Alleviating Hallucinations in Large Language Models with Skepticism Modeling

Anonymous ACL submission

Abstract

Hallucinations is a major challenge for large language models (LLMs), preventing adoption in diverse fields. Uncertainty estimation could be used for alleviating the damages of hallucinations. The skeptical emotion of human could be useful for enhancing the ability of self estimation. Inspirited by this observation, we propose a new approach called Skepticism Modeling (SM), which is formalized by combining the information of tokens and probabilities for self estimation. We construct the doubt emotion aware data, perform continual pre-training, and then fine-tune the LLMs, improve their ability of self estimation. Experimental results demonstrate this new approach effectively enhances a model's ability to estimate their uncertainty, and validate its generalization ability of other tasks by out-of-domain experiments.

1 Introduction

The generative nature of large language models (LLMs) bring the challenge of hallucination (Huang et al., 2023; Bai et al., 2024), namely their tendency to generate plausible-sounding but factually incorrect or nonsensical information. Hallucination hinders LLM's widespread adoption, particularly in domains that demand high levels of accuracy and expertise, like healthcare, legal sector and financial industry (Ji et al., 2023a; Zhang et al., 2023). Uncertainty estimation could be used for mitigating the damages of hallucinations (Huang et al., 2023). Previous studies directly used the model log-probabilities to estimate uncertainty, or used the tokens information to teach the model to express uncertainty (Huang et al., 2023). We propose a novel method that use both information for training, enhance the ability of self estimation, which is called Skepticism Modeling (SM).

Figure 1: Paradigm of Skepticism Modeling of LLM. The emojis represent self-skepticism level of the previous token.The strange phrases will arouse suspicion.

> The intense view of these manifold contradictions and imperfections in human reason has so wrought upon me, and heated my brain, that I am ready to reject all belief and reasoning, and can look upon no opinion even as more probable or likely than another [Treatise, 1.4.7.8]
>
> *David Hume*

Skepticism which means an attitude of doubt, plays a crucial role in human cognition, influencing information processing and decision-making. For instance, when people are asked "How many eyes does the finger have?" or "The capital of pigeon is ?", they will have skeptical emotion, which could be useful for identifying wrong question. When people are unfamiliar with a Mathematics question, skeptical emotion could also help them estimate the uncertainty of their own answer. The emotion-as-information theory (Schwarz and Clore, 1983) suggests that the skeptical feeling can lead to a more careful information examination. Studies have shown that skepticism is a core component of critical thinking (Facione, 1990) and important for meta-cognitive experiences (Koriat and Levy-Sadot, 1999). In fact, deep skepticism allows humans to question their own judgement, resisting the intense contemplation of the manifold contradictions and imperfections inherent in human belief and reasoning. As articulated by famously philosopher David Hume (Hume, 1978). Therefore, it is reasonable to implement LLM with skepticism

ability, which motivates this work.

In this paper, we propose an innovative paradigm to augment LLM with skepticism thinking ability. Firstly, we model the skepticism feeling as discrete tokens. Starting from a pre-trained LLM, we transform the skepticism token from the softmax probability associated with each preceding text token. We then redefine the sequence with each original text token followed by such a skepticism token. LLM learn such augmented text sequence by continual pre-training (CPT). We then conduct the supervise finetuning (SFT) stage, given question-answer samples with an extra rethinking question-answer pair similar with R-tuning (Zhang et al., 2024a). In the inference stage, the model can combine text tokens and softmax probability to enhance the ability of self estimation and alleviate the damages of hallucinations. Project code and model checkpoints can be found in https://anonymous.4open.science/r/SM-1E76.

In summary, our contributions are:

- We design a new modeling paradigm to make LLM have skepticism think, similar with humanity. By two stages' training, our LLM can self-evaluate its skepticism measures and provide more reasonable answer.
- We conduct substantial experiments to indicate our SM approach can achieve state-of-the-art (SOTA) performance in several QA benchmarks, with out-of-domain generalization abilities.
- We observe our SM approach have substantial robustness even given some unreasonable and implausible questions.

The rest of the paper is organized as follows. The connection with previous works is first discussed in Section 2. The SM methodology is stated in Section 3. Experiment results are summarized in Section 4. Finally Section 5 concludes this paper.

2 Related Works

2.1 Hallucination Detection

Many approaches in hallucination detection base on internal states (Azaria and Mitchell, 2023; Huang et al., 2023; Ling et al., 2024; Liu et al., 2024; Su et al., 2024). By analyzing the minimal token probability within key concepts, Varshney et al. assess the uncertainty of the model towards these concepts (Varshney et al., 2023). Our SM method also use token probability, however, we combine both token probability and token information to estimate uncertainty. Inspired by cognitive neuroscience, Zou et al. explore an approach to AI transparency named representation engineering (RepE), which provide traction on many problems including hallucination (Zou et al., 2023).

Finetuning LLMs can be useful for uncertainty estimation. Lin et al. train LLM to directly output verbalized probability with CalibratedMath, which is a suite of elementary mathematics problems. LLM's empirical accuracy on each type of question was used as the label (Lin et al., 2022). Our SM method also need to finetune the LLMs, but their method does not use token probability information, we combine both token probability and token information to finetune the LLMs. kadavath et al. add an additional value head to the LLMs, and finetuned the models to predict the probability that they can answer a question correctly (Kadavath et al., 2022). They only use question to train the model, we use both question and answer. And our SM method does not need to change the model structure.

Several studies probe the uncertainty of LLMs through their behavior (Huang et al., 2023; Lin et al., 2023; Hou et al., 2024; Yehuda et al., 2024). For instance, Manakul et al. propose SelfCheckGPT, a sample-based approach to detect hallucination via evaluating the consistency of multiple responses to the same prompt from a language model (Manakul et al., 2023). Motivated by truth-seeking mechanisms in law, Cohen et al. employ an "examiner" LM (EXAMINER). They leverage prompt to generate questions which are related with initial claim from LLMs, and discover factual inconsistencies among different answers (Cohen et al., 2023). Considering the reality that a single concept can be formulated in numerous ways, Farquhar et al. use semantic entropy to detect hallucinations by calculating uncertainty at the level of semantics rather than particular word sequences (Farquhar et al., 2024).

2.2 Hallucination Mitigation

Many methods have been proposed for hallucinations mitigation in LLM (Ji et al., 2023b; Dhuliawala et al., 2024; Zhang et al., 2024b,c) . No matter whether LLMs knows the knowledge or not, traditional fine-tuning approaches force LLMs to complete a sentence. If the question is beyond the inherent knowledge of LLMs, LLMs will try to fab-

ricate plausible-sounding but mistaken facts. Motivated by this, Zhang et al. propose a method called Refusal-Aware Instruction Tuning (R-Tuning), constructs a refusal-aware dataset by comparing the prediction and label, and then finetune LLMs to admit their uncertainty about the answer or refuse questions beyond its internal knowledge (Zhang et al., 2024a). Based on R-Tuning, our SM method combine token and token probability information to finetune LLMs.

RL finetuning can also mitigate hallucination (Roit et al., 2023; Sun et al., 2023), However, when face unfamiliar inputs, reward models may suffer from hallucinations. To tackle this challenge, Kang et al. propose a conservative reward models approach to avoid overestimate rewards for unfamiliar inputs. And then they use this approach to teach LLMs to generate reliable long-form responses on long text generation tasks (Kang et al., 2024).

Elaraby et al. explore teacher-student and knowledge injection methods to mitigate hallucinations in LLMs (Elaraby et al., 2023). Guan et al. presents Knowledge Graph-based Retrofitting (KGR), an approach that use knowledge graph to retrofit the initial responses of LLMs (Guan et al., 2024). Our SM method does not need external knowledge base.

3 Skepticism Modeling

In this section, we first introduce our Skepticism Modeling (SM) method , which integrates skeptical tokens into the vocabulary and includes three stages: continual pre-training, supervised finetuning and inference. Detailed framework of SM is visualized in Figure 2.

3.1 Modeling and Tokenization of Skepticism

Each skeptical emotion token corresponds to the generative probability about the previous generated token in the text. We first augment the tokenizer vocabulary with special tokens, $[< s_0 >, < s_1 > , \ldots, < s_9 >]$, indicating discretion of different skepticism levels. From "$< s_0 >$" to "$< s_9 >$", the skeptical emotion level is increasing which means the generative probability about the previous generated token is decreasing. To make the rule for converting the softmax probability to the skepticism token, we refer to the likelihood corresponding to adverbs indicating affirmation and doubt in natural language. For example, adverb "certainly" often indicates a probability greater than 0.8. We then use a special token "$< s_0 >$" to represent "certainly", if the softmax probability is more than 0.8, the skepticism token will be "$< s_0 >$", which means the lowest skeptical level. Adverb "probably" usually indicates a probability between 0.6 and 0.8, We then use a special token "$< s_1 >$" to represent "probably", if the softmax probability is between 0.6 and 0.8, the skepticism token will be "$< s_1 >$". We reformulate our tokenization with each normal text token followed by such a "skepticism token" (Figure 1).

Given a pretraining dataset, first we perform a forward pass of raw text corpus from a pretrained LLM, to obtain the token logits. Then we record the softmax probability for each token in the original corpus, discretize it and convert it into the ground truth skeptical token.

3.2 Continual Pre-Training

In this work, we conduct Continual Pre-Training (CPT) with model load from a pretrained LLM. By denoting the softmax probability of normal token as p, the softmax probability of skeptical token as s, and the previous inferenced probability is $\hat{l}$, our CPT loss can be expressed as

$$\mathcal{L}_{CPT} = -\frac{1}{T}\sum_{i=1}^{T}\log(p_i) \quad (1)$$

where T is the sequence length, i is the token position of either normal tokens or skeptical tokens.

3.3 Supervised Finetuning

During the Supervised finetuning(SFT) stage, we create our the refusal-aware data, in a similar process with R-tuning (Zhang et al., 2024a). Given a question-answer (QA) pair from SFT data, we first inference our CPT-version model, to obtain the probability of the original answer and determine our skepticism based on that result. We then augment the QA pair with another question "Are you sure you accurately answered the question based on your internal knowledge?" and the corresponding answer "I am sure/unsure." which is determined by the probability threshold. The probability thresholds perform a critical role which helps the LLM further align with the skeptical thinking.

We then perform the general Study by viewing p and s tokens as a uniform sequence. The SFT loss is

$$\mathcal{L}_{SFT} = -\frac{1}{T}\sum_{i=1}^{T_i}\log\left[Pr(y_{i+1}|\mathbf{x}_i, y_{1\ldots t}, \phi)\right] \quad (2)$$

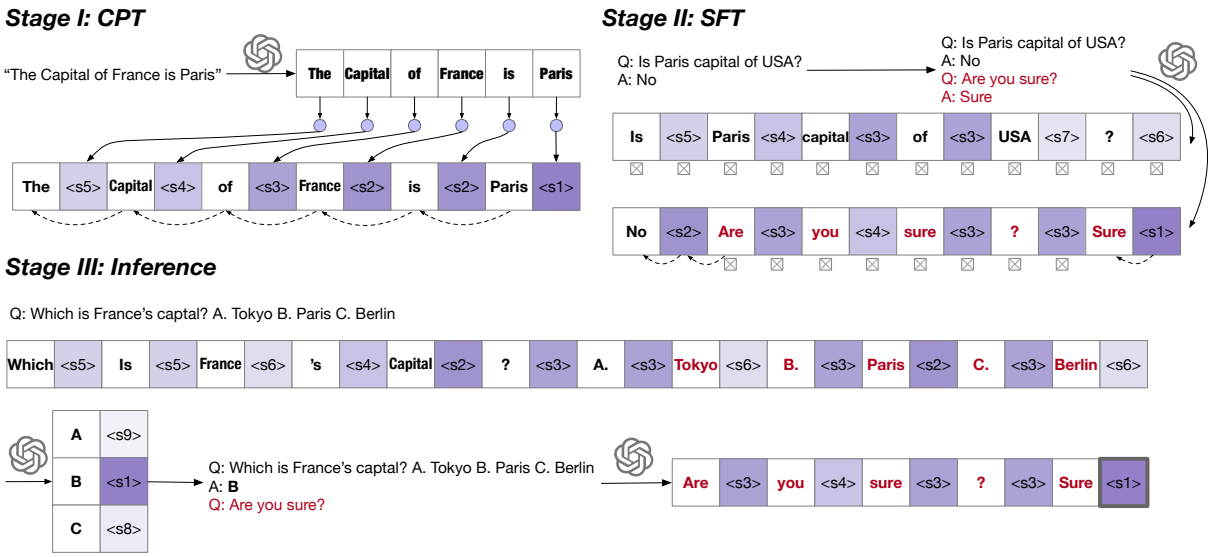

Figure 2: Detailed framework of SM. Stage I: first learn the plausibility of tokens from pretrained LLM, then continual pretraining on the corpus with vocabulary augmented with skepticism tokens. Stage II: augment the QA pair with the question 'Are you sure/unsure', inference the continual pretrained LLM to answer this augmented question, and finally finetune on these two QA pairs. Stage III: first inference on the finetuned LLM, get the most plausible answer, then concatenate with the augmented question, and inference the second time to obtain the skepticism probability.

where x is the question tokens, y is the answer tokens and T is the number of tokens in the response.

3.4 Inference

In the inference stage, we integrate skeptical tokens to the query and generate the response by our SFT version model. After that, we augment with the prompt "Are you sure you accurately answered the question based on your internal knowledge?" (Zhang et al., 2024a) then perform the second generation to obtain the final confidence.

Note that we employ the "sure" or "unsure" token as the indicator of uncertainty for the answer (Zhang et al., 2024a). The reason is when facing the open-domain question-answering task rather than multi-choice task, the answer contains multiple tokens instead of one choice token, which makes the uncertainty estimation complicated.

4 Experiments

In this section, we first introduce the training and evaluation datasets and tasks, then the comparable baselines, the evaluation methodologies, and the formal experiment results. We finally provide some typical cases to highlight our approach's ability.

4.1 Datasets

Table 1 list the sources and statistics of datasets used in our CPT and SFT stages.

4.1.1 CPT:

Datasets used in the CPT stage include Gutenberg books and wiki, which are from Dolma; as well as Opensubtitle, arxiv abstract and pubmed abstract, which are from Pile.

- **Dolma** (Soldaini et al., 2024): an open corpus of 3 trillion tokens for language model pretraining research. It encompass 5 billion documents range of sources from the web, scientific literature, code, public domain books, social media, and encyclopedias. With all the pretrained data and data curation toolkit open-sourced, it faciliatates the transparency and reproducibility of further research based on Dolma.

- **Pile** (Gao et al., 2020): a substantial corpus of English text, totaling 825 GiB. It composes 22 diverse and high-quality subsets, many of which are derived from academic or professional sources, such as PubMed Central, ArXiv, GitHub, and the US Patent and Trademark Office, among others. The construction of this dataset aims to address the growing need for data diversity in language modeling process.

4.1.2 SFT:

Datasets used in the SFT stage are natually classified into the following two categories:

- *Multiple-Choice*: Given a question with several choices, the model aims to chooses one correct option. We include **MMLU** (Hendrycks et al., 2020), **WiCE** (Kamoi et al., 2023), and **FEVER** (Thorne et al., 2018) in our experiments.
- *Question-Answering*: Given an open-domain question, the model directly generate its answer. Such type of datasets include **ParaRel** (Elazar et al., 2021) and **HotpotQA** (Yang et al., 2018).

For ease of performance comparison, we download the dataset from R-tuning (Zhang et al., 2024a) and keep the same in-domain and out-of-domain settings. For brevity, in the following context we use ID and OOD to denote in-domain and out-of-domain, respectively. We conduct sampling checks to determine if the dataset contains privacy and offensive information.

4.2 Baselines and tasks

We consider the following baselines:

- R-tuning: an instruction tuning approach that teaches large language models to identify and refrain from answering questions beyond their parametric knowledge, thereby mitigating the issue of hallucination and enhancing their ability to express uncertainty (Zhang et al., 2024a).
- VanillaFT: the vanilla approach which learns from the corpus in the conventional paradigm of LLM.

Similar with (Zhang et al., 2024a), two types of experiments, single-task and multi-task, can be analyzed. The single-task experiment studies the performance on the individual dataset, while multi-task experiment evaluates model generalization performance by training on mixture of datasets. Due to page limitations, here we only list results of multi-choice datasets. One can refer to Appendix to check results of Question-Answering datasets.

4.3 Evaluation

Models are measured with three metrics: accuracy, Average Precision (AP) score and Area Under the ROC Curve (AUC).

The accuracy is calculated as follows:

$$\text{accuracy} = \frac{\text{correctly answered questions}}{\text{all questions}}. \quad (3)$$

In the self-evaluation experiment, we first prompt the model to output an answer and then prompt it to provide its uncertainty. And we use AP score to evaluate the performance for uncertainty estimation.

The AP score is a way to summarize the precision-recall curve into a single value representing the average of all precisions. which is calculated as follows:

$$AP = \sum_{k=0}^{n-1} (R(k+1) - R(k)) \times P(k) \quad (4)$$

where n is the number of data, k is the number of data we select for the current threshold. P and R denote precision and recall. An ideal model predicts the correct answers with high confidence and the hallucinated wrong answers with relatively low confidence, leading to a high AP score.

AUC (Area Under the ROC Curve) is the area under the ROC (Receiver Operating Characteristic) curve used to measure the performance of a classifier. The closer the AUC value is to 1, the better the classifier performance; On the contrary, the closer the AUC value is to 0, the worse the classifier performance. We also use the ROC-AUC score to measure the performance for self-estimation. ROC depicts the performance of the classifier at different thresholds by taking the true positive rate (TPR) and the false positive rate (FPR) as the horizontal and vertical coordinates.

$$\text{TPR} = \frac{\text{TP}}{\text{TP + FN}} \quad (5)$$

$$\text{FPR} = \frac{\text{FP}}{\text{FP + TN}} \quad (6)$$

TP (True Positive) represents the number of correctly recognized positive cases. For example, when answer match label, the model output 'sure'. FN (False Negative) represents the number of incorrectly recognized positive cases as negative cases. FP (False Positive) represents the number of incorrectly identified negative examples as positive, while TN (True Negative) represents the number of correctly identified negative examples.

Stage	Datasets	Size	Format
CPT	gutenberg books	18G	Raw-Text
	wiki	16G	Raw-Text
	opensubtitle	0.5G	Raw-Text
	arxiv abstract	4G	Raw-Text
	pubmed abstract	1G	Raw-Text
SFT	MMLU (ID)	2439	Multiple-Choice
	MMLU (OOD)	9155	Multiple-Choice
	WiCE (Train)	3470	Multiple-Choice
	WiCE (Test)	958	Multiple-Choice
	FEVER (Train)	9999	Multiple-Choice
	FEVER (Test)	9999	Multiple-Choice
	ParaRel (ID)	5584	Question-Answering
	ParaRel (OOD)	13974	Question-Answering
	HotpotQA (Train)	10000	Question-Answering
	HotpotQA (Test)	7405	Question-Answering

Table 1: Details of Training Datasets. Sizes of CPT datasets is the file gigasizes, while sizes of SFT datasets are number of samples. SFT datasets are obtained from R-tuning (Zhang et al., 2024a)

Experiment	Stage	Parameters	Value
Single-Task	CPT	learning rate	5e-7
		weight decay	0.01
		batch size	1024
	SFT	learning rate	1e-6
		weight decay	0.01
		batch size	128
Multi-Task	CPT	learning rate	5e-7
		weight decay	0.01
		batch size	1024
	SFT	learning rate	1e-6
		weight decay	0.01
		batch size	128

Table 2: Hyper-parameters of experiments.

4.4 Implementation

We choose Qwen2-7B (Qwen Team, 2024) as the base models in our experiments, which licensed under the Apache License, Version 2.0. We use accelerator [1] and deepspeed [2] to conduct pretraining and instruction tuning, setting epoch to 1. All the experiments are implemented on Nvidia A100-80GB GPUs. Table 2 lists the hyperparameters of experiments.

[1] https://github.com/microsoft/DeepSpeed/blob/master/deepspeed/accelerator

[2] https://github.com/microsoft/DeepSpeed

4.5 Single-task Results

Table 3 lists the results of single-task experiments. The SM method demonstrates superior performance across most of the benchmarks, with seldom exceptions. Especially, SM is good at self-evaluation from the AP and AUC results, and also help the answering ACC from modeling of skepticism. Performance of SM is also robust since we consider both choice problems such as MMLU, WiCE, Fever, and question-answering tasks such as Parallel and HotpotQA. We also check the detailed results overn the ID and OOD domains for MMLU and Parallel.

Table 3 also lists results on the open-domain question-answering datasets, including Parallel and HotpotQA. Still, SM shows superiority comparing with two baselines, indicating that SM is able to build the skepticism on different scenarios and is robust to different test formats.

4.6 Multi-task Experiments

Table 4 lists the choice-problem results of multi-task experiments, also in terms of AP, AUC and ACC scores. Similar with the single task experiments, SM are also mostly the best, comparing with VanillaFT and R-tuning, except one or two exceptions. This result indicates that SM has good generalization and scaling abilities. By training with more datasets in different domains, one can expect that SM can align with their knowledge and emerge even better skepticism thinking.

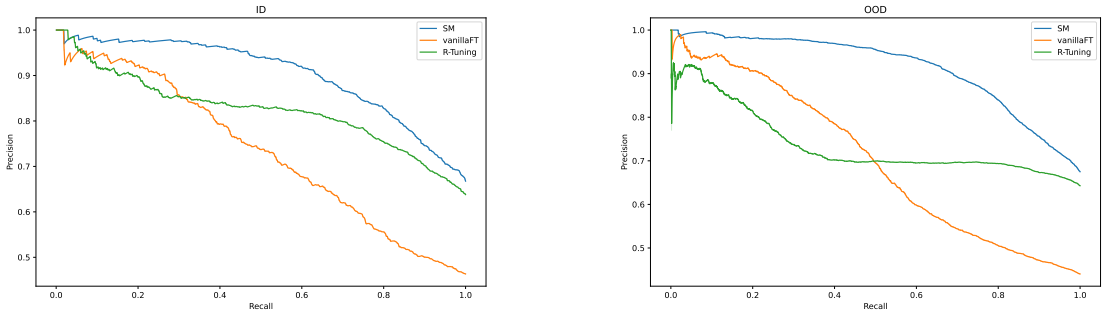

Figure 3: Multi-task Experimental Precision-Recall curves on MMLU, with ID and OOD domains.

Dataset	Domain	Metric	VanillaFT	R-tuning	SM
MMLU	ID	AP	37.04	86.89	**88.83**
		AUC	56.40	70.91	**73.12**
		ACC	68.63	**69.37**	69.00
	OOD	AP	37.71	85.78	**88.18**
		AUC	58.94	68.93	**72.33**
		ACC	69.10	68.92	**69.11**
WiCE	FULL	AP	67.14	**89.43**	85.79
		AUC	46.88	**80.15**	77.17
		ACC	29.59	**78.12**	67.35
Fever	FULL	AP	47.59	90.08	**96.99**
		AUC	63.28	73.37	**78.55**
		ACC	56.75	73.34	**91.64**
Parallel	ID	AP	59.25	**92.16**	86.52
		AUC	26.82	30.92	**64.95**
		ACC	24.02	29.33	**59.40**
	OOD	AP	57.08	**87.72**	64.95
		AUC	29.56	17.38	**45.97**
		ACC	19.31	11.47	**37.96**
HotpotQA	FULL	AP	61.55	**68.63**	63.95
		AUC	29.56	17.38	**45.97**
		ACC	19.31	11.47	**37.96**

Table 3: Single-task experiments of SM, R-tuning and VanillaFT on MMLU, WiCE, Fever, Parallel and HotpotQA datasets with AP, AUC and ACC scores (%). MMLU and Parallel are classified into ID and OOD domains, which denote in-domain and out-of-domain settings, respectively.

Dataset	Domain	Metric	VanillaFT	R-tuning	SM
MMLU	ID	AP	36.12	**87.45**	87.02
		AUC	57.37	**73.73**	71.06
		ACC	**69.13**	66.54	66.83
	OOD	AP	37.99	86.70	**86.88**
		AUC	59.08	69.66	**71.10**
		ACC	69.19	66.84	**68.84**
WiCE	FULL	AP	31.12	63.14	**67.62**
		AUC	42.6	45.38	**48.07**
		ACC	**36.32**	32.88	28.07
Fever	FULL	AP	59.94	87.43	**91.10**
		AUC	46.41	**77.61**	76.60
		ACC	35.83	74.38	**75.96**

Table 4: Multi-task experiments of SM, R-tuning and VanillaFT on MMLU, WiCE and Fever datasets with AP, AUC and ACC scores (%). MMLU results are classified into ID and OOD domains, which denote in-domain and out-of-domain settings, respectively.

Dataset	Domain	Metric	SM-noR	SM-noT	SM
MMLU	ID	AP	66.99	70.24	**69.55**
		AUC	64.21	67.21	**65.33**
		ACC	56.46	63.84	**59.32**
	OOD	AP	61.59	68.75	**74.11**
		AUC	53.50	57.56	**64.12**
		ACC	61.07	62.21	**64.89**

Table 5: Ablation results of SM on MMLU, comparing with SM-noR and SM-noT.

We also conduct multi-task experiments and exhibit the Precision-Recall curves on MMLU, with ID and OOD domains, respectively. As indicated by Figure 3, a higher AP score means better performance. This result indicates our model perform well in multi-task setting and show good generalization ability.

4.7 Abalation Study

To verify the effectiveness of each module, here theare also implement the following ablation approaches and compare with SM:

- SM-noR: our SM method without the replay mechanism. The "replay mechanism" here is different with replay mechanism in continual learning. It means to keep the inference ability for vanilla data, tenth training data are not processed with transition rule when we do the skepticism modeling.
- SM-noT: our SM method without the skepticism threshold.

Table 5 lists the ablation results. Result on the MMLU dataset reveals the full SM method's superiority over its variants, SM-noR and SM-noT, across various metrics. These results emphasize the effectiveness of the complete SM framework in uncertainty estimation, especially when generalizing to new domains.

4.8 Sensitivity Study

Since the skepticism threshold is a critical parameter in our approach, here we further conduct its sen-

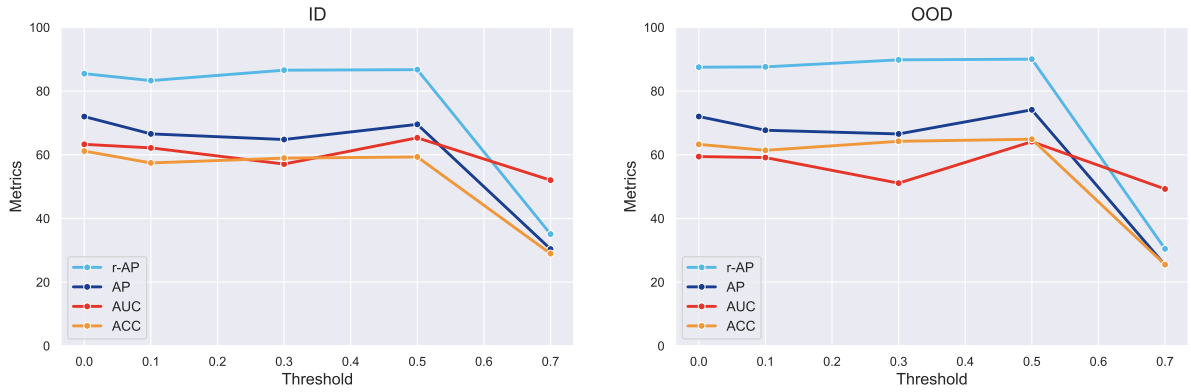

Figure 4: Sensitivity plots of MMLU metrics as functions of skepticism thresholds. Left: the ID domain; Right: the OOD domain.

sitivity analysis, as indicated in Figure 4. The sensitivity plots illustrate the performance of the MMLU metrics as functions of skepticism thresholds for both ID and OOD domains. A lower threshold may lead to more conservative predictions (higher skepticism), while a higher threshold results in more liberal predictions (lower skepticism). The peak of each curve indicates the threshold that yields the optimal metric score. Based on this analysis, We determined that the threshold of 0.5 strikes an optimal balance, offering the best trade-off between sensitivity and specificity for our model's skeptical estimation.

5 Conclusion

In this paper, we introduced a novel self-evaluation and self-justification method for large language models (LLMs) termed SM, by integrating the skepticism tokens and learning from reasoning process to enhance model's skeptical thinking ability. Our approach empowers LLMs to acknowledge their epistemic boundaries by responding with "I am unsure" when faced with questions beyond their knowledge boundary. This not only mitigates the risk of LLM hallucination but also fosters a more reliable interaction pattern with human users. Through extensive quantitative analysis, we demonstrated the superiority of our method across various data formats, domains and tasks, comparing with the vanilla fine-tuning method and R-tuning.

6 Limitations

The skepticism token models the self-skepticism level of the previous normal token. If the vocabulary size is small (less than 32000), the granularity of the tokenization results tends to be small, with many tokens representing individual letters, which is not conducive to the effectiveness of our method. The larger the vocabulary size, the greater the granularity of tokenization, and the easier it is for our method to take effect. And we do not provide strong justification for the choice of 10 skepticism levels, fewer or more levels setting could be better choices.

7 Ethics Statement

In this study, we have thoroughly evaluated the ethical implications of our research and anticipate no significant ethical concerns. All experiments have been conducted using publicly available datasets and pretrained model, mitigating potential ethical issues. Additionally, we have strictly adhered to all terms and conditions associated with the use of these datasets and pretrained model.

References

Amos Azaria and Tom Mitchell. 2023. The internal state of an llm knows when it's lying. *EMNLP 2023*, Findings of the Association for Computational Linguistics: EMNLP 2023:967–976.

Zechen Bai, Pichao Wang, Tianjun Xiao, Tong He, Zongbo Han, Zheng Zhang, and Mike Zheng Shou. 2024. Hallucination of multimodal large language models: A survey. *Preprint*, arXiv:2404.18930.

Roi Cohen, May Hamri, Mor Geva, and Amir Globerson. 2023. Lm vs lm: Detecting factual errors via cross examination. *Computing Research Repository*.

Shehzaad Dhuliawala, Mojtaba Komeili, Jing Xu, Roberta Raileanu, Xian Li, Asli Celikyilmaz, and Jason Weston. 2024. Chain-of-verification reduces hallucination in large language models. In *Findings of the Association for Computational Linguistics:*

ACL 2024, pages 3563–3578, Bangkok, Thailand. Association for Computational Linguistics.

Mohamed Elaraby, Mengyin Lu, Jacob Dunn, Xueying Zhang, Yu Wang, Shizhu Liu, Pingchuan Tian, Yuping Wang, and Yuxuan Wang. 2023. Halo: Estimation and reduction of hallucinations in open-source weak large language models. *CoRR*, abs/2308.11764.

Yanai Elazar, Nora Kassner, Shauli Ravfogel, Abhilasha Ravichander, Eduard Hovy, Hinrich Schutze, and Yoav Goldberg. 2021. Measuring and improving consistency in pretrained language models. *Transactions of the Association for Computational Linguistics*, 9.

Peter A Facione. 1990. Critical thinking: A statement of expert consensus for purposes of educational assessment and instruction. research findings and recommendations.

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

Leo Gao, Stella Biderman, Sid Black, Laurence Golding, Travis Hoppe, Charles Foster, Jason Phang, Horace He, Anish Thite, Noa Nabeshima, Shawn Presser, and Connor Leahy. 2020. The pile: An 800gb dataset of diverse text for language modeling. *CoRR*, abs/2101.00027.

Xinyan Guan, Yanjiang Liu, Hongyu Lin, Yaojie Lu, Ben He, Xianpei Han, and Le Sun. 2024. Mitigating large language model hallucinations via autonomous knowledge graph-based retrofitting. *THIRTY-EIGHTH AAAI CONFERENCE ON ARTIFICIAL INTELLIGENCE, VOL 38 NO 16*.

Dan Hendrycks, Collin Burns, Steven Basart, Andy Zou, Mantas Mazeika, Dawn Song, and Jacob Steinhardt. 2020. Measuring massive multitask language understanding. In *International Conference on Learning Representations*.

Bairu Hou, Yujian Liu, Kaizhi Qian, Jacob Andreas, Shiyu Chang, and Yang Zhang. 2024. Decomposing uncertainty for large language models through input clarification ensembling. *ICML*, abs/2311.08718.

Lei Huang, Weijiang Yu, Weitao Ma, Weihong Zhong, Zhangyin Feng, Haotian Wang, Qianglong Chen, Weihua Peng, Xiaocheng Feng, Bing Qin, and Ting Liu. 2023. A survey on hallucination in large language models: Principles, taxonomy, challenges, and open questions. *CoRR*, abs/2311.05232.

David Hume. 1978. *A Treatise of Human Nature*. Oxford University Press, Oxford. Revised P.H. Nidditch.

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

Ziwei Ji, Tiezheng Yu, Yan Xu, Nayeon Lee, Etsuko Ishii, and Pascale Fung. 2023b. Towards mitigating llm hallucination via self reflection.

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. 2022. Language models (mostly) know what they know. *Preprint*, arXiv:2207.05221.

Ryo Kamoi, Tanya Goyal, Juan Rodriguez, and Greg Durrett. 2023. Wice: Real-world entailment for claims in wikipedia. *Computing Research Repository*.

Katie Kang, Eric Wallace, Claire Tomlin, Aviral Kumar, and Sergey Levine. 2024. Unfamiliar finetuning examples control how language models hallucinate. *CoRR*, abs/2403.05612.

Asher Koriat and Ravit Levy-Sadot. 1999. Information-based and experience-based monitoring of one's own knowledge. *Dual-process theories in social psychology*, pages 483–502.

Stephanie Lin, Jacob Hilton, and Owain Evans. 2022. Teaching models to express their uncertainty in words. *Trans Mach Learn Res*, 2022.

Zhen Lin, Shubhendu Trivedi, and Jimeng Sun. 2023. Generating with confidence: Uncertainty quantification for black-box large language models. *arXivorg*, abs/2305.19187.

Chen Ling, Xujiang Zhao, Xuchao Zhang, Wei Cheng, Yanchi Liu, Yiyou Sun, Mika Oishi, Takao Osaki, Katsushi Matsuda, Jie Ji, Guangji Bai, Liang Zhao, and Haifeng Chen. 2024. Uncertainty quantification for in-context learning of large language models. *Preprint*, arXiv:2402.10189.

Linyu Liu, Yu Pan, Xiaocheng Li, and Guanting Chen. 2024. Uncertainty estimation and quantification for llms: A simple supervised approach. *Preprint*, arXiv:2404.15993.

Potsawee Manakul, Adian Liusie, and Mark Gales. 2023. Selfcheckgpt: Zero-resource black-box hallucination detection for generative large language models. *EMNLP 2023*, Proceedings of the 2023 Conference on Empirical Methods in Natural Language Processing:9004–9017.

Alibaba Group Qwen Team. 2024. QWEN2 TECHNICAL REPORT. Technical report, Alibaba Group.

Paul Roit, Johan Ferret, Lior Shani, Roee Aharoni, Geoffrey Cideron, Robert Dadashi, Matthieu Geist, Sertan Girgin, Léonard Hussenot, Orgad Keller, Nikola Momchev, Sabela Ramos, Piotr Stanczyk, Nino Vieillard, Olivier Bachem, Gal Elidan, Avinatan Hassidim, Olivier Pietquin, and Idan Szpektor. 2023. Factually consistent summarization via reinforcement learning with textual entailment feedback. *Computing Research Repository*, abs/2306.00186.

Norbert Schwarz and Gerald L Clore. 1983. Mood, misattribution, and judgments of well-being: Informative and directive functions of affective states. *Journal of personality and social psychology*, 45(3):513.

Luca Soldaini, Rodney Kinney, Akshita Bhagia, Dustin Schwenk, David Atkinson, Russell Authur, Ben Bogin, Khyathi Chandu, Jennifer Dumas, Yanai Elazar, Valentin Hofmann, Ananya Harsh Jha, Sachin Kumar, Li Lucy, Xinxi Lyu, Nathan Lambert, Ian Magnusson, Jacob Morrison, Niklas Muennighoff, Aakanksha Naik, Crystal Nam, Matthew E. Peters, Abhilasha Ravichander, Kyle Richardson, Zejiang Shen, Emma Strubell, Nishant Subramani, Oyvind Tafjord, Pete Walsh, Luke Zettlemoyer, Noah A. Smith, Hannaneh Hajishirzi, Iz Beltagy, Dirk Groeneveld, Jesse Dodge, and Kyle Lo. 2024. Dolma: an open corpus of three trillion tokens for language model pretraining research. *Preprint*, arXiv:2402.00159.

Weihang Su, Changyue Wang, Qingyao Ai, Yiran Hu, Zhijing Wu, Yujia Zhou, and Yiqun Liu. 2024. Unsupervised real-time hallucination detection based on the internal states of large language models. In *Findings of the Association for Computational Linguistics: ACL 2024*, pages 14379–14391, Bangkok, Thailand. Association for Computational Linguistics.

Zhiqing Sun, Sheng Shen, Shengcao Cao, Haotian Liu, Chunyuan Li, Yikang Shen, Chuang Gan, Liang-Yan Gui, Yu-Xiong Wang, Yiming Yang, Kurt Keutzer, and Trevor Darrell. 2023. Aligning large multimodal models with factually augmented rlhf. *arXiv (Cornell University)*, abs/2309.14525:13088–13110.

James Thorne, Andreas Vlachos, Christos Christodoulopoulos, and Arpit Mittal. 2018. Fever: a large-scale dataset for fact extraction and verification. *Proceedings of the 2018 Conference of the North American Chapter of the Association for Computational Linguistics Human Language Technologies, Volume 1 (Long Papers)*.

Neeraj Varshney, Wenlin Yao, Hongming Zhang, Jianshu Chen, and Dong Yu. 2023. A stitch in time saves nine: Detecting and mitigating hallucinations of llms by validating low-confidence generation. *Preprint*, arXiv:2307.03987.

Zhilin Yang, Qi Peng, Saizheng Zhang, Yoshua Bengiov, William W. Cohent, Ruslan Salakhutdinov, and Christopher D. Manning. 2018. Hotpotqa: A dataset for diverse, explainable multi-hop question answering. *Computing Research Repository*, pages 2369–2380.

Yakir Yehuda, Itzik Malkiel, Oren Barkan, Jonathan Weill, Royi Ronen, and Noam Koenigstein. 2024. InterrogateLLM: Zero-resource hallucination detection in LLM-generated answers. In *Proceedings of the 62nd Annual Meeting of the Association for Computational Linguistics (Volume 1: Long Papers)*, pages 9333–9347, Bangkok, Thailand. Association for Computational Linguistics.

Hanning Zhang, Shizhe Diao, Yong Lin, Yi R. Fung, Qing Lian, Xingyao Wang, Yangyi Chen, Heng Ji, and Tong Zhang. 2024a. R-tuning: Instructing large language models to say 'i don't know'. *NAACL-HLT*, pages 7113–7139.

Shaolei Zhang, Tian Yu, and Yang Feng. 2024b. TruthX: Alleviating hallucinations by editing large language models in truthful space. In *Proceedings of the 62nd Annual Meeting of the Association for Computational Linguistics (Volume 1: Long Papers)*, pages 8908–8949, Bangkok, Thailand. Association for Computational Linguistics.

Xiaoying Zhang, Baolin Peng, Ye Tian, Jingyan Zhou, Lifeng Jin, Linfeng Song, Haitao Mi, and Helen Meng. 2024c. Self-alignment for factuality: Mitigating hallucinations in LLMs via self-evaluation. In *Proceedings of the 62nd Annual Meeting of the Association for Computational Linguistics (Volume 1: Long Papers)*, pages 1946–1965, Bangkok, Thailand. Association for Computational Linguistics.

Yue Zhang, Yafu Li, Leyang Cui, Deng Cai, Lemao Liu, Tingchen Fu, Xinting Huang, Enbo Zhao, Yu Zhang, Yulong Chen, Longyue Wang, Anh Tuan Luu, Wei Bi, Freda Shi, and Shuming Shi. 2023. Siren's song in the ai ocean: A survey on hallucination in large language models. *CoRR*, abs/2309.01219.

Andy Zou, Long Phan, Sarah Chen, James Campbell, Phillip Guo, Richard Ren, Alexander Pan, Xuwang Yin, Mantas Mazeika, Ann-Kathrin Dombrowski, Shashwat Goel, Nathaniel Li, Michael J. Byun, Zifan Wang, Alex Mallen, Steven Basart, Sanmi Koyejo, Dawn Song, Matt Fredrikson, J. Zico Kolter, and Dan Hendrycks. 2023. Representation engineering: A top-down approach to ai transparency. *CoRR*, abs/2310.01405.