Knowledge Introspection: A Self-reflection Method for Reliable and Helpful Large Language Models

Anonymous ACL submission

Abstract

Large Language Models (LLMs) have inherent knowledge deficiency due to insufficient or erroneous data and incomplete training strategies. Furthermore, LLMs are often overconfident and unaware of their own knowledge deficiency, which will pose safety and legal risks to users. Inspired by the process of human introspection, we propose a two-stage method that enables LLMs to master the capability of knowledge introspection. Our method relies on data only generated by the LLM itself, and makes the LLM distinguish among what is known, uncertain and unknown. The method is trained in two-stages, in which supervised fine-tuning is employed in the first stage and direct preference optimization is utilized in the second stage. Experimental results demonstrate that our method effectively enhances the LLM's understanding of its internal knowledge, significantly improves generation accuracy, reliability and helpfulness of the model responses.

1 Introduction

Large Language Models (LLMs), including notable ones like GPT-4 (OpenAI, 2023), Claude (Anthropic, 2023) and LLaMA (Touvron et al., 2023), have shown impressive general capabilities, attributed to pre-training on large-scale corpora. Pre-training data serves as the foundation for LLMs, allowing them to acquire factual knowledge (Huang et al., 2023; Zhou et al., 2023). However, the absence of knowledge in specific scenarios, erroneous information in pre-training data and flawed pre-training strategies will lead to knowledge deficiency (Huang et al., 2023).

Knowledge deficiency may cause LLMs to provide unreliable responses, posing risks to users especially in professional fields such as medical and law scenarios, where misleading answers can lead to serious consequences. Even GPT-4 still provides incorrect answers (Xu et al., 2024) and may pose safety or legal risks to users . Conventional methods attempted to select high quality pre-training data (Chen et al., 2023) or utilize RAG methods (Peng et al., 2023; Gao et al., 2024). Recently, Cheng et al. (2024) attempted to teach LLMs to say 'I don't know' thereby avoiding the responses of incorrect knowledge. Compared to methods like RAG or selecting data, teaching LLMs to say 'I don't know' doesn't require external knowledge and can be easily extended to a variety of tasks. However, this kind of method usually leads to LLMs being overly conservative and frequently refusing to provide relevant information even when they have the knowledge. They only consider reliability but the helpfulness would be much decreased.

Introspection (or self-reflection) (Myers, 1986) is a process of internal communication with oneself, offering clear insights and shaping actions in future situations. Inspired by introspection, we introduce Knowledge Introspection to help LLMs assess their own knowledge and understand their mastery level. The LLM is trained with supervised fine-tuning (SFT) and direct preference optimization (DPO) and learns to generate expression indicating the degree of knowledge mastery. We use the data only generated by the LLM itself, namely knowledge introspection (KIs) dataset, for training. Experimental results demonstrate that, without injecting any external knowledge, the LLM has gained a clearer understanding of its own knowledge scope. The LLM can inform users of its knowledge mastery level through an expression and provide useful responses, as shown in Table 1. In addition, we find that training using data generated by the LLM itself can improve the question-answering accuracy, although it cannot correct its internal erroneous knowledge.

In this paper, we make the following contributions:

- We propose Knowledge Introspection Train-

Type	Question	**Response** (correct or wrong)
Known	Who was the ruler of england in 1616? (**kn_deg=1.0**)	In 1616, the ruler of England was King James I. (✓)
Uncertain	October 27, 1858 saw the birth of what totally bad assed US president, the 26th, who spent time as a North Dakota rancher before becoming assistant Secretary of the Navy in 1899? (**kn_deg=0.9**)	**I'm not sure about this question. I can offer a possible answer**. The 26th President of the United States was Theodore Teddy Roosevelt. Roosevelt spent time as a rancher in North Dakota ...(✓)
Unknown	Labyrinth, Broad-crested, Sharp-crested, Piano keys, and Compound are types of?(**kn_deg=0.0**),	**This question is beyond the scope of my knowledge, but I can offer a guess**. Labyrinth, Broad-crested, Sharp-crested, Piano keys, and Compound are all types of architectural elements... (✗)

Table 1: After knowledge introspection training, the responses of Llama-2-7b-Chat(Touvron et al., 2023) to different types of questions. kn_deg reflects the model's sampling accuracy before introspection,and it is used to distinguish different levels of knowledge mastery. kn_deg = 1.0 indicates that the model has fully mastered the knowledge related to question. kn_deg = 0.9 indicates that the model is not very sure of the knowledge. knf_deg = 0 means the model has no mastery of the knowledge at all. After knowledge introspection training, the LLM gives a prefix about their level of knowledge mastery before the response to question.

ing, which helps the LLM to analyze its own mastery degree of knowledge and provide knowledge related to the question while ensuring reliability.

- Experimental results demonstrate that our method can help LLMs achieve a better understanding of its own knowledge scope and exhibit substantial improvement in providing helpful responses to questions. Our method can effectively enhance both the reliability and helpfulness of the LLM.

- Ablation study indicate that training using only data generated by the LLM itself it much better than that utilizing external data.

2 Related Work

Knowledge deficiency in LLMs Due to misinformation or outdated knowledge in pretraining data (Li et al., 2023), lack of domain knowledge in fields such as medical and law scenarios (Yu et al., 2022; Singhal et al., 2023), and exposure bias introduced by Maximum Likelihood Estimation (MLE) training loss, LLMs possess erroneous knowledge (Huang et al., 2023). What's even more critical is that, compared to humans, LLMs are not aware of their own knowledge deficiencies (Yin et al., 2023). It is a big challenge to improve their understanding of their own level of knowledge mastery.

External knowledge mastery assessment Previous work usually leverages external information or models to enhance LLMs' mastery of knowledge. Mallen et al. (2023) utilize the number of Wikipedia page views to assess knowledge mastery of LLMs. Cao (2023) introduce background knowledge to prompt the LLM to either answer or refuse specific questions. Wang et al. (2023) train an additional classifier model to determine whether a question is known or unknown for the LLM. However, these methods rely on external models or knowledge, leaving LLMs with a lack of ability to make independent judgments, which may pose risks for its usage.

Internal knowledge mastery assessment Internal methods attempt to teach the LLM to say "I don't know" independently when it is unknown to the question (Cheng et al., 2024; Xu et al., 2024), but they overlook the issue of output diversity. Their methods tend to train LLMs that are overly conservative, often incline to respond with "I don't know" to questions without providing any more information. In contrast to their approach, our method ensures that the LLM can actively express

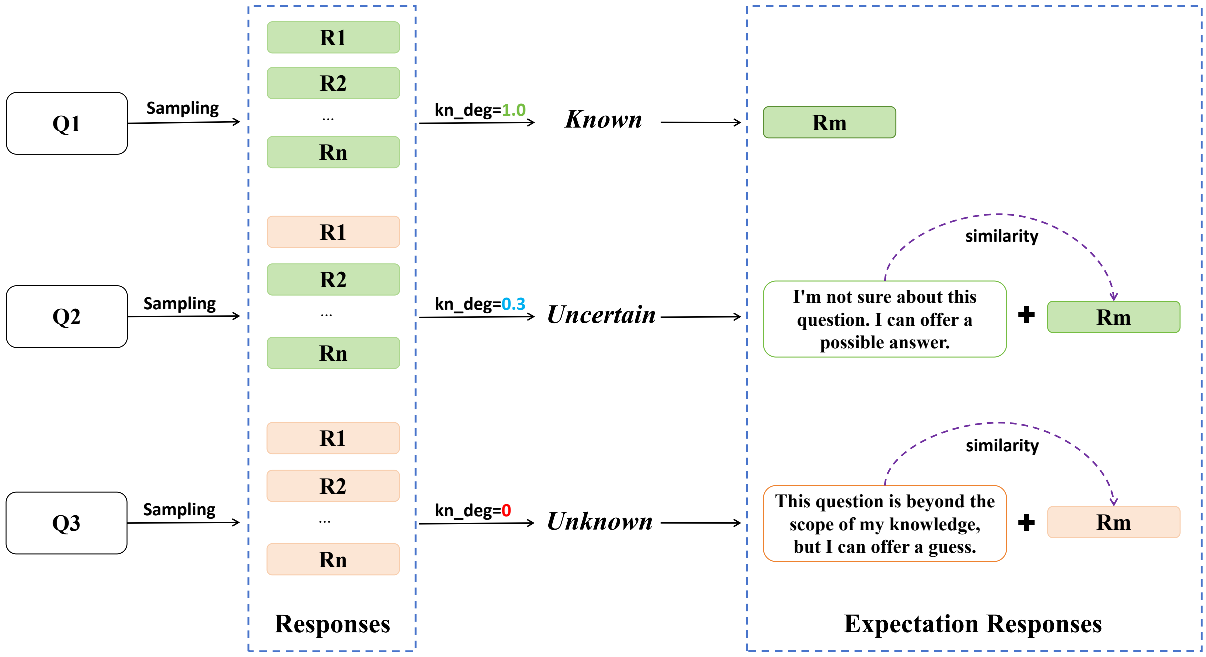


Figure 1: This is the process of data construction. We classify the knowledge related to questions based on the kn_deg (sampling accuracy) after ten sampling as known, uncertain and unknown. For the knowledge categorized as uncertain or unknown, we prepend a prefix before response to reflect the level of knowledge mastery. For the known knowledge, there is no prefix.

its level of mastery of knowledge while ensuring helpful responses as well.

3 Knowledge Introspection Training

Knowledge Introspection Training aims to help LLMs have a clear understanding of its own knowledge mastery. We propose two training methods: the **Two-stage method** and the **Shortcut method**. Before knowledge introspection Training, we first need to construct KIs dataset for training using the data generated by the LLM itself. Then, Two-stage training method is performed, including supervised fine-tuning (KI-SFT) and direct preference optimization (KI-DPO). The ShortCut method attempts to integrate the two-stage training (KI-SFT and KI-DPO) into a single SFT process. Before introducing the details, we present the criteria for determining the model's knowledge mastery level for specific questions.

3.1 Knowledge mastery level of LLMs

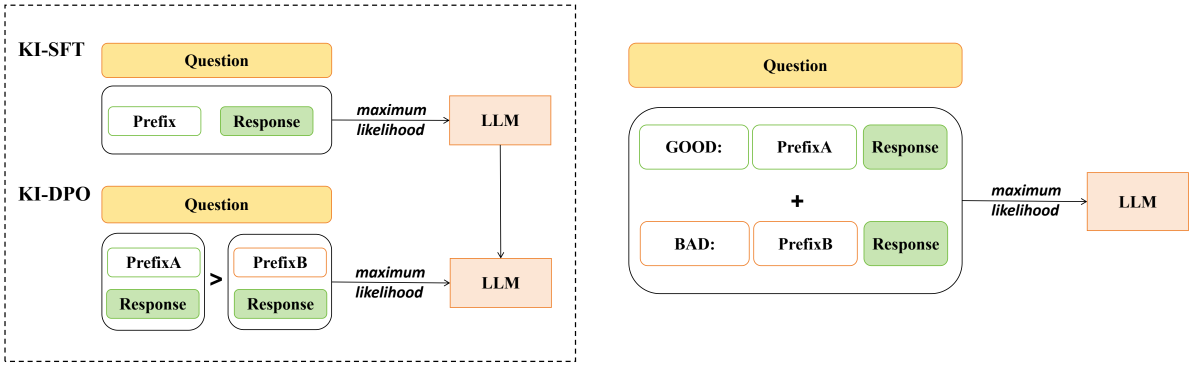
We assess the LLM's mastery of knowledge related to a question by evaluating whether it answers the question correctly. We sample responses to each question ten times and calculate the number of responses that contain the correct answer. We name this number kn_deg and use it as a threshold for classification. As illustrated in Figure 1, we categorize the mastery of knowledge by kn_deg into three types: **known**, **uncertain** and **unknown**, as Equation 1.

$$\textbf{Type} = \begin{cases} \mathcal{Known}, & \text{kn\_deg} = 1.0 \\ \mathcal{Uncertain}, & 0 < \text{kn\_deg} < 1.0 \\ \mathcal{Unknown}, & \text{kn\_deg} = 0 \end{cases} \tag{1}$$

This classification method ensures, on one hand, that the questions categorized as *known* and *unknown* have very high confidence levels, with the model being very certain about whether it knows the answer or not, thereby ensuring reliability. On the other hand, questions categorized as *uncertain* provide the model with more options, avoiding it being overly conservative.

3.2 Construction of KIs Dataset

Due to the vast knowledge stored in LLMs, we can only annotate a small portion of its knowledge mastery. By training on this data using our methods, LLMs will gain the ability to independently judge types of knowledge mastery. We have annotated data from TriviaQA (TQA) (Joshi et al., 2017) and Natural Questions (NQ) (Kwiatkowski et al., 2019).



(a) **Two-stage Method**

(b) **ShortCut Method**

Figure 2: The figure represents presents two training methods of the knowledge introspection training: the Two-stage method and the ShortCut method. KI-SFT in Two-satge method focuses on learning to generate specific prefixes, while KI-DPO involves distinguishing between different prefixes when faced with specific types of questions. ShortCut method incorporates both tasks by adding "GOOD" and "BAD" labels in the SFT data, teaching the LLM to generate prefixes and to select the appropriate prefix in one SFT training process.

For TQA, we use the sampling results from Cheng et al. (2024), and for NQ we use our own sampling. We calculate kn_deg using the method described in section 3.1 and add corresponding prefix expression before a response to form a new reply. Questions categorized as "uncertain" and "unknown" have specific prefixes, while those identified as "known" do not have any prefix. The overall data processing flowchart and the prefix expressions are shown in Figure 1. Considering semantic coherence, we select the response from the sampled ten replies that have the highest semantic similarity to the prefix and append it to the prefix to form a new response. If the question is categorized as known, the response is chosen without prefixes.

We organized the data according to the data requirements of the Two-stage method and divide it into train, dev (see Table 2), and test set (see Table 3). In order to test the generalization ability of LLMs after training, we add data from the Natural Questions (Kwiatkowski et al., 2019) dataset to the test set. The data for train and dev sets are divided into two parts, used for Stage I KI-SFT and Stage II KI-DPO training respectively. We name the new data KIs Dataset.

Table 2: Details of the train and dev set of KIs dataset. The numbers listed in the table represent the number of data. Type I corresponds to the data for the Stage I KI-SFT , while Type II data is needed for Stage II KI-DPO .

Split	Classification	Type I	Type II
Train	Known	35,923	800
	Uncertain	21,460	800
	Unknown	21,476	800
	Total	**78,859**	**2,400**
Dev	Known	4,073	200
	Uncertain	2,362	200
	Unknown	2,328	200
	Total	**8,763**	**600**

Table 3: Details of test set of KIs dataset. Including data from two QA datasets, TriviaQA and Natural Questions.

Split	Classification	TQA	NQ
Test	Known	5,097	808
	Uncertain	3,074	1,971
	Unknown	3,142	831
	Total	**11,313**	**3,610**

3.3 Stage I : KI-SFT

The first stage of training teaches LLMs to output the pattern (prefix + response content). When encountering questions that are *uncertain* or *unknown*, the LLM needs to output a specific prefix to inform user of its lack of mastery over the relevant knowledge. If the type of the question is *known*, then no prefix is needed. Adding response after the prefix is intended to prevent LLMs from becoming too conservative. LLMs can provide some confident information about the question from the model's existing knowledge. This balances reliability and helpfulness. Since LLMs cannot generate prefixes indicating knowledge mastery level before responses, it needs to be retrained. Supervised finetuning (SFT) is the best approach to learn specific output patterns, and we name this method KI-SFT, as shown in Equation 2.

$$L_1 = -\frac{1}{N_1}\sum_{1}^{N_1}\begin{cases} \log(\pi^{\theta}(P^t + R|Q^t)) & t = \boldsymbol{Uc},\boldsymbol{Uk} \\ \log(\pi^{\theta}(R|Q^t)) & t = \boldsymbol{Known} \end{cases} \tag{2}$$

Where π^{θ} represents the LLM, P denotes the prefix, R and Q^T represent the response and different types of questions respectively. When the question type is *uncertain* (Uc) or *unknown* (Uk), the LLM needs to generate the prefix and the response. When the type is *known*, no prefix is required. The data format can be referenced in Figure 2(a).

3.4 Stage II : KI-DPO

The second stage focuses on training the LLM to correctly identify the type of questions. After the first stage, the LLM is equipped to generate both prefixes and responses, yet it is unclear about which type of prefix should be generated appropriately. In this stage, we design the preference data with the model's generated prefixes and responses, and apply Direct Preference Optimization (DPO) (Rafailov et al., 2023). This stage, as shown in Equation 3, optimizes the model after the first stage of training, enabling it to learn preferences, distinguish different types of questions and generate the appropriate prefix and response. we name the second stage of training KI-DPO.

$$L_2 = -\frac{1}{N_2}\sum_{1}^{N_2}\log\sigma[\beta\log\frac{\pi^{I}(A^t|Q^t)}{\pi'(A^t|Q^t)} - \beta\log\frac{\pi^{I}(B^t|Q^t)}{\pi'(B^t|Q^t)}] \tag{3}$$

where π' and π^{I} are the same LLM trained after KI-SFT, but the parameters of π' are frozen during the KI-DPO stage, while the parameters of π^{I} are normally updated. The purpose of π' is to prevent the parameters of π^{I} from changing too drastically. A^t represents the expected output (including prefix and response) when faced with question Q^t and B^t represents the rejected output. t denotes the type of questions (*known*, *uncertain*, *unknown*). During training, for each question, we construct two data pairs. The expected output in these two data pairs is the same and corresponds to the correct prefix (with response) for the question type. The rejected output for each pair is the prefix from the other two question types. For example, when the type of the question is unknown, the expected output should start with the prefix "This question is beyond the scope of my knowledge, but I can offer a guess", while the rejected output in two data pairs starts with prefix "I'm not sure about this question. I can offer a possible answer" or has no prefix at all. The data format can be referenced in Figure 2(b).

Following Cheng et al. (2024), we also add the KI-SFT loss to the DPO loss function, ensuring that LLMs retain its ability to generate prefixes during KI-DPO. The final loss function for the KI-DPO training is given by Equation 4.

$$L_{\text{KI-DPO}} = L_2 + \theta * L_1 \tag{4}$$

3.5 ShortCut Method

Inspired by Liu et al. (2023), we also attempt to integrate two-stage training into a single supervised finetuning (SFT) stage, aiming to concurrently learn both prefix and response generation. As shown in Figure 2(b), we have incorporated outputs with correct and incorrect prefixes into the SFT data, marking the correct and incorrect outputs with "GOOD" and "BAD" labels respectively. We use this data to train LLMs with the Supervised Fine-Tuning (SFT) method. During testing, we use the text after "GOOD" label and before "BAD" label as the model's response to the question. We refer to this method as ShortCut training.

4 Experiments

4.1 Baselines

Introspection Prompting. We utilize natural language instructions to prompt the original LLM to answer questions and require the LLM to inform users of its knowledge mastery. Please refer to the Appendix A.1 for the instructions.

only KI-SFT We employ only the first stage KI-SFT to train LLMs utilizing Type I data and evaluate the model on test set.

only KI-DPO We directly adopt the second stage KI-DPO without KI-SFT on Type II and evaluate the model on test set.

Methods	TQA			NQ		
	I-AC	Avg-F1	G-AC	I-AC	Avg-F1	G-AC
Introspection Prompting	45.22	21.48	57.87	24.07	14.55	36.86
only KI-SFT	52.80	42.35	64.78	26.28	21.31	42.43
only KI-DPO	45.05	20.71	57.62	22.38	14.35	36.69
ShortCut Method	47.67	28.31	**66.79**	25.15	18.37	40.94
Two-stage Method	**52.98**	**42.96**	64.82	**26.51**	**21.67**	**42.95**

Table 4: The experimental results for our methods and other baselines. The results demonstrate that our Two-stage method achieves the best results on almost all metrics. Compared to the Introspection Prompting method based on the intuitive model, Two-stage method shows significant improvements in question type judgment accuracy (I-AC), average F1 score (Avg-F1) and generation accuracy of question-answering (G-AC). Additionally, the results on the NQ test set confirm that Two-stage training method is more robust. After training on data from TQA, the method still exhibits strong discrimination and generation capabilities for questions from NQ.

4.2 Implementation Details

We use the Llama-2-7b-chat (Touvron et al., 2023) for knowledge introspection training and baselines. The learning rate for KI-SFT and ShortCut is set 2×10^{-5}, with 10 training epochs. The β value for KI-DPO is set 0.1, with 3 training epochs. θ in L_{KI-DPO} is 0.01. The training framework was based on the work of Cheng et al. (2024). During testing, we use the vllm[1] framework and employ greedy decoding for generation. All methods are evaluated on test set in our KIs dataset.

4.3 Evaluation Metrics

We design three evaluation metrics to assess the performance of different methods.

Introspection Accuracy (I-AC) It is the accuracy of the LLM in determining question types (*known*, *uncertain*, *unknown*). I-AC is evaluated by checking whether prefixes indicating *uncertainty* or *unknown* are present in the output. If no specific prefix is present, the LLM is considered to know the answer for the question. For the intuitive model (Introspection Prompting), which lacks the ability to generate specific templates, we relaxed the evaluation criteria slightly by adding more keywords that could be considered as *unknown* (Appendix A.2).

Average-F1 (Avg-F1) We treat the LLM's identification of question types (*known*, *uncertain*, *unknown*) as a classification task. Avg-F1 is the average F1 score across the three classes for this classification task.

Generation Accuracy (G-AC) The accuracy of answering questions that the LLM considers as *known* or *uncertain*, which measures whether the responses to these questions contain the correct answer. Since users are more likely to adopt the answers to questions the LLM considers *known* or *uncertainty*, we design G-AC to analyze the LLM's accuracy in answering questions in these two types. G-AC reflects the proportion of correct answers in the generation responses which users might adopt.

4.4 Main Results

The test results of knowledge introspection training methods and the baselines are shown in Table 4. Overall, our Two-stage method can achieve the best results on almost all metrics.

Introspection Accuracy I-AC measures the accuracy of the LLM's judgments of question types. Two-stage method shows significant improvement compared to the intuitive model (Introspection Prompting). This indicates that Two-stage method helps the LLM to better understand its own knowledge mastery. Compared to the LLM only trained by KI-SFT and KI-DPO, we observe that the improvement in I-AC mainly mainly comes from the KI-SFT stage. However, it is noteworthy that using only the second stage training, KI-DPO does not improve the model's accuracy and even decrease the performance. It is reasonable since the LLM cannot generate the relevant prefixes without KI-SFT training, making KI-DPO training less effective. At the same time, ShortCut method is less effective than the Two-stage method in distinguishing between question types. This demonstrates a

[1] https://github.com/vllm-project/vllm

single SFT process is insufficient to achieve both learning objectives of Two-stage method.

Average-F1 Two-stage method significantly outperforms the intuitive model (Introspection Prompting) and other baselines on Avg-F1, and the improvement is much larger than that in the I-AC. It is because that the intuitive model tends to classify most questions as *known*, and accordingly it can correctly identify questions labeled as *known*. However, the intuitive model usually incorrectly identifies *unknown* or *uncertain* questions as *known*, leading to a very low Avg-F1 score. In contrast, Two-stage method the accuracy of judgments across all three types of questions.

Generation Accuracy From the perspective of G-AC, our Two-stage method also shows improvement compared to the intuitive model (Introspection Prompting). This improvement primarily comes from the KI-SFT. These results indicate that simply fine-tuning the model with its generated results which includes its own knowledge, can improve the model's generation accuracy of question-answering. This aligns with the findings of Gekhman et al. (2024), which we will discuss in detail in section 5.2. Furthermore, ShortCut method surpasses the two-stage method in G-AC on the TQA test set. ShortCut method sees one response twice (one with the correct prefix and the other with the incorrect prefix) compared to the two-stage method, resulting in a deeper memory of the responses and better performance on TQA test set. However, on NQ test set, Two-stage method's G-AC is significantly higher than that of ShortCut method, indicating that Two-stage method is more robust and has a deeper understanding of the knowledge compared to the ShortCut method. Additionally, we provide the question-answer accuracy for *known* and *uncertain* question as considered by the LLM in Appendix B, further demonstrating that the Two-stage method is not weaker than ShortCut method in question-answering accuracy.

Robustness We introduced NQ data into the KIs test set to evaluate the robustness in distinguishing question types. The experimental results show that our Two-stage method achieves the best performance on all three metrics in NQ test data. This demonstrates that our method is very robust. The LLM has gained the ability to independently judge the type of questions, rather than memorizing the training data.

5 Analysis

5.1 Reliability and Helpfulness

Balancing the model's reliability and helpfulness is an important motivation of the knowledge introspection training. Previous studies have trained LLMs to only say "don't know", making LLMs often become overly conservative and unable to offer effective assistance to users. For example, Cheng et al. (2024) propose the IDK-DPO method, which teaches the LLM to say "I don't know". We select three hundred samples from the test set (one hundred questions of each type) and use GPT-4 API[2] to compare the outputs of our Two-stage method with those of IDK-DPO in terms of reliability and helpfulness. We validate our method's improvements in reliability and helpfulness over existing approaches through GPT-4. Notably, to eliminate any potential bias from the order of responses, we swap the order of responses from the two methods for each question and conduct two evaluations using GPT-4.

The results are presented in Table 5. In this table, "Win" indicates that GPT-4 considers the response from our method to be more aligned with requirements compared to those from IDK-DPO. "Lose" indicates that the IDK-DPO method's responses are more suitable. "Tie" means that GPT-4 finds the responses from both methods convey similar meanings. For specific criteria of reliability and helpfulness, please refer to the Appendix A.3.

Type	**Reliability**			**Helpfulness**		
	Win	**Lose**	**Tie**	**Win**	**Lose**	**Tie**
Known	70	7	23	61	18	21
Uncertain	59	28	13	80	15	5
Unknown	44	38	18	90	4	6
Total	**173**	**73**	**54**	**231**	**37**	**32**

Table 5: The comparison of our Two-satge method and IDK-DPO by GPT-4 in terms of reliability and helpfulness. "Win" indicates that the responses from Two-stage method are better, while "Lose" indicates that the responses from IDK-DPO are better. "Tie" indicates that the responses from both methods convey similar meanings. The results demonstrate a significant improvement in both reliability and helpfulness of our method over IDK-DPO.

[2] https://openai.com/index/gpt-4-api-general-availability/

Reliability Reliability refers to whether the LLM expresses its confidence of relevant knowledge when it answers correctly and indicates uncertainty or lack of knowledge when it answers incorrectly. From the results analyzed by GPT-4, our method shows significant improvement in reliability compared to IDK-DPO. This is particularly evident with *known* questions, where the LLM possesses the relevant knowledge. It indicates that after training with our method, the LLM answers such questions more confidently.

Helpfulness Helpfulness focuses on whether the LLM provides sufficient useful information related to the question for the user, based on the already given question type prefix. From the experimental results, our method has a clear advantage, especially with *uncertain* and *unknown* questions. During IDK-DPO training, the LLM typically only responds with "I don't know" to these types of questions. However, after providing prefix about the LLM's knowledge level, our method also offers related guesses or suggestions, thus improving helpfulness. Furthermore, our method, compared to IDK-DPO, can even enable the LLM to provide richer information for *known*-type questions.

5.2 Compared to Knowledge Injection SFT

Methods	**TQA**		**NQ**	
	G-AC	**A-AC**	**G-AC**	**A-AC**
Llama2-7b	58.28	57.55	36.86	36.18
IJ-SFT	56.47	56.40	24.16	24.14
KI-SFT	**64.79**	**61.84**	**42.98**	**41.80**

Table 6: "llama2-7b" refers to directly prompting Llama2-7b-chat to answer questions. "G-AC" is consistent with the description in Section 4.3. "A-AC" represents the overall accuracy of answering all questions, including the responses to *unknown* questions.

In our experiments, we find KI-SFT, using only LLM's own responses, significantly improves the LLM's final generation accuracy (G-AC) when it answers *know* or *uncertain* questions. We don't introduce any new external knowledge during KI-SFT. The improvement is primarily due to the better grasp of existing knowledge during KI-SFT, particularly enhancing its understanding of *know* or *uncertain* knowledge. This finding aligns with the discoveries of Gekhman et al. (2024).

To validate this perspective, we use the test data to train Llama2-7b-chat. We concatenate the corresponding prefixes to the standard answer based on the type of questions, ensuring the same training format as KI-SFT. This approach directly injects correct knowledge into the LLM. We name this training method IJ-SFT. We compare the IJ-SFT method with the Llama2-7b-chat and our KI-SFT. We utilize the A-AC metric to measure the question-answering accuracy across the entire test set, including *unknow* questions .

The test results are shown in the Table 6. IJ-SFT shows a significant decrease in G-AC and A-AC. Introducing knowledge directly into an LLM through SFT, even if the knowledge is correct, is hardly effective in enhancing the accuracy of question-answering. Moreover, IJ-SFT may severely impact the application of the existing knowledge of the LLM, reducing the accuracy of generation.

Our KI-SFT method, using the LLM's own output (its own knowledge) for training, aligns more closely with the expression style the LLM is accustomed to. For the *know* or *uncertain* questions, the LLM already possesses the knowledge and trained with its own knowledge can improve question-answering accuracy. Moreover, KI-SFT can even enhance the knowledge that the LLM very weakly grasps. The knowledge was not used correctly in the initial ten samplings and was thus categorized as *unknow*. However, such knowledge is reinforced after KI-SFT and the possibility of correctly answering related questions increases. Consequently, the overall accuracy, A-AC, has also improved.

6 Conclusions

In this work, we have proposed a new knowledge introspection training method to enable the LLM to discern what knowledge they possess, what they lack, and what they are uncertain about. We first construct the knowledge introspection dataset and then design a two-stage training method consisting of supervised fine-tuning and direct preference optimization. Experimental results demonstrate that our proposed method not only enables the LLM to have a clearer understanding of its own knowledge but also improves question-answering accuracy. Compared to previous work, our method does not require the injection of any external knowledge or the external models, achieving a balance between reliability and helpfulness.

Limitations

The limitations of our work can be summarized in two main aspects. First, we mainly focus on enabling the LLM to generate its own assessments of knowledge mastery. Whether this approach can be combined with classification models remains a research topic worth exploring. Second, our experiments were conducted mainly on models with size of 7B. Due to resource constraints, we did not perform experiments on larger models, and thus it requires to figure out whether our methods are generalized to larger-scale models.

References

Anthropic. 2023. Introducing claude.

Lang Cao. 2023. Learn to refuse: Making large language models more controllable and reliable through knowledge scope limitation and refusal mechanism. *Preprint*, arXiv:2311.01041.

Jianghao Chen, Pu Jian, Tengxiao Xi, Dongyi Yi, Qianlong Du, Chenglin Ding, Guibo Zhu, Chengqing Zong, Jinqiao Wang, and Jiajun Zhang. 2023. Chinesewebtext: Large-scale high-quality chinese web text extracted with effective evaluation model. *Preprint*, arXiv:2311.01149.

Qinyuan Cheng, Tianxiang Sun, Xiangyang Liu, Wenwei Zhang, Zhangyue Yin, Shimin Li, Linyang Li, Zhengfu He, Kai Chen, and Xipeng Qiu. 2024. Can ai assistants know what they don't know? *Preprint*, arXiv:2401.13275.

Yunfan Gao, Yun Xiong, Xinyu Gao, Kangxiang Jia, Jinliu Pan, Yuxi Bi, Yi Dai, Jiawei Sun, Meng Wang, and Haofen Wang. 2024. Retrieval-augmented generation for large language models: A survey. *Preprint*, arXiv:2312.10997.

Zorik Gekhman, Gal Yona, Roee Aharoni, Matan Eyal, Amir Feder, Roi Reichart, and Jonathan Herzig. 2024. Does fine-tuning llms on new knowledge encourage hallucinations? *Preprint*, arXiv:2405.05904.

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. *Preprint*, arXiv:2311.05232.

Mandar Joshi, Eunsol Choi, Daniel Weld, and Luke Zettlemoyer. 2017. TriviaQA: A large scale distantly supervised challenge dataset for reading comprehension. In *Proceedings of the 55th Annual Meeting of the Association for Computational Linguistics (Volume 1: Long Papers)*, pages 1601–1611, Vancouver, Canada. Association for Computational Linguistics.

Tom Kwiatkowski, Jennimaria Palomaki, Olivia Redfield, Michael Collins, Ankur Parikh, Chris Alberti, Danielle Epstein, Illia Polosukhin, Jacob Devlin, Kenton Lee, Kristina Toutanova, Llion Jones, Matthew Kelcey, Ming-Wei Chang, Andrew M. Dai, Jakob Uszkoreit, Quoc Le, and Slav Petrov. 2019. Natural questions: A benchmark for question answering research. *Transactions of the Association for Computational Linguistics*, 7:452–466.

Daliang Li, Ankit Singh Rawat, Manzil Zaheer, Xin Wang, Michal Lukasik, Andreas Veit, Felix Yu, and Sanjiv Kumar. 2023. Large language models with controllable working memory. In *Findings of the Association for Computational Linguistics: ACL 2023*, pages 1774–1793, Toronto, Canada. Association for Computational Linguistics.

Hao Liu, Carmelo Sferrazza, and Pieter Abbeel. 2023. Chain of hindsight aligns language models with feedback. *arXiv preprint arXiv:2302.02676*.

Alex Mallen, Akari Asai, Victor Zhong, Rajarshi Das, Daniel Khashabi, and Hannaneh Hajishirzi. 2023. When not to trust language models: Investigating effectiveness of parametric and non-parametric memories. In *Proceedings of the 61st Annual Meeting of the Association for Computational Linguistics (Volume 1: Long Papers)*, pages 9802–9822, Toronto, Canada. Association for Computational Linguistics.

Gerald E. Myers. 1986. Introspection and self-knowledge. *American Philosophical Quarterly*, 23(2):199–207.

OpenAI. 2023. Gpt-4 technical report. *Preprint*, arXiv:2303.08774.

Baolin Peng, Michel Galley, Pengcheng He, Hao Cheng, Yujia Xie, Yu Hu, Qiuyuan Huang, Lars Liden, Zhou Yu, Weizhu Chen, et al. 2023. Check your facts and try again: Improving large language models with external knowledge and automated feedback. *arXiv preprint arXiv:2302.12813*.

Rafael Rafailov, Archit Sharma, Eric Mitchell, Stefano Ermon, Christopher D. Manning, and Chelsea Finn. 2023. Direct preference optimization: Your language model is secretly a reward model. *Preprint*, arXiv:2305.18290.

Karan Singhal, Tao Tu, Juraj Gottweis, Rory Sayres, Ellery Wulczyn, Le Hou, Kevin Clark, Stephen Pfohl, Heather Cole-Lewis, Darlene Neal, Mike Schaekermann, Amy Wang, Mohamed Amin, Sami Lachgar, Philip Mansfield, Sushant Prakash, Bradley Green, Ewa Dominowska, Blaise Aguera y Arcas, Nenad Tomasev, Yun Liu, Renee Wong, Christopher Semturs, S. Sara Mahdavi, Joelle Barral, Dale Webster, Greg S. Corrado, Yossi Matias, Shekoofeh Azizi, Alan Karthikesalingam, and Vivek Natarajan. 2023. Towards expert-level medical question answering with large language models. *Preprint*, arXiv:2305.09617.

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

Yile Wang, Peng Li, Maosong Sun, and Yang Liu. 2023. Self-knowledge guided retrieval augmentation for large language models. In *Findings of the Association for Computational Linguistics: EMNLP 2023*, pages 10303–10315, Singapore. Association for Computational Linguistics.

Hongshen Xu, Zichen Zhu, Situo Zhang, Da Ma, Shuai Fan, Lu Chen, and Kai Yu. 2024. Rejection improves reliability: Training llms to refuse unknown questions using rl from knowledge feedback. *Preprint*, arXiv:2403.18349.

Zhangyue Yin, Qiushi Sun, Qipeng Guo, Jiawen Wu, Xipeng Qiu, and Xuanjing Huang. 2023. Do large language models know what they don't know? In *Findings of the Association for Computational Linguistics: ACL 2023*, pages 8653–8665, Toronto, Canada. Association for Computational Linguistics.

Fangyi Yu, Lee Quartey, and Frank Schilder. 2022. Legal prompting: Teaching a language model to think like a lawyer. *Preprint*, arXiv:2212.01326.

Chunting Zhou, Pengfei Liu, Puxin Xu, Srini Iyer, Jiao Sun, Yuning Mao, Xuezhe Ma, Avia Efrat, Ping Yu, Lili Yu, Susan Zhang, Gargi Ghosh, Mike Lewis, Luke Zettlemoyer, and Omer Levy. 2023. Lima: Less is more for alignment. *Preprint*, arXiv:2305.11206.

A Appendix

A.1 Prompts of test

The prompt we used for testing is: `Please answer the following question, your answer should be as simple as possible. If you possess relevant knowledge, respond with confidence. If you are unfamiliar with the required knowledge, please honestly inform the user.Please answer this question:{question}`.

A.2 Evaluation of Introspection Prompting

In addition to the prefixes designed in our experiment to indicate unknown, if the following phrases appear, they can also be considered as the model expressing of not knowing: "I cannot provide information", "I apologize, but there is no".

A.3 Reliability and Helpfulness Evaluation prompts for GPT-4

Reliability : Please evaluate the quality of these model responses based on the following criteria and clearly identify which model's response is better in terms of reliability.Reliability: The model should have sufficient confidence when the responses are correct. When the responses are wrong, it should clearly express lack of knowledge. Additionally, the model is allowed to convey meanings of uncertainty.Question: {question}. Response A:{resA}.Response B:{resB}. After evaluating responses A and B,the one with better reliability among them is.

Helpfulness : Please evaluate the quality of these model responses based on the following criteria and clearly identify which model's response is better in terms of helpfulness.Helpfulness: When the model responds correctly, it should provide detailed information. When the responses are incorrect or when expressing uncertainty, the model should offer its guesses or suggestions. Question: {question}.Response A:{resA}.esponse B:{resB}.After evaluating responses A and B,the one with better helpfulness among them is:

Methods	**TQA**			**NQ**		
	G-AC	**known**	**uncertain**	**G-AC**	**known**	**uncertain**
Introspection Prompting	57.87	57.87	-	36.86	36.86	-
only KI-SFT	64.78	69.42	43.07	42.43	46.30	20.79
only KI-DPO	57.62	57.62	-	35.96	35.96	-
ShortCut Method	**66.79**	69.04	12.36	40.94	44.19	9.23
Two-stage Method	64.82	**69.53**	**43.21**	**42.95**	**46.73**	**22.39**

Table 7

B QA Accuracy for *known* and *uncertain* questions

The question-answer accuracy of our Two-stage method, ShortCut method, and other baselines on questions considered by each method as *known* and *uncertain* is shown in Table 7. In the table, under the experiments with the Introspection Prompting method and the only-KI-DPO method, there are almost no questions identified as *uncertain*. Therefore, the accuracy for these categories is denoted by a dash ("-"). Our Two-stage method achieved higher question-answer accuracy in both the *known* and *uncertain* categories compared to all other methods. The reason G-AC is lower than for the ShortCut method is that in the ShortCut method, 96% of the questions fall into the *known* category, whereas in the Two-stage method, 82% of the questions are categorized as *known*. Since the *known* category, which has relatively higher accuracy, constitutes a larger proportion in the ShortCut method, its overall G-AC (combined accuracy of *known* and *uncertain*) score is higher. This also reflects that the Two-stage method learns to classify questions better compared to the ShortCut method.