{1 - p(\boldsymbol{y} = \boldsymbol{y}^* \mid \boldsymbol{x}, \boldsymbol{w})}_{\text{aleatoric}} + \underbrace{p(\boldsymbol{y} = \boldsymbol{y}^* \mid \boldsymbol{x}, \boldsymbol{w}) - \mathrm{E}_{p(\tilde{\boldsymbol{w}}\mid\mathcal{D})}\left[p(\boldsymbol{y} = \tilde{\boldsymbol{y}}^* \mid \boldsymbol{x}, \boldsymbol{w})\right]}_{\text{epistemic}}$$

with $\boldsymbol{y}^* = \operatorname{argmax}_{\boldsymbol{y}} p(\boldsymbol{y} \mid \boldsymbol{x}, \boldsymbol{w})$ for the given language model and $\tilde{\boldsymbol{y}}^* = \operatorname{argmax}_{\boldsymbol{y}} p(\boldsymbol{y} \mid \boldsymbol{x}, \tilde{\boldsymbol{w}})$ for every possible language model. Similarly to Eq. (4), the epistemic uncertainty is a posterior expectation that remains challenging to estimate. Therefore, we again focus on the aleatoric uncertainty, which considers the likelihood of the most likely output sequence under the given language model.

While the aleatoric uncertainty derived from the logarithmic score requires approximating an expectation over the entire output sequence distribution by sampling multiple sequences (see Eq. (5) and Eq. (6)), the one derived from the zero-one score (see Eq. (8)) only requires approximating the most likely output sequence under the given language model $p(\boldsymbol{y} = \boldsymbol{y}^* \mid \boldsymbol{x}, \boldsymbol{w}) = \max_{\boldsymbol{y}} p(\boldsymbol{y} \mid \boldsymbol{x}, \boldsymbol{w})$. This distinction is crucial, as approximating the most likely output sequence aligns directly with standard inference techniques widely used in language models. We propose to approximate the aleatoric uncertainty in Eq. (8) using the greedily decoded output sequence under the given language model. For numerical stability, we consider the negative log-likelihood (NLL) of this output sequence. This gives rise to our uncertainty measure G-NLL, defined as:

$$\texttt{G-NLL} := -\sum_{t=1}^{T} \log \left(\max_{y_t} p(y_t \mid \boldsymbol{x}, \boldsymbol{y}_{<t}, \boldsymbol{w}) \right) \tag{9}$$
$$\approx -\max_{\boldsymbol{y}} \log p(\boldsymbol{y} \mid \boldsymbol{x}, \boldsymbol{w}) \propto 1 - p(\boldsymbol{y} = \boldsymbol{y}^* \mid \boldsymbol{x}, \boldsymbol{w})$$

Our proposed uncertainty measure challenges the prevailing reliance on sampling multiple sequences and semantic clustering to estimate uncertainty in NLG. By solely relying on the output sequences generated with greedy decoding, our approach significantly reduces computational overhead while maintaining theoretical rigor through its foundation in proper scoring rules. Although uncertainty measures based on the logarithmic score could theoretically excel if the full distribution over output sequences $p(\boldsymbol{y} \mid \boldsymbol{x}, \boldsymbol{w})$ were accessible, as is the case in standard classification tasks, this distribution is intractable for NLG tasks due to the autoregressive language modeling in LLMs. As a result, sampling-based methods often yield crude approximations, constrained by computational costs and sampling variability. In contrast, G-NLL offers a principled alternative while eliminating the need for extensive sampling, making our method highly practical and straightforward.

3 Experiments

We aligned our experiments on evaluating uncertainty estimation methods with related work by focusing on free-form question answering tasks (Kuhn et al., 2023; Duan et al., 2024; Bakman et al., 2024; Nikitin et al., 2024; Aichberger et al., 2024; Kossen et al., 2024). While Farquhar et al. (2024) additionally concerns experiments with paragraph-length generations, their approach breaks down the paragraph into factual claims and constructs corresponding questions. Therefore, performance on this task is expected to align with general free-form question answering tasks, and we therefore focused on those for a clearer and more direct evaluation. This focus further avoids potential confounding factors introduced by additional experimental complexities.

Datasets. We evaluated uncertainty estimation methods on three different datasets. We used the more than 3,000 test instances from *TriviaQA* (Joshi et al., 2017) concerning trivia questions, the more than 300 test instances from *SVAMP* (Patel et al., 2021) concerning elementary-level math problems, and the more than 3,600 test instances from *NQ-Open* (Lee et al., 2019) to assess natural questions aggregated from Google Search. Each dataset was used for two distinct tasks: (1) generating concise answers in the form of short phrases and (2) generating more detailed answers in the form of full sentences, following the experimental setup in Farquhar et al. (2024). The six overall tasks cover a broad range, ensuring a comprehensive evaluation.

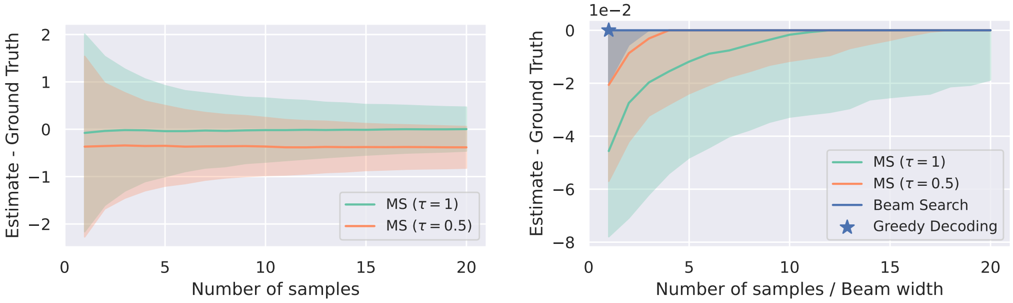
Table 1: Average AUROC across TriviaQA, SVAMP, and NQ datasets, using uncertainty estimates to distinguish between correct and incorrect answers. Varying model architectures (*transformer*, *state-space*), model sizes (*7B*, *8B*, *70B*), and training stages (*PT*, *IT*) are considered. The reference answer is generated using greedy decoding, either as a whole sentence (*long*) or a short phrase (*short*). The reference answers correctness is assessed by *F1* score using SQuAD > 0.5 as decision rule or LLM-as-a-judge (*LLM*). *PE*, *LN-PE*, *SE*, *LN-SE*, and *D-SE* use 10 output sequences (by multinomial sampling) to obtain an uncertainty estimate. G-NLL solely uses the reference answer to obtain an uncertainty estimate.

Uncertainty measure generating scoring rule					*Logarithmic*					*Zero-One*
Language Model			**Generation**	**Metric**	**PE**	**LN-PE**	**SE**	**LN-SE**	**D-SE**	**G-NLL**
Transformer	**8B**	PT	short	F1	.776	.795	.775	.793	.804	**.824**
			short	LLM	.698	.714	.690	.706	.719	**.726**
			long	LLM	.562	.555	.545	.553	.600	**.649**
		IT	short	F1	.772	.801	.805	.814	.806	**.838**
			short	LLM	.676	.697	.704	.709	.694	**.722**
			long	LLM	.551	.548	.599	.601	.609	**.615**
	70B	PT	short	F1	.775	.790	.793	.803	.791	**.820**
			short	LLM	.693	.709	.718	.722	.715	**.723**
			long	LLM	.552	.534	.558	.569	.571	**.649**
		IT	short	F1	.748	.781	.790	**.799**	.783	.792
			short	LLM	.681	.698	.703	**.709**	.699	.699
			long	LLM	.555	.557	.568	.595	**.600**	.562
State-Space	**7B**	PT	short	F1	.811	.815	.809	.822	.828	**.843**
			short	LLM	.705	.711	.701	.711	.716	**.728**
			long	LLM	.567	.597	.574	.611	**.624**	.612
		IT	short	F1	.793	.814	.797	.816	.829	**.838**
			short	LLM	.690	.701	.689	.699	.711	**.719**
			long	LLM	.588	.587	.597	.618	**.629**	.615
Average					.677	.689	.690	.703	.707	**.721**

Models. We conducted our evaluations on six different language models covering various architectures, sizes, and training stages. Specifically, we used the transformer model series *Llama-3.1* (Dubey et al., 2024) and the state-space model series *Falcon Mamba* (Gu & Dao, 2024; Zuo et al., 2024), representing two prominent paradigms. To assess the effect of training stage and scale on uncertainty estimation in NLG, we considered pre-trained (*PT*) and instruction-tuned (*IT*) language models with 7, 8, and 70 billion parameters, covering a wide spectrum of model characteristics.

Baselines. We compare our method against the commonly used uncertainty measures based on the logarithmic score as of Eq. (5) and Eq. (6) and their variants. These include Predictive Entropy (*PE*), length-normalized Predictive Entropy (*LN-PE*) (Malinin & Gales, 2021), Semantic Entropy (*SE*), length-normalized Semantic Entropy (*LN-SE*), and Discrete Semantic Entropy (*D-SE*) (Kuhn et al., 2023; Farquhar et al., 2024). For a given output sequence $\boldsymbol{y}'$, the length-normalized variants consider $\bar{p}(\boldsymbol{y}' \mid \boldsymbol{x}, \boldsymbol{w})$ instead of $p(\boldsymbol{y}' \mid \boldsymbol{x}, \boldsymbol{w})$ to compute the uncertainty estimates. D-SE completely disregards the likelihood of the output sequence and only considers the proportion of output sequences that belong to the same semantic cluster (Farquhar et al., 2024).

Evaluation. Effective uncertainty measures should accurately reflect the reliability of answers generated by the language model. In other words, higher uncertainty should correspond to a greater tendency for incorrect outputs. Thus, to evaluate the performance of an uncertainty estimator, we assess how well it correlates with the correctness of the language model's answers. Correct answers should be assigned a lower uncertainty estimator than incorrect answers. To determine whether an answer is correct, it has to be compared to the respective ground truth answer. To do so, we check if the F1 score of the commonly used SQuAD metric exceeds 0.5 (Rajpurkar et al., 2016). Although there are some limitations to using such a simple metric, it has relatively small errors in standard datasets and, therefore, remains widely used in practice. However, this metric is only applicable for short-phrase generations that align with the ground truth answer. Therefore, we additionally employ Llama-3.1 with 70 billion parameters (Dubey et al., 2024) as an LLM-as-a-judge to assess the correctness of both short-phrase and full-sentence generations. To measure the correlation between the incorrectness of answers and the respective uncertainty estimates, we use the AUROC. Higher



(a) Estimating $\mathrm{H}(p(\boldsymbol{y} \mid \boldsymbol{x}))$

(b) Estimating $\max_{\boldsymbol{y}} p(\boldsymbol{y} \mid \boldsymbol{x})$

Figure 1: Quality of estimators for synthetic predictive distributions $p(\boldsymbol{y} \mid \boldsymbol{x})$ with $|\mathcal{V}| = 20$ and $T = 4$. The predictive entropy $\mathrm{H}(p(\boldsymbol{y} \mid \boldsymbol{x}))$ is estimated as in Eq. (5) using multinomial sampling (MS) with different temperatures (τ). The maximum sequence likelihood $\max_{\boldsymbol{y}} p(\boldsymbol{y} \mid \boldsymbol{x})$ is estimated by the maximum over N samples obtained by beam search ($N = 1$ represents greedy decoding) or MS with different τ. Statistics are obtained by sampling different $p(\boldsymbol{y} \mid \boldsymbol{x})$. (a) Lines show average, shades denote one std. (b) Lines show the median, and shades denote 5% to 95% quantile range.

AUROC values indicate better performance of the uncertainty estimator, as it reflects a stronger alignment between the correctness of the language model's answers and their respective uncertainty estimates. In summary, this evaluation process follows established methodologies for assessing the performance of uncertainty measures in NLG (Kuhn et al., 2023; Duan et al., 2024; Farquhar et al., 2024; Nikitin et al., 2024; Aichberger et al., 2024; Kossen et al., 2024).

3.1 Main Results

Tab. 1 summarizes the performance of the uncertainty measures across the six language models, six tasks, and two evaluation metrics. We report the average AUROC across the three datasets, highlighting the best-performing measure in bold. Additionally, the best-performing measure based on the logarithmic score is underlined, unless it also represents the overall best. In 13 out of 18 scenarios, G-NLL achieves the best performance among all uncertainty measures and remains competitive in the remaining 5 scenarios. This strong performance is particularly evident in tasks involving the generation of short phrases, suggesting its effectiveness in capturing the essential part of the output sequence that contains the factual answer to a question. This is especially valuable in practical scenarios, where the uncertainty about a specific fact is often more critical than the uncertainty about the entire generated sentence. Overall, our measure significantly outperforms all other measures when considering the average across all scenarios. This demonstrates that our measure achieves state-of-the-art performance while considering only a single output sequence. Detailed results for individual datasets and additional evaluations are provided in Apx. B.

3.2 Quality of Estimators

The strong empirical performance of G-NLL on question answering NLG tasks suggests that it effectively captures key aspects of uncertainty, even when using significantly fewer output sequences compared to uncertainty measures based on the logarithmic score. To examine the underlying factors driving this behavior, we analyze these estimators in both synthetic and real-world settings.

Synthetic setting. For the first study, we examine the reliability of estimators of aleatoric uncertainty under the zero-one score compared to those under the logarithmic score in the low-sample regimes typical of uncertainty estimation in NLG. To this end, we conduct a synthetic experiment where we sample predictive distributions $p(\boldsymbol{y} \mid \boldsymbol{x})$ with smaller vocabulary sizes and shorter sequence lengths, while preserving the distributional characteristics typical of language models. This approach allows us to obtain ground truths for both $\mathrm{H}(p(\boldsymbol{y} \mid \boldsymbol{x}))$ and $\max_{\boldsymbol{y}} p(\boldsymbol{y} \mid \boldsymbol{x})$, which become intractable with larger vocabulary sizes and sequence lengths. Using this synthetic setup, we aim to evaluate how the quality of estimators improves with the number of samples.

Fig. 1a summarizes the results for estimating the entropy, derived from the logarithmic score. The results show that low sample sizes lead to high estimator variance. Similarly, Fig. 1b summarizes the results for estimating the maximum sequence likelihood, derived from the zero-one score. The results indicate that, while multinomial sampling (MS) also exhibits higher variance with fewer samples, heuristics such as beam search (BS) and even greedy decoding provide accurate estimates of the maximum sequence probability with negligible variance. Details on the sampling procedure of the predictive distributions, as well as additional experiments are provided in Apx. C. This synthetic experiment suggests that MS is ineffective at estimating $\max_{\boldsymbol{y}} p(\boldsymbol{y} \mid \boldsymbol{x})$, whereas greedy decoding, as we propose in Eq. (9), performs well and could be sightly improved by using BS with larger width. In the following ablation, we further investigate the empirical performance of the different sampling methods with an actual language model.

Real-world setting. For the second study, we evaluate the empirical effectiveness of approximating the aleatoric uncertainty derived from the zero-one score in Eq. 8. We recall that this requires approximating the NLL of the most likely output sequence under the given language model $-\log \max_{\boldsymbol{y}} p(\boldsymbol{y} \mid \boldsymbol{x}, \boldsymbol{w})$. We investigate the performance of various sampling methods for this approximation. Specifically, we utilize MS with different temperatures τ, BS with different beam sizes, and greedy decoding, which is the sampling method proposed for G-NLL. For each sampling method, we generate a single output sequence per instance in the TriviaQA dataset. We then use the corresponding NLL as the uncertainty estimate, following the same evaluation process as in the main experiments above. Notably, the baselines are unaffected by the choice of sampling method for computing the NLL, as we again use their optimal hyperparameter settings for sampling.

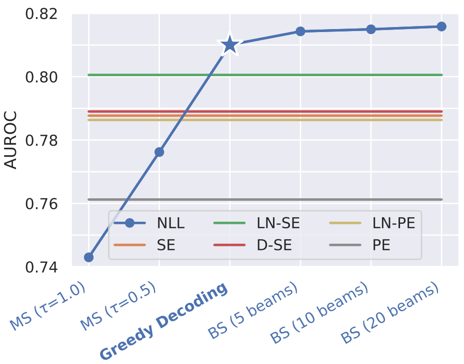


Figure 2: Performance of NLL as uncertainty measure based on the zero-one score, when the output sequence is generated with different sampling methods (MS, greedy decoding, and BS), compared to uncertainty measures based on the logarithmic score.

The results in Fig. 2 show that better approximations of the most likely output sequences indeed lead to higher uncertainty estimation performance, reinforcing the validity of our alternative measure derived from the zero-one score. Additionally, it can be observed that sampling output sequences using greedy decoding significantly outperforms MS. While performance improves further with BS, as anticipated, the marginal benefits are relatively small. This supports the claim that greedy decoding provides a strong approximation to the most likely output sequence. Since BS is computationally more expensive (as its beam size corresponds to the number of sampled output sequences), using greedy decoding in G-NLL achieves the best trade-off between effectiveness and efficiency.

4 Conclusion

In this work, we introduced an alternative uncertainty measure: the NLL of the most likely output sequence under a given language model. This measure is motivated by the general notion of proper scoring rules, utilizing the zero-one score as an alternative to the prevalent logarithmic score. Unlike previous measures, it does not require sampling multiple output sequences but can be efficiently approximated using G-NLL through a single, greedily decoded output sequence. The experiments demonstrate that our measure largely outperforms previous measures that entail considerably higher computational costs and algorithmic complexity.

Although G-NLL effectively captures uncertainty, it currently does not account for the semantics of the generated output sequence. Future work should explore extensions that incorporate semantic meaning to further enhance the uncertainty estimator while preserving its computational efficiency. Moreover, measures based on proper scoring rules rely on heuristics, such as length normalization, to address variations in sequence lengths (Malinin & Gales, 2021; Duan et al., 2024; Bakman et al., 2024; Yaldiz et al., 2024). Investigating theoretically grounded methods to handle these characteristics that vary for different sequences represents another promising direction for future work.

While there remain opportunities for refinement, our proposed measure provides a solid foundation for efficient and reliable uncertainty estimation in NLG. It paves the way for efficient methods that build upon a single output sequence. Given its simplicity and minimal computational overhead, G-NLL serves as a strong baseline for developing new uncertainty measures and represents an important step toward scalable uncertainty estimation in real-world applications.

REFERENCES

Yasin Abbasi-Yadkori, Ilja Kuzborskij, András György, and Csaba Szepesvari. To believe or not to believe your LLM: Iterative prompting for estimating epistemic uncertainty. In *The Thirty-eighth Annual Conference on Neural Information Processing Systems*, 2024.

Lukas Aichberger, Kajetan Schweighofer, Mykyta Ielanskyi, and Sepp Hochreiter. Semantically diverse language generation for uncertainty estimation in language models. *arXiv*, 2406.04306, 2024.

Yavuz Faruk Bakman, Duygu Nur Yaldiz, Baturalp Buyukates, Chenyang Tao, Dimitrios Dimitriadis, and Salman Avestimehr. MARS: Meaning-aware response scoring for uncertainty estimation in generative LLMs. In Lun-Wei Ku, Andre Martins, and Vivek Srikumar (eds.), *Proceedings of the 62nd Annual Meeting of the Association for Computational Linguistics*, pp. 7752–7767. Association for Computational Linguistics, 2024.

Chao Chen, Kai Liu, Ze Chen, Yi Gu, Yue Wu, Mingyuan Tao, Zhihang Fu, and Jieping Ye. INSIDE: LLMs' internal states retain the power of hallucination detection. In *The Twelfth International Conference on Learning Representations*, 2024.

Roi Cohen, Mor Geva, Jonathan Berant, and Amir Globerson. Crawling the internal knowledge-base of language models. In Andreas Vlachos and Isabelle Augenstein (eds.), *Findings of the Association for Computational Linguistics: EACL 2023*, pp. 1856–1869. Association for Computational Linguistics, 2023a.

Roi Cohen, May Hamri, Mor Geva, and Amir Globerson. LM vs LM: Detecting factual errors via cross examination. In Houda Bouamor, Juan Pino, and Kalika Bali (eds.), *Proceedings of the 2023 Conference on Empirical Methods in Natural Language Processing*, pp. 12621–12640. Association for Computational Linguistics, 2023b.

Jinhao Duan, Hao Cheng, Shiqi Wang, Alex Zavalny, Chenan Wang, Renjing Xu, Bhavya Kailkhura, and Kaidi Xu. Shifting attention to relevance: Towards the predictive uncertainty quantification of free-form large language models. In Lun-Wei Ku, Andre Martins, and Vivek Srikumar (eds.), *Proceedings of the 62nd Annual Meeting of the Association for Computational Linguistics*, pp. 5050–5063, 2024.

Abhimanyu Dubey, Abhinav Jauhri, Abhinav Pandey, Abhishek Kadian, Ahmad Al-Dahle, Aiesha Letman, Akhil Mathur, Alan Schelten, Amy Yang, Angela Fan, Anirudh Goyal, Anthony Hartshorn, Aobo Yang, Archi Mitra, Archie Sravankumar, Artem Korenev, Arthur Hinsvark, Arun Rao, Aston Zhang, and Aurelien Rodriguez et al. The llama 3 herd of models. *arXiv*, 2407.21783, 2024.

Ekaterina Fadeeva, Roman Vashurin, Akim Tsvigun, Artem Vazhentsev, Sergey Petrakov, Kirill Fedyanin, Daniil Vasilev, Elizaveta Goncharova, Alexander Panchenko, Maxim Panov, Timothy Baldwin, and Artem Shelmanov. LM-polygraph: Uncertainty estimation for language models. In Yansong Feng and Els Lefever (eds.), *Proceedings of the 2023 Conference on Empirical Methods in Natural Language Processing: System Demonstrations*, pp. 446–461. Association for Computational Linguistics, 2023.

Ekaterina Fadeeva, Aleksandr Rubashevskii, Artem Shelmanov, Sergey Petrakov, Haonan Li, Hamdy Mubarak, Evgenii Tsymbalov, Gleb Kuzmin, Alexander Panchenko, Timothy Baldwin, Preslav Nakov, and Maxim Panov. Fact-checking the output of large language models via token-level uncertainty quantification. In Lun-Wei Ku, Andre Martins, and Vivek Srikumar (eds.), *Findings of the Association for Computational Linguistics: ACL 2024*, pp. 9367–9385. Association for Computational Linguistics, 2024.

Sebastian Farquhar, Jannik Kossen, Lorenz Kuhn, and Yarin Gal. Detecting hallucinations in large language models using semantic entropy. *Nature*, 630(8017):625–630, 2024.

Yarin Gal. *Uncertainty in deep learning*. PhD thesis, University of Cambridge, 2016.

Deep Ganguli, Amanda Askell, Nicholas Schiefer, Thomas I. Liao, Kamilė Lukošiūtė, Anna Chen, Anna Goldie, Azalia Mirhoseini, Catherine Olsson, Danny Hernandez, Dawn Drain, Dustin Li, Eli Tran-Johnson, Ethan Perez, Jackson Kernion, Jamie Kerr, Jared Mueller, Joshua Landau, Kamal Ndousse, Karina Nguyen, Liane Lovitt, Michael Sellitto, Nelson Elhage, Noemi Mercado, Nova DasSarma, Oliver Rausch, Robert Lasenby, Robin Larson, Sam Ringer, Sandipan Kundu, Saurav Kadavath, Scott Johnston, Shauna Kravec, Sheer El Showk, Tamera Lanham, Timothy Telleen-Lawton, Tom Henighan, Tristan Hume, Yuntao Bai, Zac Hatfield-Dodds, Ben Mann, Dario Amodei, Nicholas Joseph, Sam McCandlish, Tom Brown, Christopher Olah, Jack Clark, Samuel R. Bowman, and Jared Kaplan. The capacity for moral self-correction in large language models. *arXiv*, 2302.07459, 2023.

Tilmann Gneiting and Adrian E Raftery. Strictly proper scoring rules, prediction, and estimation. *Journal of the American statistical Association*, 102(477):359–378, 2007.

Sebastian Gruber and Florian Buettner. Uncertainty estimates of predictions via a general bias-variance decomposition. In Francisco Ruiz, Jennifer Dy, and Jan-Willem van de Meent (eds.), *Proceedings of The 26th International Conference on Artificial Intelligence and Statistics*, volume 206, pp. 11331–11354. PMLR, 2023.

Albert Gu and Tri Dao. Mamba: Linear-time sequence modeling with selective state spaces. In *First Conference on Language Modeling*, 2024.

Paul Hofman, Yusuf Sale, and Eyke Hüllermeier. Quantifying aleatoric and epistemic uncertainty with proper scoring rules. *arXiv*, 2404.12215, 2024.

Neil Houlsby, Ferenc Huszár, Zoubin Ghahramani, and Máté Lengyel. Bayesian active learning for classification and preference learning. *arXiv*, 1112.5745, 2011.

Eyke Hüllermeier and Willem Waegeman. Aleatoric and epistemic uncertainty in machine learning: An introduction to concepts and methods. *Machine Learning*, 110:457–506, 2021.

Mandar Joshi, Eunsol Choi, Daniel S. Weld, and Luke Zettlemoyer. Triviaqa: A large scale distantly supervised challenge dataset for reading comprehension. In *Proceedings of the 55th Annual Meeting of the Association for Computational Linguistics*. Association for Computational Linguistics, 2017.

Saurav Kadavath, Tom Conerly, Amanda Askell, Tom Henighan, Dawn Drain, Ethan Perez, Nicholas Schiefer, Zac Hatfield-Dodds, Nova DasSarma, Eli Tran-Johnson, Scott Johnston, Sheer El-Showk, Andy Jones, Nelson Elhage, Tristan Hume, Anna Chen, Yuntao Bai, Sam Bowman, Stanislav Fort, Deep Ganguli, Danny Hernandez, Josh Jacobson, Jackson Kernion, Shauna Kravec, Liane Lovitt, Kamal Ndousse, Catherine Olsson, Sam Ringer, Dario Amodei, Tom Brown, Jack Clark, Nicholas Joseph, Ben Mann, Sam McCandlish, Chris Olah, and Jared Kaplan. Language models (mostly) know what they know. *arXiv*, 2207.05221, 2022.

Jannik Kossen, Jiatong Han, Muhammed Razzak, Lisa Schut, Shreshth Malik, and Yarin Gal. Semantic entropy probes: Robust and cheap hallucination detection in llms. *arXiv*, 2406.15927, 2024.

Nikita Kotelevskii and Maxim Panov. Predictive uncertainty quantification via risk decompositions for strictly proper scoring rules. *arXiv*, 2402.10727, 2024.

Lorenz Kuhn, Yarin Gal, and Sebastian Farquhar. Semantic uncertainty: Linguistic invariances for uncertainty estimation in natural language generation. In *The Eleventh International Conference on Learning Representations*, 2023.

Meelis Kull and Peter Flach. Novel decompositions of proper scoring rules for classification: Score adjustment as precursor to calibration. In Annalisa Appice, Pedro Pereira Rodrigues, Vítor Santos Costa, Carlos Soares, João Gama, and Alípio Jorge (eds.), *Machine Learning and Knowledge Discovery in Databases*, pp. 68–85. Springer International Publishing, 2015.

Salem Lahlou, Moksh Jain, Hadi Nekoei, Victor I Butoi, Paul Bertin, Jarrid Rector-Brooks, Maksym Korablyov, and Yoshua Bengio. DEUP: Direct epistemic uncertainty prediction. *Transactions on Machine Learning Research*, 2023. ISSN 2835-8856.

Kenton Lee, Ming-Wei Chang, and Kristina Toutanova. Latent retrieval for weakly supervised open domain question answering. In Anna Korhonen, David Traum, and Lluís Màrquez (eds.), *Proceedings of the 57th Annual Meeting of the Association for Computational Linguistics*, pp. 6086–6096. Association for Computational Linguistics, 2019.

Stephanie Lin, Jacob Hilton, and Owain Evans. Teaching models to express their uncertainty in words. *Transactions on Machine Learning Research*, 2022. ISSN 2835-8856.

Andrey Malinin. *Uncertainty estimation in deep learning with application to spoken language assessment*. PhD thesis, University of Cambridge, 2019.

Andrey Malinin and Mark Gales. Uncertainty estimation in autoregressive structured prediction. In *International Conference on Learning Representations*, 2021.

Potsawee Manakul, Adian Liusie, and Mark Gales. SelfCheckGPT: Zero-resource black-box hallucination detection for generative large language models. In Houda Bouamor, Juan Pino, and Kalika Bali (eds.), *Proceedings of the 2023 Conference on Empirical Methods in Natural Language Processing*, pp. 9004–9017. Association for Computational Linguistics, 2023.

Sabrina J. Mielke, Arthur Szlam, Emily Dinan, and Y-Lan Boureau. Reducing conversational agents' overconfidence through linguistic calibration. *Transactions of the Association for Computational Linguistics*, 10:857–872, 2022.

Alexander V Nikitin, Jannik Kossen, Yarin Gal, and Pekka Marttinen. Kernel language entropy: Fine-grained uncertainty quantification for LLMs from semantic similarities. In *The Thirty-eighth Annual Conference on Neural Information Processing Systems*, 2024.

Arkil Patel, Satwik Bhattamishra, and Navin Goyal. Are NLP models really able to solve simple math word problems? In Kristina Toutanova, Anna Rumshisky, Luke Zettlemoyer, Dilek Hakkani-Tur, Iz Beltagy, Steven Bethard, Ryan Cotterell, Tanmoy Chakraborty, and Yichao Zhou (eds.), *Proceedings of the 2021 Conference of the North American Chapter of the Association for Computational Linguistics: Human Language Technologies*, pp. 2080–2094. Association for Computational Linguistics, 2021.

Benjamin Plaut, Nguyen X. Khanh, and Tu Trinh. Probabilities of chat llms are miscalibrated but still predict correctness on multiple-choice q&a. *arXiv*, 2402.13213, 2024.

Xin Qiu and Risto Miikkulainen. 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*, 2024.

Victor Quach, Adam Fisch, Tal Schuster, Adam Yala, Jae Ho Sohn, Tommi S. Jaakkola, and Regina Barzilay. Conformal language modeling. In *The Twelfth International Conference on Learning Representations*, 2024.

Alec Radford, Jeffrey Wu, Rewon Child, David Luan, Dario Amodei, Ilya Sutskever, et al. Language models are unsupervised multitask learners. *OpenAI blog*, 1(8):9, 2018.

Pranav Rajpurkar, Jian Zhang, Konstantin Lopyrev, and Percy Liang. SQuAD: 100,000+ questions for machine comprehension of text. In Jian Su, Kevin Duh, and Xavier Carreras (eds.), *Proceedings of the 2016 Conference on Empirical Methods in Natural Language Processing*, pp. 2383–2392. Association for Computational Linguistics, 2016.

Jie Ren, Jiaming Luo, Yao Zhao, Kundan Krishna, Mohammad Saleh, Balaji Lakshminarayanan, and Peter J Liu. Out-of-distribution detection and selective generation for conditional language models. In *The Eleventh International Conference on Learning Representations*, 2023a.

Jie Ren, Yao Zhao, Tu Vu, Peter J. Liu, and Balaji Lakshminarayanan. Self-evaluation improves selective generation in large language models. *arXiv*, 2312.09300, 2023b.

Kajetan Schweighofer, Lukas Aichberger, Mykyta Ielanskyi, and Sepp Hochreiter. Introducing an improved information-theoretic measure of predictive uncertainty. *arXiv*, 2311.08309, 2023a.

Kajetan Schweighofer, Lukas Aichberger, Mykyta Ielanskyi, Günter Klambauer, and Sepp Hochreiter. Quantification of uncertainty with adversarial models. In A. Oh, T. Naumann, A. Globerson, K. Saenko, M. Hardt, and S. Levine (eds.), *Advances in Neural Information Processing Systems*, volume 36, pp. 19446–19484. Curran Associates, Inc., 2023b.

Kajetan Schweighofer, Lukas Aichberger, Mykyta Ielanskyi, and Sepp Hochreiter. On information-theoretic measures of predictive uncertainty. *arXiv*, 2410.10786, 2024.

Ilya Sutskever, Oriol Vinyals, and Quoc V Le. Sequence to sequence learning with neural networks. In Z. Ghahramani, M. Welling, C. Cortes, N. Lawrence, and K.Q. Weinberger (eds.), *Advances in Neural Information Processing Systems*, volume 27. Curran Associates, Inc., 2014.

Katherine Tian, Eric Mitchell, Allan Zhou, Archit Sharma, Rafael Rafailov, Huaxiu Yao, Chelsea Finn, and Christopher Manning. Just ask for calibration: Strategies for eliciting calibrated confidence scores from language models fine-tuned with human feedback. In Houda Bouamor, Juan Pino, and Kalika Bali (eds.), *Proceedings of the 2023 Conference on Empirical Methods in Natural Language Processing*, pp. 5433–5442. Association for Computational Linguistics, 2023.

Hugo Touvron, Louis Martin, Kevin Stone, Peter Albert, Amjad Almahairi, Yasmine Babaei, Nikolay Bashlykov, Soumya Batra, Prajjwal Bhargava, Shruti Bhosale, Dan Bikel, Lukas Blecher, Cristian Canton Ferrer, Moya Chen, Guillem Cucurull, David Esiobu, Jude Fernandes, Jeremy Fu, Wenyin Fu, Brian Fuller, Cynthia Gao, Vedanuj Goswami, Naman Goyal, Anthony Hartshorn, Saghar Hosseini, Rui Hou, Hakan Inan, Marcin Kardas, Viktor Kerkez, Madian Khabsa, Isabel Kloumann, Artem Korenev, Punit Singh Koura, Marie-Anne Lachaux, Thibaut Lavril, Jenya Lee, Diana Liskovich, Yinghai Lu, Yuning Mao, Xavier Martinet, Todor Mihaylov, Pushkar Mishra, Igor Molybog, Yixin Nie, Andrew Poulton, Jeremy Reizenstein, Rashi Rungta, Kalyan Saladi, Alan Schelten, Ruan Silva, Eric Michael Smith, Ranjan Subramanian, Xiaoqing Ellen Tan, Binh Tang, Ross Taylor, Adina Williams, Jian Xiang Kuan, Puxin Xu, Zheng Yan, Iliyan Zarov, Yuchen Zhang, Angela Fan, Melanie Kambadur, Sharan Narang, Aurelien Rodriguez, Robert Stojnic, Sergey Edunov, and Thomas Scialom. Llama 2: Open foundation and fine-tuned chat models. *arXiv*, 2307.09288, 2023.

Artem Vazhentsev, Ekaterina Fadeeva, Rui Xing, Alexander Panchenko, Preslav Nakov, Timothy Baldwin, Maxim Panov, and Artem Shelmanov. Unconditional truthfulness: Learning conditional dependency for uncertainty quantification of large language models. *arXiv*, 2408.10692, 2024.

Yuxin Xiao, Paul Pu Liang, Umang Bhatt, Willie Neiswanger, Ruslan Salakhutdinov, and Louis-Philippe Morency. Uncertainty quantification with pre-trained language models: A large-scale empirical analysis. In Yoav Goldberg, Zornitsa Kozareva, and Yue Zhang (eds.), *Findings of the Association for Computational Linguistics: EMNLP 2022*, pp. 7273–7284. Association for Computational Linguistics, 2022.

Duygu Nur Yaldiz, Yavuz Faruk Bakman, Baturalp Buyukates, Chenyang Tao, Anil Ramakrishna, Dimitrios Dimitriadis, Jieyu Zhao, and Salman Avestimehr. Do not design, learn: A trainable scoring function for uncertainty estimation in generative llms. *arXiv*, 2406.11278, 2024.

Andi Zhang, Tim Z. Xiao, Weiyang Liu, Robert Bamler, and Damon Wischik. Your finetuned large language model is already a powerful out-of-distribution detector. *arXiv*, 2404.08679, 2024.

Susan Zhang, Stephen Roller, Naman Goyal, Mikel Artetxe, Moya Chen, Shuohui Chen, Christopher Dewan, Mona Diab, Xian Li, Xi Victoria Lin, Todor Mihaylov, Myle Ott, Sam Shleifer, Kurt Shuster, Daniel Simig, Punit Singh Koura, Anjali Sridhar, Tianlu Wang, and Luke Zettlemoyer. Opt: Open pre-trained transformer language models. *arXiv*, 2205.01068, 2022.

Kaitlyn Zhou, Dan Jurafsky, and Tatsunori Hashimoto. Navigating the grey area: How expressions of uncertainty and overconfidence affect language models. In Houda Bouamor, Juan Pino, and Kalika Bali (eds.), *Proceedings of the 2023 Conference on Empirical Methods in Natural Language Processing*, pp. 5506–5524. Association for Computational Linguistics, 2023.

Jingwei Zuo, Maksim Velikanov, Dhia Eddine Rhaiem, Ilyas Chahed, Younes Belkada, Guillaume Kunsch, and Hakim Hacid. Falcon mamba: The first competitive attention-free 7b language model. *arXiv*, 2410.05355, 2024.

A Related Work

There is a body of work that extends the concept of SE (Kuhn et al., 2023; Farquhar et al., 2024), for example, by improving semantic clustering (Nikitin et al., 2024; Qiu & Miikkulainen, 2024), improving the sampling of output sequences (Aichberger et al., 2024), or directly approximating the measure from hidden states of the language model (Kossen et al., 2024; Chen et al., 2024). In addition, multiple works build upon PE (Malinin & Gales, 2021), for example, by considering a weighting factor for individual token and sequence likelihoods to account for the importance on a semantic level (Duan et al., 2024; Bakman et al., 2024; Yaldiz et al., 2024).

There are also works that use the likelihood of a single output sequence as a heuristic baseline. For example, Fadeeva et al. (2023), Fadeeva et al. (2024), and Vazhentsev et al. (2024) consider the most likely output sequence in their experiments. Bakman et al. (2024) and the follow-up work by Yaldiz et al. (2024) consider the sequence likelihood as part of their uncertainty estimation method. Abbasi-Yadkori et al. (2024) use greedy decoded sequence likelihood as a baseline. Plaut et al. (2024) show that maximum softmax probabilities predict correctness in question answering. Ren et al. (2023a) use perplexity as a baseline for OOD detection, stating that it alone is ill-suited for this task. Zhang et al. (2024) use the likelihood ratio between pre-trained and fine-tuned language models for OOD detection, claiming that this ratio achieves high performance.

Other works on uncertainty estimation in NLG do not consider uncertainty measures grounded in proper scoring rules. For example, several approaches leverage the language model itself to directly predict uncertainty, either through numerical estimates or verbal explanations (Mielke et al., 2022; Lin et al., 2022; Kadavath et al., 2022; Cohen et al., 2023a; Ganguli et al., 2023; Ren et al., 2023b; Tian et al., 2023). Additionally, Cohen et al. (2023b) employ cross-examination, where one language model generates an output sequence and another model acts as an examiner to assess uncertainty. Zhou et al. (2023) explore the behavior of language models when expressing their uncertainty, providing insights into how models articulate confidence in their predictions. Manakul et al. (2023) propose using sampled output sequences as input to another language model to assess uncertainty.Xiao et al. (2022) provide an empirical analysis of how factors such as model architecture and training data influence uncertainty estimates. Finally, conformal prediction (Quach et al., 2024) offers another approach by calibrating a stopping rule for output sequence generation.

B Detailed Main Results

In this section, we provide detailed insights to complement the main results presented in Sec. 3.1.

To compute G-NLL, greedy decoding is used to generate the reference answer, which is equivalent to beam search with a single beam and multinomial sampling with a sampling temperature of zero. The correctness of the reference answer is assessed by checking if the F1 score of the commonly used SQuAD metric exceeds 0.5 (*F1*) or if the Llama-3.1-70B model used as LLM-as-a-judge considers it as correct (*LLM*). To compute the logarithmic score based measures (*PE*, *LN-PE*, *SE*, *LN-SE*, *D-SE*) ten output sequences are generated via multinomial sampling. For each dataset, we performed a hyperparameter search on held-out instances to determine the best-performing temperature $\tau \in \{0.5, 1.0, 1.5\}$ for sampling output sequences used for the logarithmic score based measures.

AUROC is used as the primary performance metric throughout this paper, consistent with standard evaluation practices in this field. We report the results for the individual datasets in Tab. 2, which has been averaged over in Tab. 1.

In addition to AUROC, we also consider the average rejection accuracy, i.e., the accuracy of model predictions when allowing the rejection of a certain budget of predictions based on the uncertainty estimate. Results are presented in Tab. 3, where predictions are only evaluated for the 80% most certain predictions, and we again use greedy decoding for our measure based on the zero-one score. This further suggests that our measure remains highly competitive across various settings.

Additionally, we look into how much the alternative measure derived from the zero-one score benefits from better approximating the most likely output sequences through increasing the beam width to 5. The results summarized in Tab. 4 show that the performance does not significantly improve compared to greedy decoding, which is consistent with the ablation study presented in Fig. 2. This further supports the claim that G-NLL is a strong measure of uncertainty, despite its algorithmic simplicity and computational efficiency.

Table 2: **AUROC Evaluation.** Average AUROC using uncertainty estimates of different measures as a score to distinguish between correct and incorrect answers of each dataset. The reference answer is generated using greedy decoding, either as a whole sentence (*long*) or a short phrase (*short*).

Uncertainty measure generating scoring rule						*Logarithmic*					*Zero-One*
$\mathcal{D}$	**Language Model**			**Gen.**	**Metric**	**PE**	**LN-PE**	**SE**	**LN-SE**	**D-SE**	**G-NLL**
TriviaQA	**Transformer**	**8B**	PT	short	F1	.758	.778	.788	.798	.787	**.810**
				short	LLM	.675	.694	.703	.704	.682	**.722**
				long	LLM	.592	.604	.640	.631	.650	**.704**
			IT	short	F1	.735	.768	.790	.800	.777	**.809**
				short	LLM	.660	.684	.708	.710	.680	**.716**
				long	LLM	.603	.627	**.678**	.672	.670	.670
		70B	PT	short	F1	.707	.730	.741	.743	.702	**.744**
				short	LLM	.650	.660	.696	.695	.656	**.698**
				long	LLM	.538	.533	.625	.574	.563	**.692**
			IT	short	F1	.698	.714	.722	**.726**	.688	.722
				short	LLM	.663	.675	.685	.679	.633	**.701**
				long	LLM	.530	.553	.564	**.571**	.564	.543
	State-Space	**7B**	PT	short	F1	.786	.793	.812	.818	.810	**.832**
				short	LLM	.687	.697	.712	.714	.695	**.724**
				long	LLM	.597	.653	.675	.680	.689	**.705**
			PT	short	F1	.780	.799	.810	.819	.811	**.827**
				short	LLM	.696	.701	.714	.717	.703	**.730**
				long	LLM	.645	.654	.688	**.698**	.692	.694
SVAMP	**Transformer**	**8B**	PT	short	F1	.847	.867	.865	.870	.868	**.885**
				short	LLM	.779	.788	.753	.772	**.791**	.772
				long	LLM	.575	.563	.519	.534	.601	**.669**
			IT	short	F1	.879	.903	.914	.912	.887	**.931**
				short	LLM	.706	.725	.736	.731	.701	**.753**
				long	LLM	.556	.524	.590	.608	.631	**.662**
		70B	PT	short	F1	.892	.906	.925	.929	.923	**.936**
				short	LLM	.794	.817	.814	.815	**.819**	.799
				long	LLM	.578	.554	.553	.579	.571	**.665**
			IT	short	F1	.830	.895	.915	**.922**	.915	.909
				short	LLM	.703	.744	.734	.748	**.762**	.713
				long	LLM	.601	.577	.613	.649	**.663**	.597
	State-Space	**7B**	PT	short	F1	.882	.893	.874	.883	.889	**.914**
				short	LLM	.752	.757	.730	.738	.757	**.776**
				long	LLM	.536	.585	.534	.602	**.612**	.579
			IT	short	F1	.843	.891	.854	.876	.892	**.905**
				short	LLM	.706	.730	.704	.709	.737	**.744**
				long	LLM	.577	.586	.578	.616	**.639**	.613
NQ	**Transformer**	**8B**	PT	short	F1	.725	.739	.673	.710	.758	**.776**
				short	LLM	.639	.661	.615	.641	**.683**	**.683**
				long	LLM	.517	.498	.478	.495	.550	**.573**
			IT	short	F1	.702	.732	.711	.731	.756	**.774**
				short	LLM	.662	.682	.669	.685	**.700**	.697
				long	LLM	.494	.491	**.530**	.524	.527	.514
		70B	PT	short	F1	.727	.733	.711	.737	.748	**.779**
				short	LLM	.634	.649	.642	.657	.671	**.672**
				long	LLM	.538	.514	.494	.553	.580	**.589**
			IT	short	F1	.718	.734	.734	**.748**	.746	.743
				short	LLM	.676	.674	.689	.698	**.702**	.681
				long	LLM	.535	.540	.526	.566	**.574**	.545
	State-Space	**7B**	PT	short	F1	.766	.758	.741	.765	**.785**	.782
				short	LLM	.675	.680	.661	.681	**.697**	.683
				long	LLM	.567	.553	.512	.551	**.572**	.554
			IT	short	F1	.755	.751	.727	.754	**.783**	.781
				short	LLM	.669	.672	.648	.671	**.692**	.683
				long	LLM	.541	.521	.526	.541	**.554**	.537
Average						.677	.689	.690	.703	.707	**.721**

Table 3: **Rejection Accuracy Evaluation.** Average Rejection Accuracy (80%) across all datasets. The reference answer is generated using greedy decoding, either as a whole sentence (*long*) or a short phrase (*short*).

Uncertainty measure generating scoring rule				*Logarithmic*					*Zero-One*
	Language Model	**Gen.**	**Metric**	**PE**	**LN-PE**	**SE**	**LN-SE**	**D-SE**	**G-NLL**
Transformer	**8b** PT	short	F1	.661	.672	.651	.643	.655	**.681**
			LLM	.774	**.782**	.767	.766	.765	.778
			LLM-Instruct	.704	.721	.693	.688	.702	**.723**
		long	LLM	.596	.590	.598	.592	.590	**.619**
			LLM-Instruct	.667	.684	.632	.643	.644	**.686**
	IT	short	F1	.668	.684	.680	.673	.687	**.702**
			LLM	.775	.781	.779	.775	.778	**.788**
			LLM-Instruct	.723	.742	.732	.726	.743	**.751**
		long	LLM	.628	.630	.651	.644	**.653**	.652
			LLM-Instruct	.713	.724	.705	.713	.727	**.734**
	70b PT	short	F1	.818	.827	.822	.827	.829	**.836**
			LLM	.844	.852	.846	.847	.851	**.855**
			LLM-Instruct	.867	.875	.876	.881	**.885**	.881
		long	LLM	.704	.699	.719	.707	.705	**.724**
			LLM-Instruct	.789	.795	.776	.781	.788	**.812**
	IT	short	F1	.795	.813	.814	.809	.819	**.823**
			LLM	.836	.842	.842	.837	.844	**.845**
			LLM-Instruct	.850	.867	.866	.865	**.874**	.870
		long	LLM	.706	.706	.712	.715	**.721**	.715
			LLM-Instruct	.855	.850	.827	.842	**.861**	.851
State-Space	**7b** PT	short	F1	.598	.596	.585	.579	.583	**.612**
			LLM	.729	.737	.723	.721	.733	**.742**
			LLM-Instruct	.638	.640	.626	.621	.632	**.651**
		long	LLM	.613	**.627**	.612	.624	.620	.623
			LLM-Instruct	.606	.611	.601	.611	.618	**.633**
	IT	short	F1	.592	.603	.588	.581	.589	**.615**
			LLM	.737	.742	.730	.726	.740	**.744**
			LLM-Instruct	.632	.646	.625	.619	.637	**.653**
		long	LLM	.611	.617	.618	.612	**.625**	**.625**
			LLM-Instruct	.643	.652	.628	.628	.654	**.658**
Average				.712	.720	.711	.710	.718	**.729**

Table 4: **AUROC Evaluation.** Average AUROC across all datasets. The reference answer is generated using **beam search with 5 beams**, either as a whole sentence (*long*) or a short phrase (*short*).

Uncertainty measure generating scoring rule				*Logarithmic*					*Zero-One*
	Language Model	**Gen.**	**Metric**	**PE**	**LN-PE**	**SE**	**LN-SE**	**D-SE**	**G-NLL**
Transformer	**8B** PT	short	F1	.775	.791	.765	.787	.799	**.822**
		short	LLM	.700	.712	.686	.704	.713	**.726**
		long	LLM	.556	.540	.493	.520	.578	**.591**
	IT	short	F1	.778	.808	.805	.819	.811	**.845**
		short	LLM	.682	.704	.706	.713	.698	**.729**
		long	LLM	.535	.520	.584	.585	**.586**	.559
	70B PT	short	F1	.788	.799	.796	.812	.798	**.833**
		short	LLM	.700	.717	.719	**.727**	.718	.725
		long	LLM	.540	.552	.489	.531	.552	**.608**
	IT	short	F1	.756	.786	.796	**.806**	.788	.800
		short	LLM	.680	.697	.701	**.707**	.695	**.707**
		long	LLM	.534	.533	.544	.569	**.574**	.534
State-Space	**7b** PT	short	F1	.814	.818	.806	.823	.825	**.846**
		short	LLM	.703	.709	.699	.711	.712	**.719**
		long	LLM	.570	.595	.550	**.609**	.602	.563
	IT	short	F1	.799	.815	.794	.817	.828	**.845**
		short	LLM	.699	.713	.694	.709	.720	**.730**
		long	LLM	.574	.575	.582	**.621**	.607	.577
Average				.677	.688	.678	.698	.700	**.709**

C Detailed Quality of Estimators

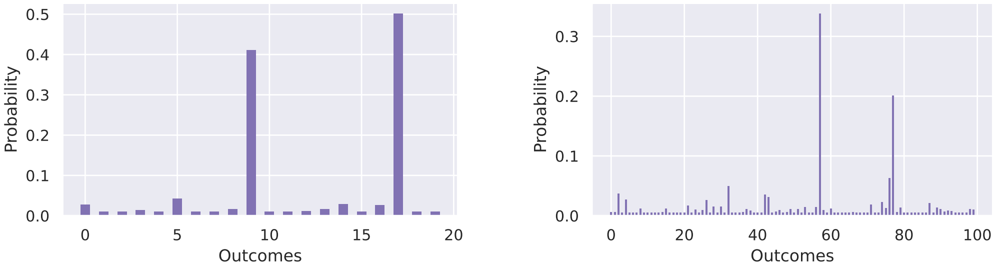
In this section, we provide detailed insights into the studies presented in Sec. 3.2, especially Fig. 1.

To recall, we empirically investigate the performance of estimators for the predictive entropy $\mathrm{H}(p(\boldsymbol{y} \mid \boldsymbol{x}))$ (Eq. (5)) and the maximum sequence likelihood $\max_{\boldsymbol{y}} p(\boldsymbol{y} \mid \boldsymbol{x})$ (Eq. (9)) in a controlled setting. Therefore, we consider a synthetic experiment with the following setup. We are given a space of possible outcomes $\mathcal{V}$ with $|\mathcal{V}| = \{20, 100\}$. The task is to predict a sequence $\boldsymbol{y} = (y_1, ...y_T) \in \mathcal{Y}_T$ where $y \in \mathcal{V}$ and T is 2, 3, or 4. Predictive distributions $p(\boldsymbol{y} \mid \boldsymbol{x})$ are not represented by a neural network, but are randomly sampled according to a Dirichlet distribution $\mathrm{Dir}(\{\alpha_1, ..., \alpha_{|\mathcal{V}|}\})$. The alpha parameters of the Dirichlet distribution are specified to yield typical predictive distributions as encountered in language models that follow a Zipf distribution. For $|\mathcal{V}| = 20$ we have $\alpha_{1,2} = 10$ and $\alpha_{3-20} = 0.2$. For $|\mathcal{V}| = 100$ we have $\alpha_{1,2} = 10$, $\alpha_{3-6} = 1$ and $\alpha_{7-100} = 0.2$. Note that the order of alpha values is randomly shuffled before drawing each predictive distribution. Representative predictive distributions sampled from this Dirichlet distribution are shown in Fig. 3a and Fig. 3b.

The experiments investigate the quality of the estimators depending on the number of samples $\{\boldsymbol{y}_n\}_{n=1}^N$. This is feasible because the ground truth values for both entropy and maximum sequence likelihood can be calculated for this small synthetic example through exhaustive enumeration. In the experiments we present in the appendix, we average over 1,000 runs, meaning that new sets of samples $\{\boldsymbol{y}_n\}_{n=1}^N$ are drawn to calculate the respective estimators. As beam search is deterministic, it does not vary in this experimental setting, compared to Fig. 1b in the main paper, where we investigated the quality of estimators for different $p(\boldsymbol{y} \mid \boldsymbol{x})$.

The results for estimating the entropy are shown in Fig. 4. We observe that the variance of estimators increases for larger vocabulary sizes $|\mathcal{V}|$ and sequence lengths T. Furthermore, lower temperatures decrease the variance of the estimator at the cost of introducing bias.

The results for estimating the maximum sequence likelihood are shown in Fig. 5. We observe that low-temperature multinomial sampling and beam search find the maximum sequence log-likelihood with a low number of samples with high probability. Greedy decoding (beam width of 1) finds the maximum for all experimental settings except one, where it takes a beam width of 2 to find it with high probability.



(a) Vocabulary size (width) = 20

(b) Vocabulary size (width) = 100

Figure 3: Exemplary predictive distributions $p(y_t \mid \boldsymbol{y}_{<t}, \boldsymbol{x})$ for different vocabulary sizes (widths).

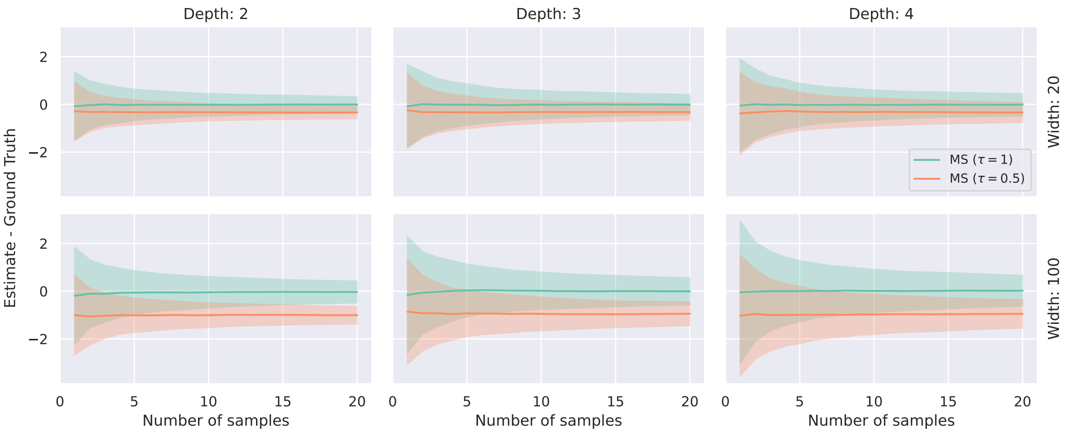


Figure 4: **Estimator of Predictive Entropy.** Results for different vocabulary sizes (width) and sequence lengths (depth). We estimate the entropy $\mathrm{H}(p(\boldsymbol{y} \mid \boldsymbol{x}))$ using N Monte-Carlo samples (cf. Eq. (5)). Lines denote the average over runs, while shades denote one standard deviation. We compare MS for two commonly used τ. The experiments show that the decreased temperature leads to lower variance but introduces bias.

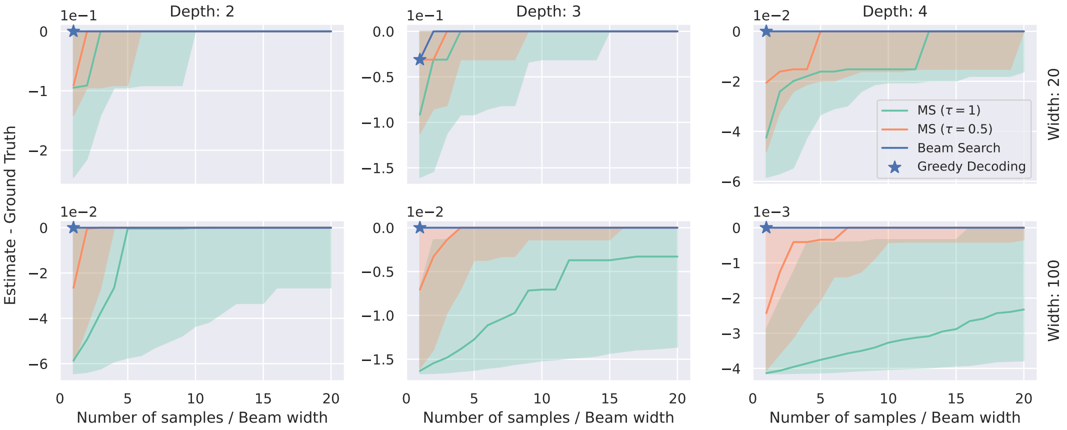


Figure 5: **Estimator of maximum sequence likelihood.** Results for different vocabulary sizes (width) and sequence lengths (depth). We estimate $\max_{\boldsymbol{y}} p(\boldsymbol{y} \mid \boldsymbol{x})$ using the maximum over N sampled obtained by beam search ($N = 1$ is greedy decoding) or MS with different τ. Lines denote the median, shades signify the possible values between the 5 and 95 percent quantile. Beam search is deterministic for any given distribution $p(\boldsymbol{y} \mid \boldsymbol{x})$. Even with a very low number of samples, low-temperature MS and beam search are able to find the maximum with high probability.