SPUQ: Perturbation-Based Uncertainty Quantification for Large Language Models

Xiang Gao, Jiaxin Zhang, Lalla Mouatadid, Kamalika Das
Intuit AI Research
2700 Coast Avenue, Mountain View, CA 94043
{xiang_gao, jiaxin_zhang, lalla_mouatadid, kamalika_das}@intuit.com

Abstract

In recent years, large language models (LLMs) have become increasingly prevalent, offering remarkable text generation capabilities. However, a pressing challenge is their tendency to make confidently wrong predictions, highlighting the critical need for uncertainty quantification (UQ) in LLMs. While previous works have mainly focused on addressing aleatoric uncertainty, the full spectrum of uncertainties, including epistemic, remains inadequately explored. Motivated by this gap, we introduce a novel UQ method, sampling with perturbation for UQ (SPUQ), designed to tackle both aleatoric and epistemic uncertainties. The method entails generating a set of perturbations for LLM inputs, sampling outputs for each perturbation, and incorporating an aggregation module that generalizes the sampling uncertainty approach for text generation tasks. Through extensive experiments on various datasets, we investigated different perturbation and aggregation techniques. Our findings show a substantial improvement in model uncertainty calibration, with a reduction in Expected Calibration Error (ECE) by 50% on average. Our findings suggest that our proposed UQ method offers promising steps toward enhancing the reliability and trustworthiness of LLMs[1].

1 Introduction

Large language models (LLMs) (Brown et al., 2020; Chowdhery et al., 2022; Touvron et al., 2023; OpenAI, 2022; Chung et al., 2022; OpenAI, 2023) have demonstrated remarkable success in various natural language processing tasks, such as text generation and question answering. However, a significant concern with LLMs is their proclivity to hallucinate, or make confidently wrong predictions (Maynez et al., 2020; Zhang et al., 2023; Ji et al., 2023; Chen et al., 2023), even for advanced models like GPT-4 (OpenAI, 2023). To address this issue, a robust approach to quantify uncertainty is necessary, enhancing the reliability and trustworthiness of these models (Si et al., 2022; Lin et al., 2022; Zhou et al., 2023; Kuhn et al., 2023). This is particularly important in online scenarios where the reference answers are not available, so that a response with a low confidence score can be treated with appropriate skepticism.

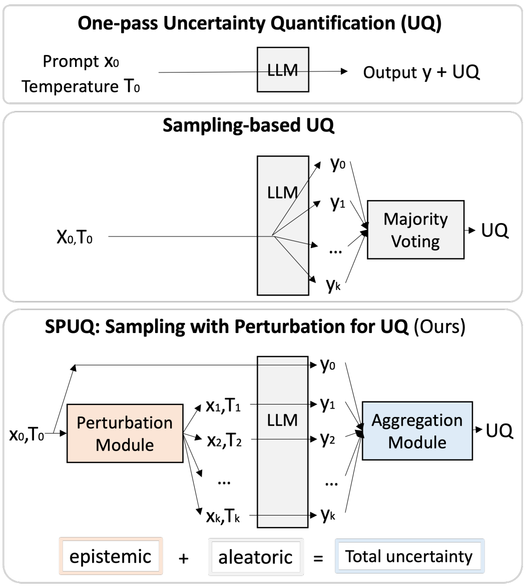

Figure 1: Overview of uncertainty quantification techniques: one-pass (Lin et al., 2022; Kadavath et al., 2022; Chen et al., 1998), sampling-based (Si et al., 2022; Wang et al., 2022), and our SPUQ method. SPUQ addresses both epistemic (via perturbation) and aleatoric (via sampling) uncertainties. Aggregation yields the total uncertainty, distinguishing SPUQ from traditional methods focused mainly on aleatoric uncertainty.

Uncertainty in machine learning models can be categorized into *aleatoric* (data-wise) and *epistemic* (model-wise) uncertainty (Hora, 1996; Hüllermeier and Waegeman, 2021). In the case of LLMs, *aleatoric* uncertainty arises from the inherent variability of natural language, where multiple or even an infinite number of valid outputs can exist for the same input. Existing work has proposed

[1] https://github.com/intuit-ai-research/SPUQ

Proceedings of the 18th Conference of the European Chapter of the Association for Computational Linguistics
Volume 1: Long Papers, pages 2336–2346
March 17-22, 2024 ©2024 Association for Computational Linguistics

methods relevant to aleatoric uncertainty. A common approach is to use the (normalized) generation likelihood (or the reciprocal of perplexity (Chen et al., 1998)) as an uncertainty measure. However, many popular LLMs, such as ChatGPT and GPT-4, do not provide access to token probabilities in their APIs, rendering this approach inapplicable. Additionally, sampling-based methods (Wang et al., 2022; Si et al., 2022; Kuhn et al., 2023) examine the deviation among multiple outputs generated from the original input. However, when an LLM makes a confidently incorrect prediction, resampling tends to yield similar results, leading to an overconfident and poorly calibrated score.

Epistemic uncertainty, on the other hand, is due to the lack of knowledge or suboptimal modeling. It captures the uncertainty that arises from limitations in the model's ability to learn from the data and generalize to new situations. Existing literature has linked epistemic uncertainty with adversarial examples (Tuna et al., 2022) and quantified it by introducing *perturbation* during inference time (Tuna et al., 2022; Gal and Ghahramani, 2016; Seeböck et al., 2019; Cremades Rey et al., 2019). However, there has been limited exploration of this uncertainty in the context of LLMs.

In this paper, we introduce a novel approach, Sampling with Perturbation for Uncertainty Quantification (SPUQ), as depicted in Fig.1. SPUQ addresses both *aleatoric* and *epistemic* uncertainties in LLMs by amalgamating existing methods that assess uncertainty from disparate angles. Our method tackles *epistemic* uncertainty through a perturbation module tailored for LLMs. This module, inspired by (Cremades Rey et al., 2019; Tuna et al., 2022), gauges model sensitivity to input perturbations. Coupled with this is our approach to *aleatoric* uncertainty, which adopts the principles of sampling methodologies (Wang et al., 2022; Si et al., 2022), enhanced by an aggregation module.

Within the perturbation module, we adjust temperature and/or prompts using a variety of techniques, such as paraphrasing, dummy tokens, and perturbed system messages. Our aggregation module not only generalizes the"exact match" method used in existing sampling approaches (Wang et al., 2022; Si et al., 2022) as a more general *inter-sample* matching method but also incorporates *intra-sample* metrics (Chen et al., 1998; Lin et al., 2022). In essence, SPUQ distinguishes itself from traditional sampling-based UQ methods through its innovative perturbation and aggregation modules. The former ensures a comprehensive treatment of epistemic uncertainty, while the latter adapts the method to a wider array of text generation tasks.

We perform thorough experimental studies on different perturbation and aggregation techniques using multiple question-answering datasets for various LLMs, such as GPT and PaLM series (Chowdhery et al., 2022). Through comparing our perturbation-based UQ method with existing baselines, we exhibit the efficiency of our approach in enhancing model uncertainty calibration, reducing Expected Calibration Error (ECE) by 50% on average.

In summary, our key contributions in this work are threefold: 1) We introduce a novel perturbation sampling-based uncertainty quantification (SPUQ) framework tailored for LLMs. This framework effectively addresses both aleatoric and epistemic uncertainties, leading to improved model uncertainty calibration. 2) We unify and generalize existing methods, integrating them into our perturbation and aggregation modules, making them adaptable to a broad range of LLM tasks. 3) We demonstrate significant improvement in uncertainty calibration through comprehensive experimental studies on multiple datasets across different LLMs.

2 SPUQ: Sampling with Perturbation for Uncertainty Quantification

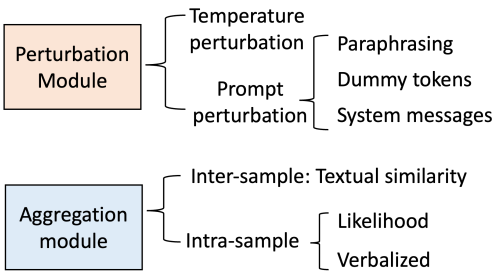

Figure 2: Options associated with the perturbation (Section 2.1) and aggregation modules (Section 2.2) of the SPUQ method.

In this section, we introduce the Sampling with Perturbation for Uncertainty Quantification (SPUQ) technique, as depicted in Fig.1 and elaborated in Algorithm 1. SPUQ operates on an LLM's original input temperature T_0 and prompt x_0. It derives perturbed variants (T_i, x_i) using the perturbation module and aggregates the outputs y_i from both the original and perturbed LLM inputs to compute a confidence score, c. This score rep-

Input x_i	**Prediction** y_i	**Likelihood**
Original		
Will Jay-Z reach the age of 60 before Kendrick Lamar?	No *(Incorrect)*	92%
Perturbed prompt		
Is it likely that Jay-Z will turn 60 before Kendrick Lamar does?	Yes	83%
Will Jay-Z hit the age of 60 before Kendrick Lamar does?	No	97%
Before Kendrick Lamar, will Jay-Z reach the age of 60?	Yes	94%
Is it possible that Jay-Z will turn 60 before Kendrick Lamar?	No	95%

Table 1: A motivating example from the StrategyQA dataset. GPT-3 confidently predicts an incorrect answer with a high likelihood (92%). In this case, uncertainty quantification relying solely on likelihood or re-sampling leads to overconfidence. However, input perturbation via paraphrasing reveals significant prediction instability, enabling the detection of epistemic uncertainties.

resents the quantified uncertainty, where, in our implementation, c ranges from 0 to 1; a higher c denotes reduced uncertainty. As previously highlighted, SPUQ addresses *epistemic* uncertainties via its perturbation module and *aleatoric* uncertainties through the sampling approach. We have proposed and evaluated diverse techniques for both the perturbation and aggregation modules, represented in Fig.2.

Algorithm 1 SPUQ

```
Require: Original temperature $T_0$, original prompt $x_0$, perturbation and aggregation module hyperparameters (Fig. 2)
Ensure: Confidence score $c$
Obtain $k$ perturbed inputs $(T_i, x_i)$ from the perturbation module, where $i = 1$ to $k$.
for $j = 0$ to $k$ do
    Send $(T_j, x_j)$ to LLM being tested
    Obtain output $y_j$ from LLM
end for
Send set $\{y_j\}$ to the aggregation module
Obtain the confidence score $c$
return $c$
```

2.1 Perturbation Module

To encapsulate epistemic uncertainties, our focus shifts towards understanding the susceptibility of model outputs to minor perturbations. Existing methods introduce perturbations using Monte Carlo Dropout (Seeböck et al., 2019; Tuna et al., 2022; Gal and Ghahramani, 2016). However, this approach is not feasible for closed LLM APIs. As an alternative, we introduce perturbations to the model inputs: the temperature and the prompt.

For temperature perturbation, we either adopt a temperature deviating from T_0 consistently across all inputs, or we sample temperatures randomly[2] to determine each y_i. When perturbing prompts, our objective remains consistent: introduce lexical variations without altering the core meaning. Our experiments encompass three strategies:

Paraphrasing We generate k paraphrased inputs, $\{x_i\}$ = paraphraser(x_0), for $i = 1, \ldots, k$, by querying ChatGPT (`gpt-35-turbo-v0301`) using the following prompt: " `Suggest {k} ways to paraphrase the text in triple quotes above. If the original text is a question, ensure your suggestions retain a question. Provide your response in JSON format: {"paraphrased": list of str}` " It's noteworthy that this procedure involves only a single LLM call to obtain all k paraphrased prompts.

Dummy Tokens We randomly select tokens, denoted by d, that marginally influence the original meaning and prepend or append them to x_0. Such tokens could be newline characters, tab spaces, ellipses, or supplementary punctuation marks like extra question marks for queries. The altered prompts can be described as $x_i = x_0 + d_i$ or $x_i = d_i + x_0$.

System Messages For chat-mode LLM, such as ChatGPT, GPT-4, and PaLM2-Chat, perturbations can be introduced not just to the user prompt—which includes the actual query—but also to the system message. Given an original system message like "`You are a helpful assistant`", we implement perturbations by replacing it with a randomly chosen message from a predefined set. Examples encompass phrases such as an empty system message, or semantically similar messages like "`You are a friendly assistant`", "`You are a question-answering assistant`", and

[2] For GPT series, we sample temperature uniformly from 0.0 to 2.0, and for PaLM series, the range is restricted from 0.0 to 1.0 by the API.

"You are a supportive question-answering assistant"[3].

2.2 Aggregation Module

The vanilla sampling techniques (Si et al., 2022; Wang et al., 2022) are proposed in scenarios where $\{y_i\}$ can be compared using the "exact match" criterion, which may not be suitable for a broader array of text generation tasks. To address this, we introduce an augmented method tailored for general language models, incorporating various aggregation methods (refer to Fig. 2) to derive the confidence score c without necessitating an exact match.

Inter-sample This approach revolves around the textual similarity among sample outputs $\{y_i\}$.

$$c_{\text{inter}} = \frac{\sum_{i=0,i\neq j}^{k} s(y_j, y_i) w_i}{\sum_{i=0,i\neq j}^{k} w_i} \tag{1}$$

Here, j signifies the index of the output being assessed for accuracy. The function $s(y_j, y_i)$ measures the textual similarity between outputs y_i and y_j, and w_i designates the weight allocated to the output variant y_i. Vanilla sampling's majority-voting approach (Si et al., 2022; Wang et al., 2022) becomes a specific instance of this formula if j is set to the most frequent answer, $s(y_j, y_i)$ is configured as the exact match function (yielding a value of 1 exclusively when y_i precisely aligns with y_j), and $w_i = 1$ uniformly for all i.

To calibrate uncertainty relative to the accuracy of original prompts, we assign $j = 0$. For perturbed prompts, we determine the weight w_i as $s(x_0, x_i)$, signifying the similarity between the perturbed prompt x_i and the original x_0. This configuration prioritizes milder perturbations over extreme ones, mitigating potential repercussions from severe perturbations, such as unsatisfactory paraphrasing.

Regarding the similarity function $s(\cdot, \cdot)$, our experiments span three metrics: `BERTScore`[4] (Zhang et al., 2019), cosine similarity derived from `SentenceBERT` embeddings[5] (Reimers and Gurevych, 2019), and the `RougeL Score`[6] (Lin, 2004).

Intra-sample The second category computes the average of the uncertainties discerned individually for each sample output, $c(x_i, y_i)$. We employ two strategies to obtain $c(x_i, y_i)$. First, we utilize likelihood (or the reciprocal of perplexity (Chen et al., 1998)). This approach, however, is confined to LLM APIs that grant access to the predicted token distribution. Subsequently, we adopt the verbalized uncertainty strategy posited by Lin et al. (2022). Post-generation of the output y_i, we prompt the LLM to articulate its uncertainty, either as words or as numbers, with the prompts listed in Table 4 in the *Appendix*.

$$c_{\text{intra}} = \frac{\sum_{i=0}^{k} c(x_i, y_i)}{k+1} \tag{2}$$

2.3 Selecting Hyperparamters

The optimal selection of specific perturbation and aggregation methods may vary depending on the dataset and LLM in use. We employ a development set to fine-tune these hyperparameters based on ECE and find that performance remains reasonably robust regardless of the development set chosen (refer to Table 2 for an example). Furthermore, through empirical investigations spanning five LLMs and four question-answering datasets, we note that certain perturbation and aggregation methods consistently yield better calibration results than others (refer to Section 4.2). This offers guidance on the recommended choices for these hyperparameters.

3 Experimental Setup

Large Language Models (LLMs) We conduct experiments using five LLMs[7]: GPT-3 (Brown et al., 2020) (`text-davinci-003`), ChatGPT (OpenAI, 2022) (`gpt-35-turbo-v0301`), GPT-4 (OpenAI, 2023) (`gpt-4`), PaLM2 (Chowdhery et al., 2022) (`text-bison`), and PaLM2-Chat (Chowdhery et al., 2022) (`chat-bison`).

Datasets We evaluate our methods on a summarization task dataset, XSUM (Narayan et al., 2018), and four question-answering datasets:

[3] Our current implementation and analysis may be somewhat constrained to these basic system messages. The ramifications of perturbations on more sophisticated and complex system messages warrant exploration in future works.

[4] https://huggingface.co/spaces/evaluate-metric/bertscore

[5] https://www.sbert.net

[6] https://github.com/google-research/google-research/tree/master/rouge

[7] GPT series are accessed via Azure OpenAI API. PaLM series are accessed via Google Cloud Platform API.

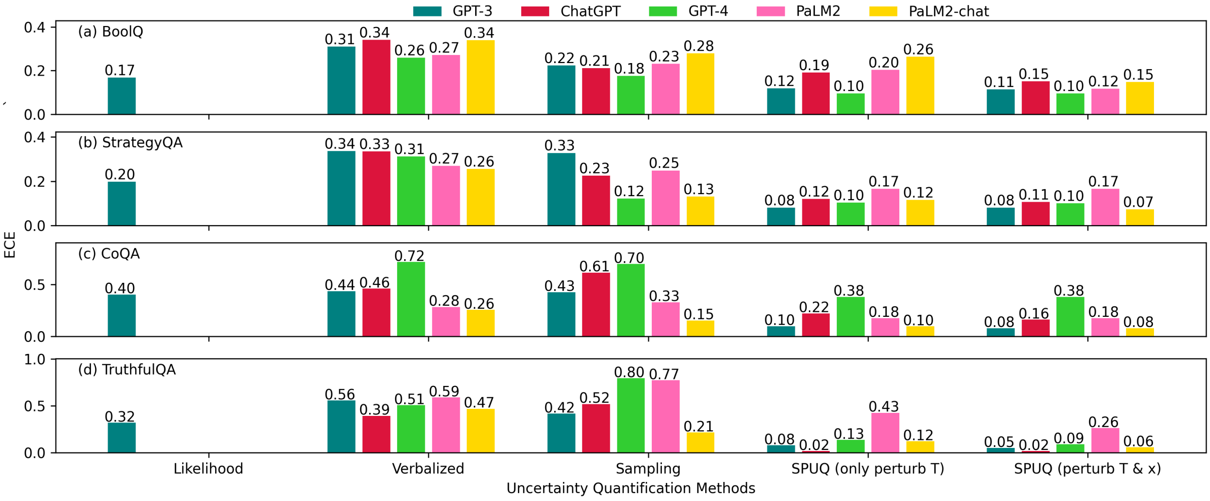

Figure 3: An overview of the uncertainty calibration performance, measured by the Expected Calibration Error (ECE), for various uncertainty calibration methods across five LLMs over four question-answering datasets. A lower ECE indicates better uncertainty calibration.

Dev set	Tuned SPUQ hyperparameters			Test performance	
	T perturbation	x perturbation	Aggregation	ECE ↓	Correlation ↑
I	+0.3	Paraphrasing	inter-sample, RougeL	0.082	0.690
II	+1.3	Paraphrasing	inter-sample, RougeL	0.102	0.670
III	+0.3	Paraphrasing	inter-sample, RougeL	0.082	0.690
V	+1.3	Dummy tokens	inter-sample, RougeL	0.103	0.672
IV	+0.3	Paraphrasing	inter-sample, RougeL	0.082	0.690

Table 2: Hyperparameters tuned on five separate development sets based on best ECE for GPT-4 on the TruthfulQA dataset and base value $T_0 = 0.7$. Test performance is reasonably robust to the choice of development set.

1. *Classification-type* datasets, which include two binary (yes/no) sets: StrategyQA (Geva et al., 2021) and BoolQ (Clark et al., 2019).

2. *Generation-type* datasets, featuring the CoQA (Reddy et al., 2019) and TruthfulQA (Lin et al., 2021) collections.

Baselines Our SPUQ method is benchmarked against several established baselines:

- Likelihood (Chen et al., 1998), where the confidence score is defined as the length-normalized LM likelihood. Only applicable to GPT-3 due to API constraints.

- Verbalized (Lin et al., 2022) method (refer to Table 4 for the prompts we employed).

- Sampling without perturbation (Si et al., 2022). To adapt this method to generation-type datasets, we substitute the exact match criterion with textual similarity[8].

Evaluation The calibration quality of the uncertainty is assessed using a set of metrics: Expected Calibration Error (ECE)[9] following (Si et al., 2022; OpenAI, 2023) and the Pearson's correlation (ρ) between confidence score c and accuracy. LLM accuracy is assessed via the "exact match" criterion for classification-type datasets, and F1 criterion for generation-type datasets following (Reddy et al., 2019). As mentioned in Section 2.3, hyperparameters for perturbation and aggregation may be tuned on a development set. To examine its sensitivity to development sets, for SPUQ, we report the average evaluation results of five tuning runs. For each run, hyperparameters are selected based on ECE on a development set of 30 randomly selected samples.

4 Results and Discussion

4.1 Enhanced Uncertainty Calibration

Overview Our results, depicted in Fig.3 and Table 3, highlight SPUQ's efficacy: it consistently

[8] For fair comparison, we tune on a development set to find the best text similarity metric for this baseline.

[9] ECE calculated as the mean of the absolute difference between each confidence bucket's accuracy and average confidence score c

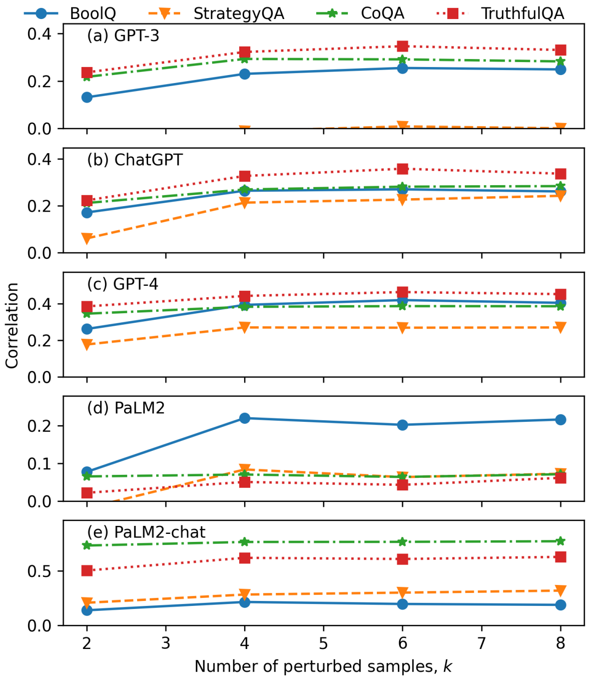

Figure 4: The dependence of the uncertainty calibration, measured by the average confidence-accuracy Pearson correlation, on the number of perturbed samples, k. The general trend indicates that calibration improves as k increases, but it plateaus approximately at k=5.

posts the lowest ECE across most tested language models and datasets, outperforming baselines with a 50% reduction on average (30% to 70% depending on the dataset and LLM). This superior calibration also aligns with the observed correlation between confidence and accuracy, detailed in Fig. 9 in the *Appendix*. The enhancement achieved by SPUQ, when compared to the sampling approach without perturbation, suggests that the improvement is primarily due to the perturbation module, specifically designed to address epistemic uncertainty. Notably, both temperature and prompt perturbations play significant roles in this enhancement.

LLM	Likelihood	Sampling	SPUQ
GPT-3	0.386	0.393	0.214
ChatGPT	-	0.406	0.209

Table 3: ECE on the XSUM summarization task.

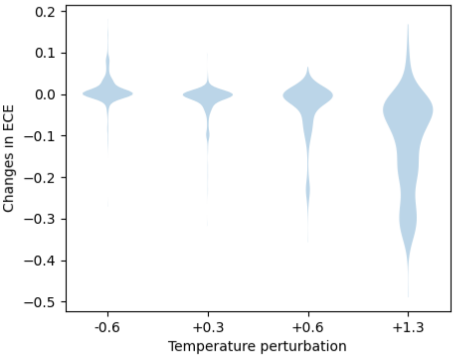

Figure 5: The distribution of ECE changes for specific temperature perturbations, taking into account variations in other hyperparameters. An increase in temperature (base value is $T_0 = 0.7$) during the sampling process tends to enhance calibration (decreased ECE).

A Case Study Table 1 elucidates the impact of perturbation. Occasionally, LLMs may produce confidently erroneous predictions. In the given example, there's a striking 92% likelihood of generating the incorrect response "No". Utilizing conventional sampling with unaltered input parameters (Si et al., 2022), the model typically mirrors the initial output, yielding an overconfident score close to 1, which results in poor uncertainty calibration. In contrast, paraphrasing the prompt brings forth epistemic uncertainty. Even minor changes can lead to starkly different outputs. For instance, in our example, the probabilities associated with "Yes" and "No" fluctuate markedly across the five y_i. Thus, SPUQ proves more discerning, delivering a confidence score of 0.50 since only half of the sampled y_i matches y_0

Mitigating Overconfidence Our case study highlights SPUQ's capacity to measure epistemic uncertainties by exploiting prediction instability through input perturbations. This is expected to temper the overconfidence frequently displayed by LLMs. Notably, confidence score distributions from SPUQ demonstrate a more even spread than those obtained via unperturbed sampling, as showcased in Figure 6.

4.2 Dependence on Hyperparameters

In this subsection, we delve into how calibration is influenced by various hyperparameters.

Number of Perturbed Samples Intuitively, increasing the number of samples should render the uncertainty quantification more consistent and trustworthy, eventually converging to a specific value. Our empirical findings affirm this hypothesis. As delineated in Fig. 4, the overarching trend signifies that calibration progressively refines with an upsurge in the number of perturbed samples k. However, the improvement begins to level off, approximately at $k = 5$.

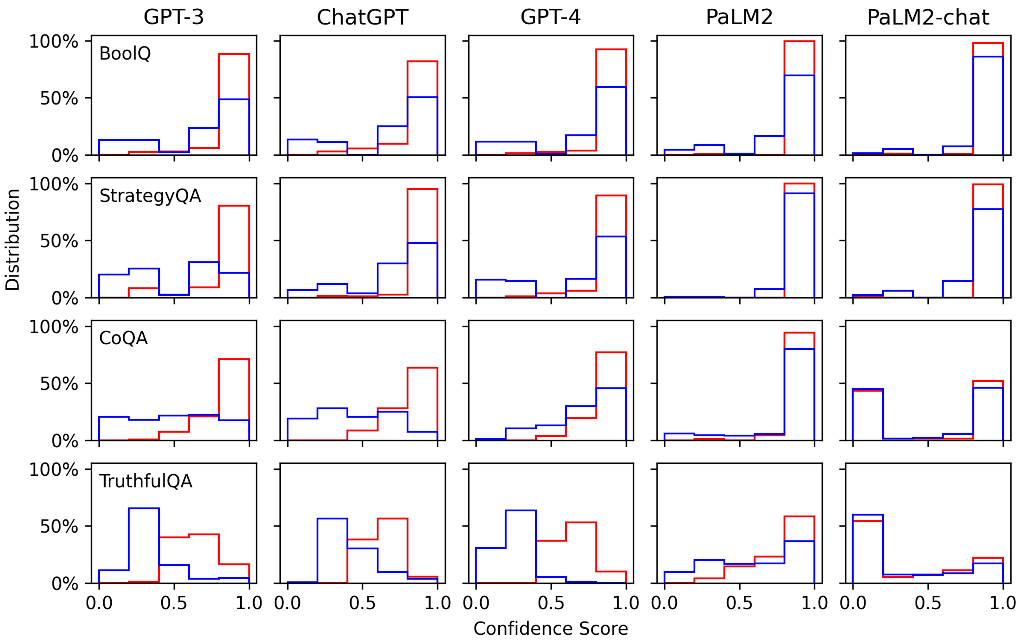

Figure 6: The empirical distribution of the confidence scores obtained using the conventional sampling (red) and our SPUQ (blue) approach. SPUQ displays a flatter distribution and less frequent over-confidence.

Temperature Perturbation By sampling at a heightened temperature, the predicted distribution leans towards uniformity, making the generated outcomes more disparate from the original output. Consequently, the sampling method becomes less skewed by overconfidence, enhancing uncertainty calibration. This dynamic is visually corroborated in Fig. 5. As the temperature during the sampling procedure escalates, ECE diminishes, signaling superior calibration. [10]

Prompt Perturbation While each of the three prompt perturbation methods we experimented with is designed to retain the original meaning, they manifest distinct perturbation characteristics. Paraphrasing can modify a significant portion of the input tokens. In contrast, dummy tokens and system messages do not make any changes to the question being asked. Fig. 7 shows that the paraphrasing method demonstrates superior calibration in more than half of the test cases, compared to the other prompt perturbation methods.

Aggregation Module While the inter-sample aggregation method has been employed in prior work (Si et al., 2022), we have further generalized it by incorporating textual similarity. Among the three similarity metrics assessed[11], the RougeL score consistently outperforms the other two, namely BERTScore and SentenceBERT, as shown in Fig. 8. Interestingly, our findings also indicate that the inter-sample aggregation isn't the sole viable approach; the intra-sample method also emerges as a compelling choice. No single aggregation method distinctly surpasses the others in all test scenarios (Refer to Fig. 10 in the *Appendix* for details). Consequently, we advocate for experimentation with both inter and intra-sample aggregation methods, rather than exclusively adhering to the inter-sample approach prevalent in existing literature.

Robustness to the Development Set As previously mentioned, we repeat the hyperparameter tuning process five times to assess its robustness to the choice of the development set. Notably, test outcomes remain robust across different development sets, with ECE standard deviation being roughly 10% of the mean—minor when contrasted with the 30% to 70% improvement. Table 2 offers a sample of these tuning results.

5 Related works

Hallucination Hallucination in LLMs is a significant challenge, as it can be induced by data, training, and inference processes (Ji et al., 2023).

[10] Our observations did not show improvements in calibration with random temperature perturbation, suggesting that directional adjustments (specifically increasing the temperature) are more effective.

[11] BoolQ and StrategyQA are only assessed by "exact match" as they are binary (Yes/No) question answering

Detecting hallucination on-the-fly remains a daunting task. Language models tend to over-commit to early mistakes, leading to more errors and contributing to hallucination snowballing (Zhang et al., 2023).

Improving Reliability Techniques to improve LLMs' reliability and reduce uncertainty have been proposed in the literature (Zhou et al., 2023). Wang et al. (Wang et al., 2022) found that sampling and aggregating multiple chain-of-thought reasoning paths can enhance LLMs' performance and reliability. Good in-context prompting strategies, such as few-shot prompting, can improve GPT-3's reliability(Si et al., 2022).

Uncertainty Quantification UQ in deep learning models has been explored using various techniques, such as Bayesian approximation and ensemble learning (Abdar et al., 2021; Malinin and Gales, 2020). LLMs are prompted to self-evaluate their previous predictions (Kadavath et al., 2022) or to express their uncertainty in natural language (Lin et al., 2022). On the other hand, sampling-based methods (Si et al., 2022) like Semantic Uncertainty (Kuhn et al., 2023) consider linguistic invariances when quantifying uncertainty.

6 Conclusion

We introduced the SPUQ method to enhance uncertainty calibration in LLMs, achieving a notable reduction in Expected Calibration Error by 30% to 70%. Our ablation study linked this improvement largely to our perturbation mechanism, underscoring its role in addressing epistemic uncertainty. The application of SPUQ offers a path to more reliable LLM outputs. Future work should expand SPUQ's applicability on beyond present datasets with simple prompts, exploring its effectiveness across diverse tasks and complex prompt structures.

Limitations

We experimented with datasets where accuracy can be assessed relatively easily with the reference answer. Future works are encouraged on tasks where accuracy is less well defined, such as in conversation and content generation. Our approach introduces additional computational costs due to multiple generations and/or paraphrasing, which may increase the latency of the output. The steps, however, can be parallelized to ensure $\mathcal{O}(1)$ complexity.

Ethics Statement

This work quantifies and reduces uncertainty in LLM outputs. It may help to reduce the generation and use of mistaken or misleading content from LLMs, to encourage a safer use of these models.

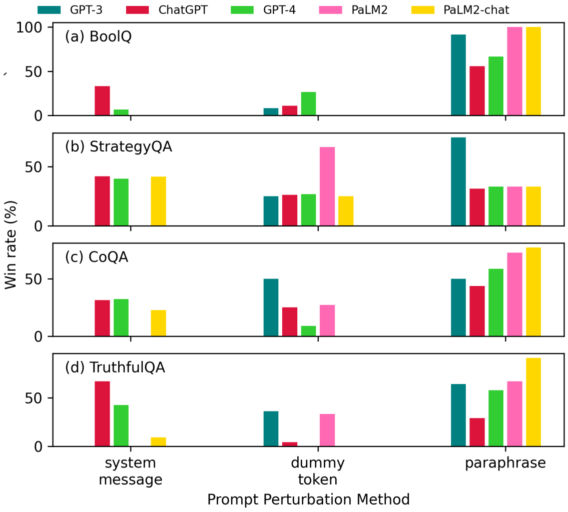

Figure 7: Dependence of uncertainty calibration on the prompt perturbation method. The "win rate" indicates the percentage to achieve the lowest ECE against others.

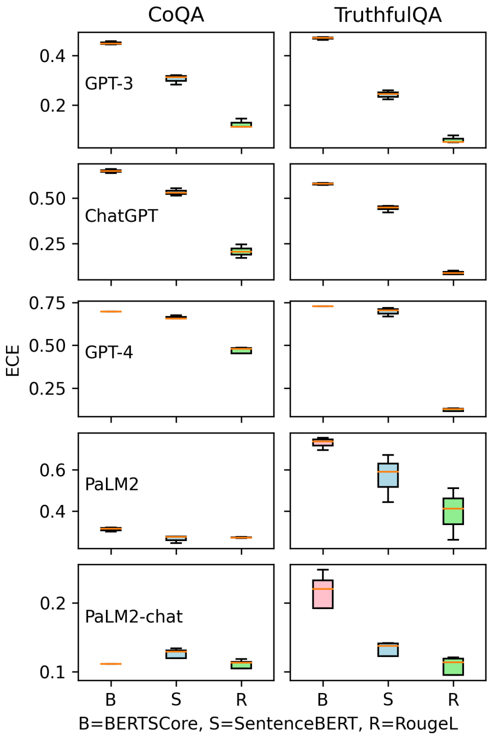

Figure 8: The distribution of ECE on various text similarity metrics employed by SPUQ.

References

Moloud Abdar, Farhad Pourpanah, Sadiq Hussain, Dana Rezazadegan, Li Liu, Mohammad Ghavamzadeh, Paul Fieguth, Xiaochun Cao, Abbas Khosravi, U Rajendra Acharya, et al. 2021. A review of uncertainty quantification in deep learning: Techniques, applications and challenges. *Information fusion*, 76:243–297.

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

Stanley F Chen, Douglas Beeferman, and Roni Rosenfeld. 1998. Evaluation metrics for language models.

Xuanting Chen, Junjie Ye, Can Zu, Nuo Xu, Rui Zheng, Minlong Peng, Jie Zhou, Tao Gui, Qi Zhang, and Xuanjing Huang. 2023. How robust is gpt-3.5 to predecessors? a comprehensive study on language understanding tasks. *arXiv preprint arXiv:2303.00293*.

Aakanksha Chowdhery, Sharan Narang, Jacob Devlin, Maarten Bosma, Gaurav Mishra, Adam Roberts, Paul Barham, Hyung Won Chung, Charles Sutton, Sebastian Gehrmann, et al. 2022. Palm: Scaling language modeling with pathways. *arXiv preprint arXiv:2204.02311*.

Hyung Won Chung, Le Hou, Shayne Longpre, Barret Zoph, Yi Tay, William Fedus, Eric Li, Xuezhi Wang, Mostafa Dehghani, Siddhartha Brahma, et al. 2022. Scaling instruction-finetuned language models. *arXiv preprint arXiv:2210.11416*.

Christopher Clark, Kenton Lee, Ming-Wei Chang, Tom Kwiatkowski, Michael Collins, and Kristina Toutanova. 2019. Boolq: Exploring the surprising difficulty of natural yes/no questions. *arXiv preprint arXiv:1905.10044*.

Luis F Cremades Rey, Denis F Hinz, and Mahdi Abkar. 2019. Reynolds stress perturbation for epistemic uncertainty quantification of rans models implemented in openfoam. *Fluids*, 4(2):113.

Yarin Gal and Zoubin Ghahramani. 2016. Dropout as a bayesian approximation: Representing model uncertainty in deep learning. In *international conference on machine learning*, pages 1050–1059. PMLR.

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

Stephen C Hora. 1996. Aleatory and epistemic uncertainty in probability elicitation with an example from hazardous waste management. *Reliability Engineering & System Safety*, 54(2-3):217–223.

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

Ziwei Ji, Nayeon Lee, Rita Frieske, Tiezheng Yu, Dan Su, Yan Xu, Etsuko Ishii, Ye Jin Bang, Andrea Madotto, and Pascale Fung. 2023. Survey of hallucination in natural language generation. *ACM Computing Surveys*, 55(12):1–38.

Saurav Kadavath, Tom Conerly, Amanda Askell, Tom Henighan, Dawn Drain, Ethan Perez, Nicholas Schiefer, Zac Hatfield-Dodds, Nova DasSarma, Eli Tran-Johnson, et al. 2022. Language models (mostly) know what they know. *arXiv preprint arXiv:2207.05221*.

Lorenz Kuhn, Yarin Gal, and Sebastian Farquhar. 2023. Semantic uncertainty: Linguistic invariances for uncertainty estimation in natural language generation. *arXiv preprint arXiv:2302.09664*.

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

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

Stephanie Lin, Jacob Hilton, and Owain Evans. 2022. Teaching models to express their uncertainty in words. *arXiv preprint arXiv:2205.14334*.

Andrey Malinin and Mark Gales. 2020. Uncertainty estimation in autoregressive structured prediction. *arXiv preprint arXiv:2002.07650*.

Joshua Maynez, Shashi Narayan, Bernd Bohnet, and Ryan McDonald. 2020. On faithfulness and factuality in abstractive summarization. *arXiv preprint arXiv:2005.00661*.

Shashi Narayan, Shay B Cohen, and Mirella Lapata. 2018. Don't give me the details, just the summary! topic-aware convolutional neural networks for extreme summarization. *arXiv preprint arXiv:1808.08745*.

OpenAI. 2022. ChatGPT.

OpenAI. 2023. Gpt-4 technical report. *arXiv*, pages 2303–08774.

Siva Reddy, Danqi Chen, and Christopher D Manning. 2019. Coqa: A conversational question answering challenge. *Transactions of the Association for Computational Linguistics*, 7:249–266.

Nils Reimers and Iryna Gurevych. 2019. Sentence-bert: Sentence embeddings using siamese bert-networks. *arXiv preprint arXiv:1908.10084*.

Philipp Seeböck, José Ignacio Orlando, Thomas Schlegl, Sebastian M Waldstein, Hrvoje Bogunović, Sophie Klimscha, Georg Langs, and Ursula Schmidt-Erfurth. 2019. Exploiting epistemic uncertainty of anatomy segmentation for anomaly detection in retinal oct. *IEEE transactions on medical imaging*, 39(1):87–98.

Chenglei Si, Zhe Gan, Zhengyuan Yang, Shuohang Wang, Jianfeng Wang, Jordan Boyd-Graber, and Lijuan Wang. 2022. Prompting gpt-3 to be reliable. *arXiv preprint arXiv:2210.09150*.

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

Omer Faruk Tuna, Ferhat Ozgur Catak, and M Taner Eskil. 2022. Exploiting epistemic uncertainty of the deep learning models to generate adversarial samples. *Multimedia Tools and Applications*, 81(8):11479–11500.

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

Muru Zhang, Ofir Press, William Merrill, Alisa Liu, and Noah A Smith. 2023. How language model hallucinations can snowball. *arXiv preprint arXiv:2305.13534*.

Tianyi Zhang, Varsha Kishore, Felix Wu, Kilian Q Weinberger, and Yoav Artzi. 2019. Bertscore: Evaluating text generation with bert. *arXiv preprint arXiv:1904.09675*.

Kaitlyn Zhou, Dan Jurafsky, and Tatsunori Hashimoto. 2023. Navigating the grey area: Expressions of overconfidence and uncertainty in language models. *arXiv preprint arXiv:2302.13439*.

A Appendix

In the appendix, we include the results for the aggregation method (Fig. 10), the overall evaluation using accuracy-confidence correlation (Fig. 9), and the prompts we used to obtain verbalized uncertainty (Table 4).

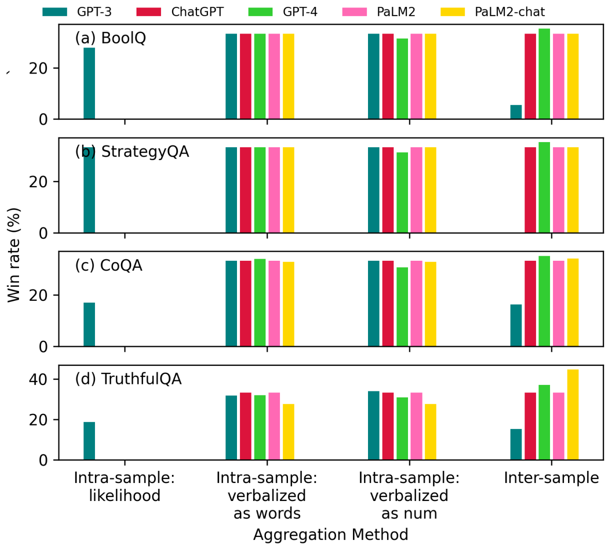

Figure 10: Dependence of uncertainty calibration on the aggregation method. The "win rate" indicates the percentage of instances a method achieves the lowest ECE against others.

Verbalized	LLM	Prompt
Words	chat	Your confidence is? (low, medium, high)
	text	Confidence (low, medium, high):
Numbers	chat	Your confidence is? (a score between 0.0 to 1.0)
	text	Confidence (a score between 0.0 to 1.0):

Table 4: Prompts employed to derive verbalized confidence Lin et al. (2022), used for aggregating intra-sample uncertainty. For uncertainty verbalized as words, we set $c = 0.25$ for "low", 0.5 for "medium", 0.75 for "high"

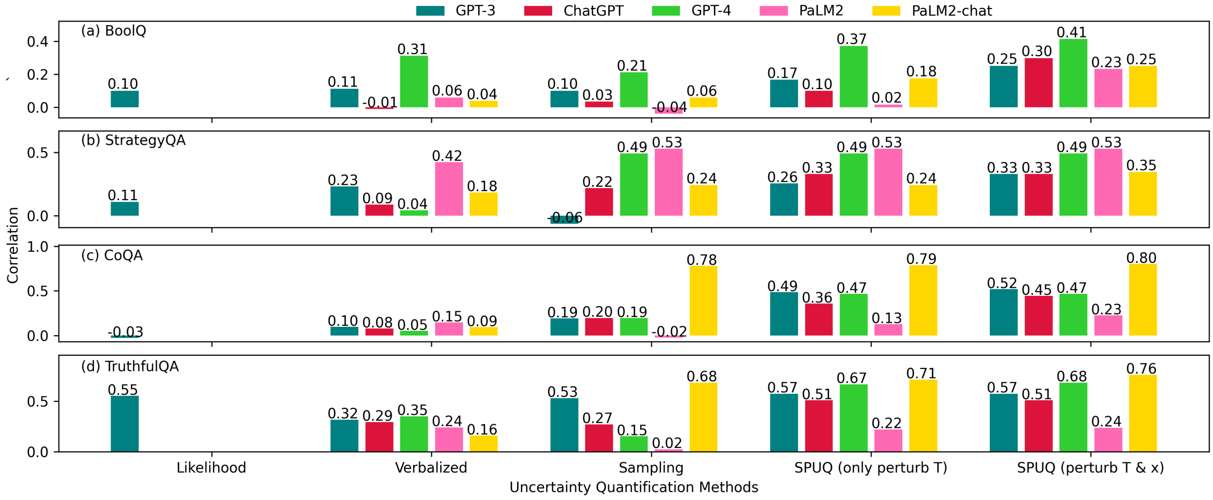

Figure 9: An overview of the uncertainty calibration performance, measured by, ρ, the Pearson's correlation between confidence score c and accuracy, for various uncertainty calibration methods across five LLMs over four question-answering datasets. A higher ρ indicates better uncertainty calibration. Our method, sampling with perturbation, exhibits the highest ρ for a given LLM in most cases.