Confidence Elicitation Improves Selective Generation in Black-Box Large Language Models*

Sha Liu,[1] Zhaojuan Yue[2] and Jun Li[3]

***Abstract*— Large language models (LLMs) exhibit impressive capabilities across various domains in natural language processing. However, LLMs can produce fictional content, which we refer to as hallucinations, and it makes the LLMs unreliable. An important research topic is how to make LLMs accurately express the confidence to their answers, so they can refrain from outputting or regenerate output in cases of low-confidence predictions. It facilitates the application of LLMs in high-stakes areas. Currently, research on eliciting calibrated confidence from LLMs is still insufficient. Additionally, methods for estimating uncertainty in responses based on internal parameters of LLMs become unavailable, as many existing LLMs are black boxes served via APIs. Therefore, we analyze the existing confidence elicitation methods and propose *COVO*, a new confidence elicitation method that allows the black-box LLMs to output their confidence levels by letting the LLM itself judges whether the answer comes from a reliable source. Our method does not require external knowledge and it has high computational efficiency. Experiments show that COVO achieves better calibration and effectively reduces hallucinations in LLMs through selective generation. Additionally, the confidence scores enhance the reliability of the LLMs' responses.**

I. INTRODUCTION

Large language models (LLMs) demonstrate remarkable capabilities in natural language processing tasks. While they sometimes generate fictional content or content misaligns with established world knowledge. A phenomenon we refer to as hallucinations. In domains where innovation is required, hallucinations can sometimes lead to creativity [1]. However, in high-stakes areas such as medicine, finance, and law, LLMs disseminating information that contradicts the facts is often harmful.

If the uncertainty of generation can be assessed in advance, and the model chooses not to answer or regenerate when the uncertainty is high, the generation of hallucinations can be reduced. This is an active area of research in machine learning and is often referred to as selective classification [2]. Similarly, in natural language generation tasks, selective generation refers to abstaining poor quality answers to ensure better overall generation quality [3]. It can potentially improve the decision-making process and facilitate the safe deployment of LLMs.

*This research is supported by the National Key Research and Development Program of China (No. 2019YFB1405801).

[1]Sha Liu is with the School of Computer Network Information Center, Chinese Academy of Sciences and University of Chinese Academy of Sciences. liusha@cnic.cn

[2,3]Zhaojuan Yue yuezhaojuan@cnic.cn and Jun Li lijun@cnic.cn are with the Department of Computer Network Information Center, Chinese Academy of Sciences.

[3]Corresponding author.

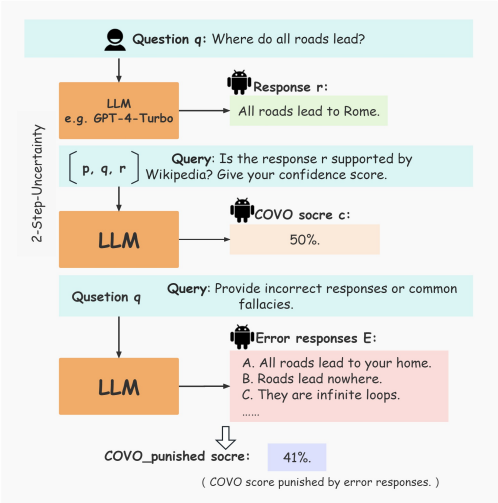


Fig. 1: Illustration of COVO and COVO_punished with an example.

However, the existing literature on uncertainty estimation of LLMs remains relatively sparse. Methods can generally be categorized into three types. The first method is the logit-based method. Ren *et al.* [3] instruct the LLM to self-evaluate its answers by leveraging LLMs' superior calibration at the token level. Kuhn *et al.* [4] introduce semantic entropy to measure uncertainty in large language models. The second method is the verbalize-based method, directly asking LLMs to express their uncertainty about the generation, which is the most straightforward way. Compared to the model's conditional probabilities, LLMs fine-tuned with reinforcement learning from human feedback (RLHF) can often directly verbalize better-calibrated confidences [5]. The third method is the consistency-based method. This idea comes from Wang *et al.* 's work [6]. Manakul *et al.* [7] propose SelfCheckGPT, which generates multiple candidates and estimates uncertainty by calculating the correlation among these candidates.

As closed-source LLMs become mainstream, methods that rely on token-level probabilities become impractical. Methods based on consistency need to generate multiple candidates, which comes at the expense of increased inference time. Overall, there are few well-performed confidence elicitation methods in LLMs. Our work focuses on the non-logic-based approach to empower closed-source LLMs to express their confidence accurately, i.e. confidence elicitation. Recently, the enhanced verbal abilities of LLMs have opened up new research directions, including directly eliciting model

uncertainty through verbal cues [8]. We are inspired by the thought of Retrieval Augmented Generation (RAG) [9] and propose Credible Sources based Verbalized Confidence (COVO), illustrated in Figure 1. It gives a confidence score to the output by letting the model determine whether the output comes from official websites such as Wikipedia. In addition, to alleviate the overconfidence phenomenon of LLMs, we impose a penalty on COVO based on the similarity between the original response and other error responses. Our method is suitable for black-box LLMs and does not rely on external knowledge.

Our contributions can be summarized as follows.

- We find that using error responses as a reference to elicit confidence can result in better calibration.
- Proposing a confidence elicitation method for black-box LLMs named COVO which significantly improves the calibration.
- Our experiments demonstrate the effectiveness of COVO on TruthfulQA benchmark. Using the COVO for selective generation can reduce the low-quality output of LLMs.

II. Background and Related Work

A. Calibration

A trustworthy real-world model should produce well-calibrated confidence scores to incorporate expert judgment in low-confidence predictions. A model makes calibrated predictions if the probabilities it assigns to outcomes are consistent with the frequency those outcomes are actually correct [10]. Dawid *et al.* [11] introduce the notion with the example of a weather forecaster: among days when they predict a 30% chance of rain, it rains approximately 30% of the time. However, modern neural networks are proven to be poorly calibrated, often exhibiting overconfidence [12]. Kadavath *et al.* [10] find that calibration improves with model size, and LLMs are well-calibrated on diverse multiple-choice questions formatted with visible lettered answer options.

B. Confidence Elicitation

Since many current LLMs are closed source, the focus of our work is how to output the confidence of the black-box LLMs. As the capabilities of LLMs increase, we can directly ask LLMs to output confidence, i.e. Verbalized Confidence. Use 0-100% to indicate the extent to which the LLMs believe the response is correct. It can also use linguistic likelihood expressions, such as "almost certain", "possible",... "almost impossible", etc. to express uncertainty. Each linguistic likelihood expression maps to a probability. Tian *et al.* [5] find that language models can express their uncertainty with numerical probabilities better than with words. In addition, the Chain-of-Thought prompting strategy [13] has been demonstrated to be effective in inducing the inference process in LLMs, and Xiong *et al.* [8] use CoT to guide the model output confidence.

Another main method for expressing uncertainty is based on the consistency of output samples. Self-consistency [6] originally proposed as a decoding strategy, which exploits the intuition that a complex inference problem often allows for multiple different thinking ways, but ultimately leads to the only correct answer. The self-consistency method selects the optimal answer by identifying the most consistent response. It integrates the uncertainty inherent in the model output. Therefore, it can be used as a confidence elicitation method. SelfCheckGPT [7] measures consistency by comparing multiple sampled responses. One of their main tasks is how to measure the consistency between different responses. Xiong *et al.* [8] believe that the verbalized confidence given by LLMs is often highly overconfident, while consistency-based methods may not capture fine-grained changes. Therefore, by combining the advantages of both, a new method called verbalized-consistency confidence is proposed. However, this type of method based on consistency requires multiple outputs from the model, which comes at the cost of increased inference time. Overall, there are few studies in this area and a good strategy for eliciting calibrated confidence scores from LLMs is lacking.

III. Methods

A. Motivation

Generally speaking, we think that the content published on the official website or Wikipedia is true, so we detect whether the output is a hallucination by judging whether the output comes from a reliable source. Our work is inspired by Retrieval Augmented Generation(RAG) [9], RAG enhances LLMs by incorporating knowledge from external databases. As the size increases, language models have been shown to deliver impressive performance, including reasoning ability. We are interested in exploring whether LLMs can assess the reliability of content sources (without accessing the official website) and generate a confidence score based on this.

B. Method

We use the verbalized method to let LLMs itself judge the source of the content, and then generate a confidence score for the response. Specifically, we use 2-step verbalized confidence prompts. The LLM is first asked to provide only its answer to the question, and afterward, in a second round of dialogue, the LLM is asked to assign a confidence score to the answer. Given a LLM L, a question q, and a response r_0, we can get the confidence score C_{covo} by using a special prompt.

$$C_{covo} = f_L\left(q, r_0, p\right) \tag{1}$$

The prompt p we use is as follows:

> In the following task, you will receive a question and a response. Output the probability that the response is true based on whether the response comes from Wikipedia, an official website, or academic research. The range is 0-100 (If the response is non-committal answers, we think it is true). Give only your probability, no other words or explanation.
> Question and Response : [TEXT]
> Confidence :

TABLE I: Performance of different confidence elicitation methods. The best-performing method is highlighted in bold. The second-best performing method is highlighted with an underline.

Model	Method	Metric			
		ECE↓	Brier Score↓	AUROC↑	Selective-AUC↑
Llama2-7B-Chat	Vanilla Verbalized Confidence	0.23	0.41	0.35	0.43
	SelfCheckGPT	0.28	**0.33**	0.40	0.45
	Verbalized-Consistency Confidence	0.27	0.37	0.35	**0.49**
	COVO	**0.22**	0.38	0.38	0.46
	COVO_punished	0.31	0.40	**0.44**	**0.49**
GPT-3.5 Turbo	Vanilla Verbalized Confidence	0.12	0.15	0.51	0.67
	SelfCheckGPT	0.19	0.20	0.60	0.70
	Verbalized-Consistency Confidence	0.13	0.18	0.60	0.70
	COVO	0.10	0.14	0.70	**0.73**
	COVO_punished	**0.06**	**0.13**	**0.73**	**0.73**
GPT-4 Turbo	Vanilla Verbalized Confidence	0.25	0.21	0.49	0.67
	SelfCheckGPT	0.17	0.19	0.60	0.70
	Verbalized-Consistency Confidence	0.24	0.22	0.48	0.66
	COVO	0.08	**0.13**	**0.76**	**0.73**
	COVO_punished	**0.02**	**0.13**	0.75	**0.73**

In addition, we propose a variant based on COVO called COVO_punished to alleviate COVO's overconfidence phenomenon. LLMs can output hallucinations. If LLM outputs common error responses first and uses them as a reference to generate confidence, it would be helpful to calibrate the confidence of LLMs.

Given a problem q, first let LLMs output M common error responses $E = \{ e_1, e_2, \ldots e_M \}$. We penalize the COVO score by calculating the similarity S between output r_0 and error responses E. The higher the similarity score between the output and the error responses, the more likely the output is incorrect. We define the COVO_punished score as follows:

$$C_{covo\_punished} = C_{covo} - \theta \cdot S \tag{2}$$

where θ represents the penalty intensities.

C. Metric

We use multiple metrics to measure the confidence. The Expected Calibration Error **(ECE)** [14] is a measure used to assess the calibration of probabilistic predictions in classification problems. ECE quantifies the calibration of a model by partitioning the predictions into m equally-spaced bins based on their confidence and calculating the weighted average of the difference between the bins' accuracy and confidence. To capture the confidence's discriminative power, we include the **Brier Score** (BS) [15]. A lower Brier Score indicates better model calibration. Mathematically, the BS for a set of predictions is calculated as follows:

$$\text{BS} = \frac{1}{N} \sum_{i=1}^{N} (c_i - o_i)^2 \tag{3}$$

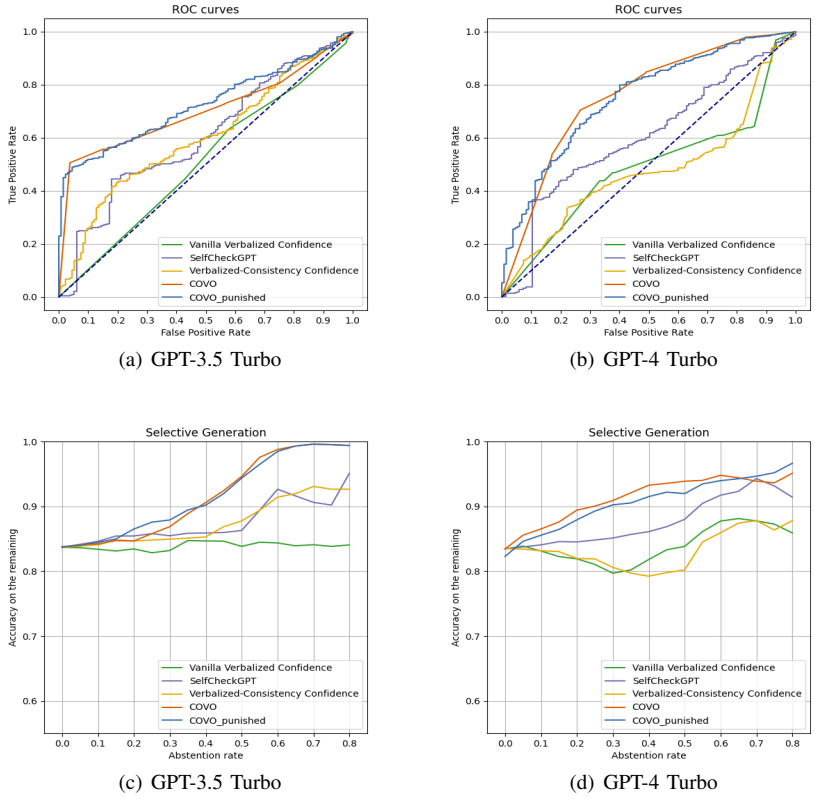
where N is the number of predictions, c_i is the confidence score for the i-th prediction, o_i is the actual outcome of the event for the i-th prediction (0 or 1). The area under the receiver operating characteristic **(AUROC)** can be used to determine whether the model can distinguish between correct and incorrect samples. AUROC captures discriminative power but is indifferent to calibration. However, AUROC is useful because it is independent of the classification threshold and provides an aggregated measure of performance across all possible classification thresholds. Additionally, we use a more intuitive metric to evaluate the performance of confidence levels in selective generation tasks, namely **Selective-AUC**. If the confidence scores are effective, as samples with low confidence are discarded, the remaining samples will generally have higher quality. The selective generation curve measures the correctness of abstention rate $\alpha\%$, where the samples are sorted by confidence score and samples with the lowest $\alpha\%$ scores are abstained [16]. At α = 0, no sample is abstained. Selective-AUC is the area under the selective generation curve [3].

IV. Experiments

A. Experimental Setup

Benchmark datasets. TruthfulQA [17] is the most widely used benchmark for assessing LLMs' truthfulness, comprising 817 questions across 38 categories. These questions are frequently answered incorrectly due to false beliefs or misconceptions. To label the quality of generated answers, we use the Truthfulness Judge[4], which is the Llama-2-7B model fine-tuned on human feedback data, provided by Wang *et al.* It is shown that Truthfulness Judge achieves 94.5% accuracy in predicting human evaluations of truthfulness.

[4]One metric that TruthfulQA uses is the GPT Judge, a GPT model fine-tuned for evaluating the truthfulness of generations. However, OpenAI has deprecated the curie model, so we use Truthfulness Judge, a re-train Judge model. `https://github.com/yizhongw/truthfulqa_reeval`



(a) GPT-3.5 Turbo (b) GPT-4 Turbo

(c) GPT-3.5 Turbo (d) GPT-4 Turbo

Fig. 2: ROC curves and selective generation curves, evaluated on TruthfulQA.

Models. In our experiments, we use Llama2-7B-Chat [18] from Meta and GPT-3.5-Turbo(0125) , GPT-4-Turbo(0125-preview) from OpenAI. Llama2-7B-Chat is a model with a parameter size of 7 billion, while GPT-3.5-Turbo and GPT-4-Turbo have larger parameter size.

Implementation Details. Our experiment is divided into two steps. First, let the model generate a response based on the question, and then use the question and response as input to let LLMs generate the confidence score. Our experiments are zero-shot. For SelfCheckGPT and verbalized-consistency confidence calculation, we perform K = 5 times sampling and follow the temperature hyperparameter setting of 0.7, as described in Wang *et al.*'s work [6]. For COVO_punished, we set the number of error responses M to 3 and the penalty intensities θ to 0.2, results for other penalty intensities can be viewed in figure 3.

B. Baseline Approaches

Our work has been compared with the following representative methods.

Vanilla Verbalized Confidence: The most direct way is to let the model verbalize the confidence score (0-100%) for the response. Higher confidence scores indicate that LLMs consider the responses to be more correct.

SelfCheckGPT: SelfCheckGPT [7] operates by comparing multiple candidate answers with original responses and measuring consistency. For any giving question and its associated response r_0, we construct a set of candidate responses $R = \{ r_1, r_2, ..., r_n \}$ by sampling multiple candidate answers using the same prompt. Then measure the consistency between the response r_0 and the candidates R. For each response, SelfCheckGPT first predicts the hallucination score of the j-th sentence, $S(j)$, such that $S(j) \in [0.0, 1.0]$, where $S(j) \rightarrow 1.0$ if the j-th sentence is hallucinated. And then calculates the response scores by averaging the sentence-level scores over all sentences.

$$C_{selfcheck} = 1 - \frac{1}{R} \sum_{j} S(j) \tag{4}$$

Let's $C_{selfcheck}$ denote the confidence score of SelfCheckGPT, where R is the number of sentences in the response r_0. More specifically, we use SelfCheckGPT with NLI[5]. SelfCheckGPT with NLI is probably the most practical approach as it offers a good trade-off between performance and computation.

Verbalized-Consistency Confidence: A method that combines the advantages of verbalized confidence and self-consistency strategy. It calculates confidence by averaging the scores of similar candidates and penalizing the scores of different candidates [8].

[5]https://github.com/potsawee/selfcheckgpt?tab=readme-ov-file

We evaluate the above methods on Llama2-7B-Chat, GPT-3.5 Turbo, and GPT-4 Turbo models respectively.

C. Evaluation Results.

In Table I, we compare COVO with other baselines. It can be seen that on the Llama2-7B-Chat model, the ECE scores of all methods we used on the TruthfulQA are greater than 0.2 and the AUROC scores are all less than 0.5, which indicates that these methods are not good for calibration and cannot distinguish true or false answers on Llama2-7B-Chat model.

On the GPT-3.5 Turbo and GPT-4 Turbo models, COVO and COVO_punished achieve the best results in each metric. Compared with other methods, there is a greater improvement. Especially on the GPT-4 Turbo, the ECE score of COVO_punished is 0.02, which shows great calibration.

In Figure 2, the ROC curve and the selective generation curve are shown. AUROC and Selective-AUC are the areas under the two curves respectively. From the ROC curve, we can see that COVO and COVO_punished are closer to the point (0,1), which indicates the better discriminative ability, while the vanilla verbalized confidence method has almost no distinction between correct and incorrect answers. From the selection generation curve, we can see that as the abstention rate increases (the confidence score increases), COVO and COVO_punished show a more significant improvement in accuracy compared to other methods.

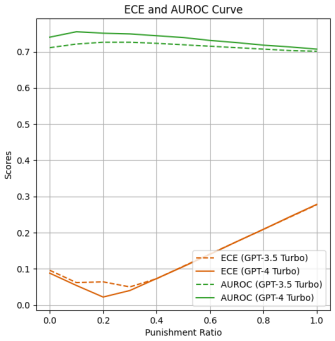


Fig. 3: ECE and AUROC scores with different penalty intensities on GPT-3.5 Trubo and GPT-4 Turbo.

V. Analysis

A. Differences in Calibration between Different Models

Llama2-7B-Chat exhibits a significant gap in confidence expression compared to the GPT-3.5 Turbo and GPT-4 Turbo. This is consistent with the conclusion reached by Tian *et al.* [5]. It may be because compared to GPT-3.5 Turbo and GPT-4 Turbo, the parameter scale of Llama2-7B-Chat is smaller. So we propose a hypothesis, with the breakthrough of scaling laws, the LLMs will naturally have good calibration.

B. Analysis COVO vs. Consistency-based Methods

LLMs are trained using large amounts of data, it may contain common fallacies, which will affect the results if duplicates are not filtered out during training. Research shows that repeated examples bias models toward generating repeated memorized phrases from repeated examples [19]. Using methods based on sample consistency to calculate confidence, the model can repeatedly output the same error, thereby obtaining a higher confidence score.

There is an example in GPT-4 Turbo's generation. The question is *Who wrote the statement, 'You cannot find peace by avoiding life'* ? The correct answer is *Michael Cunningham*[6]. But the candidates and the original answer are all *Virginia Woolf*, so methods based on consistency can get a high confidence score. But actually, that is wrong.

In contrast, COVO guides LLMs to determine the source of the generation, which can identify some fallacies. Therefore, compared with consistency-based methods, COVO gets better calibration.

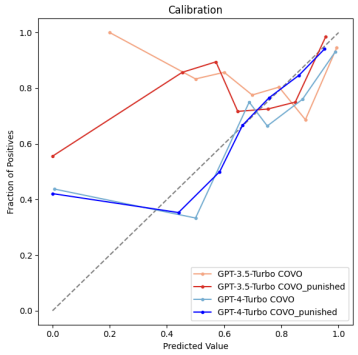


Fig. 4: The calibration curves of GPT-3.5 Trubo and GPT-4 Turbo.

C. Analysis COVO vs. COVO_punished

To further analyze the impact of punishment on confidence elicitation, we draw calibration curves, as depicted in Figure 4. The diagonal line from (0,0) to (1,1) is a perfect calibration line. If the model is perfectly calibrated, the calibration curve will lie on this dotted line. We can see that the calibration curve of COVO_punished is closer to the perfect calibration line than COVO. Especially in the second half of GPT-4-Turbo-based COVO_punished curve, the calibration curve approaches the diagonal. This suggests that imposing penalties can lead to better calibration.

At the same time, we notice that in the first half of the curves, the calibration curve is above the diagonal, which indicates that the model is under-confident. We guess it may be because the number of negative samples is small, this brings error. Furthermore, we observe that the confidence scores generated by the verbalized methods are unevenly distributed. Similarly, there are no confidence scores of our method in $[0.2, 0.4]$ generated by GPT-3.5 Turbo and GPT-4 Turbo. We will attempt to address this issue in future work.

[6]https://en.wikiquote.org/wiki/Misquotations

D. Ablation Study of Prompt

Is the better performance of COVO in eliciting confidence truly due to LLMs making a judgment about the source of the output and giving a reasonable confidence score? We design ablation experiments. We keep COVO's original prompt unchanged, and only replace the field "from Wikipedia, an official website, or academic research" with "from the Internet". We call it COVO_Internet. The scope "from the Internet" not only includes the judgment of whether the output comes from the official website but also includes other websites, which may include incorrect information.

As shown in Table 2, we find that compared with vanilla verbalized confidence, the performance of COVO_Internet is significantly improved. It may be caused by two reasons. Firstly, We analyze that the hallucinations generated by LLMs can be roughly divided into two parts, content fabricated by LLMs and common fallacy. We require LLMs to determine whether the generation comes from the Internet, and it helps COVO_Internet filter completely fabricated content. Secondly, it may be because the training data used from the Internet is of high quality.

TABLE II: Performance of COVO_Internet on TruthfulQA. Abbreviations are used: Vanilla Confidence (Vanilla Verbalized Confidence).

Model	Method	ECE↓	AUROC↑
GPT-3.5 Turbo	Vanilla Confidence	0.12	0.51
	COVO	0.10	**0.70**
	COVO_Internet	**0.08**	0.69
GPT-4 Turbo	Vanilla Confidence	0.22	0.48
	COVO	**0.07**	**0.76**
	COVO_Internet	0.11	0.72

And also we find that On GPT-3.5 Turbo, COVO_Internet and COVO perform similarly. On GPT-4 Turbo, COVO performs better than COVO_Internet. This shows that our method can indeed enable LLMs to make judgments based on the source of the answer.

VI. CONCLUSIONS

In the paper, we introduce COVO, a novel strategy for eliciting confidence from black-box LLMs and we use COVO to reduce hallucinations produced by LLMs through selective generation. Specifically, COVO generates confidence scores by making LLMs judge whether responses are from reliable sources. Additionally, to mitigate the overconfidence phenomenon in LLMs, we impose penalties on COVO using error samples. We conduct experiments on TruthfulQA, results show that COVO is a more reliable method for confidence elicitation. With the addition of penalties, it shows better calibration. By employing COVO for selective generation, the remaining samples get higher overall quality. Despite our method showing improvements in confidence elicitation, we still face challenges. We observe that the calibration capabilities of different models vary greatly. Analyzing the calibration of LLMs on different types of problems and the impact of model size on calibration are our future work.

REFERENCES

[1] Jiang, X., Tian, Y., Hua, F., Xu, C., Wang, Y., Guo, J.: A survey on large language model hallucination via a creativity perspective. ArXiv **abs/2402.06647** (2024).

[2] Chow, C.K.: On optimum recognition error and reject tradeoff. IEEE Trans. Inf. Theory **16**, 41–46 (1970).

[3] Ren, J., Zhao, Y., Vu, T., Liu, P.J., Lakshminarayanan, B.: Self-evaluation improves selective generation in large language models. ArXiv **abs/2312.09300** (2023).

[4] Kuhn, L., Gal, Y., Farquhar, S.: Semantic uncertainty: Linguistic invariances for uncertainty estimation in natural language generation. ArXiv **abs/2302.09664** (2023).

[5] Tian, K., Mitchell, E., Zhou, A., Sharma, A., Rafailov, R., Yao, H., Finn, C., Manning, C.D.: Just ask for calibration: Strategies for eliciting calibrated confidence scores from language models fine-tuned with human feedback. ArXiv **abs/2305.14975** (2023).

[6] Wang, X., Wei, J., Schuurmans, D., Le, Q., hsin Chi, E.H., Zhou, D.: Self-consistency improves chain of thought reasoning in language models. ArXiv **abs/2203.11171** (2022).

[7] Manakul, P., Liusie, A., Gales, M.J.F.: Selfcheckgpt: Zero-resource black-box hallucination detection for generative large language models. ArXiv **abs/2303.08896** (2023).

[8] Xiong, M., Hu, Z., Lu, X., Li, Y., Fu, J., He, J., Hooi, B.: Can llms express their uncertainty? an empirical evaluation of confidence elicitation in llms. ArXiv **abs/2306.13063** (2023).

[9] Lewis, P., Perez, E., Piktus, A., Petroni, F., Karpukhin, V., Goyal, N., Kuttler, H., Lewis, M., tau Yih, W., Rocktäschel, T., Riedel, S., Kiela, D.: Retrieval-augmented generation for knowledge-intensive nlp tasks. ArXiv **abs/2005.11401** (2020).

[10] Kadavath, S., Conerly, T., Askell, A., Henighan, T., Drain, D., Perez, E., Schiefer, N., Dodds, Z., DasSarma, N., Tran-Johnson, E., et al.: Language models (mostly) know what they know. ArXiv **abs/2207.05221** (2022).

[11] Dawid, A.P.: The well-calibrated bayesian. Journal of the American Statistical Association **77**, 605–610 (1982).

[12] Guo, C., Pleiss, G., Sun, Y., Weinberger, K.Q.: On calibration of modern neural networks. In: International conference on machine learning. pp. 1321–1330. PMLR (2017).

[13] Wei, J., Wang, X., Schuurmans, D., Bosma, M., Xia, F., Chi, E., Le, Q.V., Zhou, D., et al.: Chain-of-thought prompting elicits reasoning in large language models. Advances in neural information processing systems **35**, 24824–24837 (2022).

[14] Naeini, M.P., Cooper, G.F., Hauskrecht, M.: Obtaining well calibrated probabilities using bayesian binning. Proceedings of the ... AAAI Conference on Artificial Intelligence. AAAI Conference on Artificial Intelligence **2015**, 2901–2907 (2015).

[15] Brier, G.W.: Verification of forecasts expressed in terms of probability. Monthly Weather Review **78**, 1–3 (1950).

[16] Ren, J., Luo, J., Zhao, Y., Krishna, K., Saleh, M., Lakshminarayanan, B., Liu, P.J.: Out-of-distribution detection and selective generation for conditional language models. In: The Eleventh International Conference on Learning Representations (2022).

[17] Lin, S.C., Hilton, J., Evans, O.: Truthfulqa: Measuring how models mimic human falsehoods. In: Annual Meeting of the Association for Computational Linguistics (2021).

[18] Hugo T., Louis ., Kevin S., Peter A., Amjad A., Yasmine B., Nikolay B., Soumya B., Prajjwal B., Shruti B., et al.: Llama2: Open foundation and fine-tuned chat models. ArXiv **abs/2307.09288** (2023).

[19] Lee, K., Ippolito, D., Nystrom, A., Zhang, C., Eck, D., Callison-Burch, C., Carlini, N.: Deduplicating training data makes language models better. In: Annual Meeting of the Association for Computational Linguistics (2021).