Learning to Trust Your Feelings: Leveraging Self-awareness in LLMs for Hallucination Mitigation

Anonymous ACL submission

Abstract

We evaluate the ability of Large Language Models (LLMs) to discern and express their internal knowledge state, a key factor in countering factual hallucination and ensuring reliable application of LLMs. We observe a robust self-awareness of internal knowledge state in LLMs, evidenced by over 85% accuracy in knowledge state probing. However, LLMs often fail to faithfully express their internal knowledge during generation, leading to factual hallucinations. We develop an automated hallucination annotation tool, DreamCatcher, which merges knowledge probing and consistency checking methods to rank factual preference data. Using knowledge preference as reward, We propose a Reinforcement Learning from Knowledge Feedback (RLKF) training framework, leveraging reinforcement learning to enhance the factuality and honesty of LLMs. Our experiments across multiple models show that RLKF training effectively enhances the ability of models to utilize their internal knowledge state, boosting performance in a variety of knowledge-based and honesty-related tasks.

1 Introduction

Large Language Models (LLMs), including notable examples such as GPT-3 (Brown et al., 2020), LLaMA (Touvron et al. (2023a), Touvron et al. (2023b)), and PaLM (Chowdhery et al., 2023), have emerged as a transformative tool in diverse fields due to their robust capabilities in various tasks. However, despite this significant progress and success, an inherent challenge continues to persist: their tendency to "hallucinate", i.e., generate content misaligned with actual facts. This issue is particularly problematic in critical applications, such as clinical or legal scenarios, where the reliability and accuracy of generated content is vital. Therefore, mitigating hallucinations in LLMs is a crucial step toward enhancing their practical application scope and improving the overall trust in these emerging technologies.

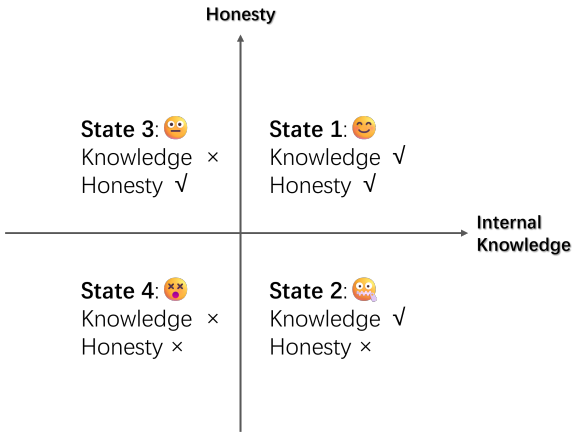


Figure 1: Internal knowledge state categorization of LLMs, based on the possession of corresponding internal knowledge and the capacity to express it honestly.

Hallucinations of LLMs can be categorized into three types (Zhang et al., 2023b): input conflict, context conflict, and factual conflict. This paper focus on the issue of fact-conflicting hallucination, where LLM produces fluent and seemingly plausible content, but conflicts with real-world facts, pose risks of misleading users and compromise the models' fact-based reasoning.

Commonly used hallucination mitigation methods, such as retrieval augmentation generation (RAG), address fact-conflict hallucination of LLM by bringing in external knowledge, but at the cost of introducing a costly and complex retrieval system. In this paper, we propose to mitigate the factual hallucination problem from the perspective of enhancing the model's utilization of internal knowledge.

Previous works (Azaria and Mitchell (2023), Agrawal et al. (2023)) have shown that LLMs have the capability to discern the validity of factual statements, supported further by Kadavath et al. (2022) suggesting these models' capacity to assess their ability in responding to specific questions. Nev-

ertheless, the universality and extent of models' self-awareness of their internal knowledge remains an open question. In light of this, we conducted exploratory experiments to probe the knowledge state of various models across different scales, employing linear probes to ascertain the accuracy of models' self-awareness regarding their internal knowledge states. The results revealed that all models under analysis demonstrated proficient accuracy in recognizing whether they have the internal knowledge about certain facts.

However, during generation, such accurate judgments do not translate into honest output; instead, in the absence of specific internal knowledge, models often manifest a tendency towards hallucinations. Therefore, to mitigate factual hallucinations, it is crucial that models leverage their self-awareness of internal knowledge states.

Leveraging this self-awareness of LLMs, we propose a training framework named reinforcement learning from knowledge feedback (RLKF) to improve the factuality and honesty of LLM with reinforcement learning using factual preferences as reward. Through the hallucination annotation method DreamCatcher – a blend of knowledge probing and consistency-based judgments – we rank the knowledge-based Question-Answering (QA) data adhering to a preference hierarchy delineated as: $factuality > uncertainty > hallucination$. This factual preference data is then utilised to train the reward model which is deployed to optimize the Large Language Model via Proximal Policy Optimisation (PPO) algorithm.

The primary contributions of this paper are articulated as follows:

1. We carry out extensive experiments on different models' capacity to discern their own internal knowledge. The results indicate that LLMs are highly adept at discerning their internal knowledge, with an impressive accuracy over **85%** in most settings with a limited amount of data.

2. We develop and open source **DreamCatcher**[1], an automatic hallucination detection tool for scoring the degree of hallucination in LLM generations. DreamCatcher integrates knowledge probing methods and consistency judgments, achieving 81% agreement with human annotator.

3. We introduce the Reinforcement Learning from Knowledge Feedback (RLKF) training framework to optimize LLM against the factual preference. The experiment results on multiple knowledge and reasoning tasks indicate that RLKF not only enhances the honesty and factuality of LLMs but also improves their general capabilities.

2 Problem Setup

Hallucination, in the context of Large Language Models, refers to a set of inconsistencies in model generation. The central focus of this paper is exploring the fact-conflict hallucination which is defined as the inconsistency between the generated content of the model and the established facts. Although the definition provides a description of the generation results, the causes underlying this phenomenon are multifaceted.

In general, LLMs encode factual knowledge into parameters during training and utilize this internal knowledge for generation during inference. However, LLMs do not always honestly express the knowledge in its parameters, which is one of the major causes of fact-conflict hallucination.

For a given question that requires factual knowledge, the model output can be classified into one of four states, depending on the model's internal knowledge and its honesty. These states are illustrated in Figure 1:

State 1: The model has relevant internal knowledge and expresses it faithfully.

State 2: Despite having the relevant internal knowledge, the model fails to express it honestly. This discrepancy could be due to various factors such as the decoding strategy (Lee et al., 2022; Chuang et al., 2023), hallucination snowballing (Zhang et al., 2023a), or misalignment issues (Schulman, 2023).

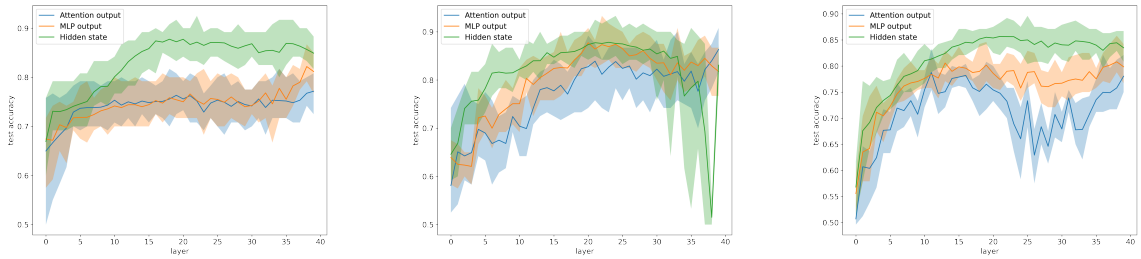
State 3: The model lacks the necessary internal knowledge but honestly indicates unawareness.

State 4: The model lacks the necessary internal knowledge and produces a hallucinated response.

Outputs in **State 2** and **State 4** are both considered forms of hallucination, despite the differing conditions of internal knowledge.

In the upper section of Figure 1, the model's outputs are devoid of hallucinations, honestly mirroring its internal knowledge. Here, **State 1** stands out as the most desirable state, where the model both possesses and faithfully outputs the relevant

[1] https://anonymous.4open.science/r/dreamcatcher-8503



(a) Llama2-chat-13B (b) Qwen-chat-14B (c) Ziya-reader-13B

Figure 2: Accuracy of knowledge state probing across different models with different internal representations. The light-colored area in the figure shows the range of accuracy for ten repetitions of the experiment, and the solid line shows the mean accuracy. More results shown in A.2

knowledge.

Many efforts have been deployed to transition model toward state 1.

Retrieval-augmented generation (RAG) attempts to bypass the lack of internal knowledge by providing knowledge via context, thereby enabling the model to transition from **State 3/4** to **State 1**. On another front, certain strategies, like those of Li et al. (2023b) and Chuang et al. (2023), seek to move the model from **State 2** to **State 1** by intervening the model's internal representation or the decoding process during inference. While these methods improve the model's capacity to express existing internal knowledge, they disregard scenarios where the model lacks relevant internal knowledge. Also, interference at inference time can potentially lead to unpredictable effects on other types of tasks.

Without the introduction of external knowledge, the mitigation of the model's fact-conflict hallucination correspond to an upward movement of the state in Figure 1. In essence, this symbolizes the enhancement of the model's capacity to accurately express its internal knowledge state. A critical question, then, is how to discern the internal knowledge state of LLMs?

3 Knowledge State Probing

This section delves into the complexities of discerning a model's internal knowledge state. It comprises two perspectives. The first, an external perspective, discuss how to determine if a model possesses specific knowledge based on the model generations; The second perspective, an internal view, questions if it is possible to determine whether a model possesses specific knowledge by its internal activation.

For the following pilot experiments, we select three families of models with different sizes: Llama2-chat(Touvron et al., 2023b) (13B and 7B); Qwen-chat(Bai et al., 2023) (14B and 7B); Ziya-reader(Junqing et al., 2023) (13B).

As for data, We randomly select passages from Chinese and English Wikipedia and instruct GPT3.5 to generate a knowledge-related question-answer pair. The answer generated by GPT3.5 based on the original Wikipedia is considered as the correct answer. We refer to the QA pairs obtained by this method as **wiki-QA** in this paper. Examples of instructions and corresponding output are shown in Appendix A.1.

3.1 External perspective

Determining whether a model has specific knowledge through its generation is a straightforward way. But it is challenging to accurately assess the model's knowledge state through a singular generation result, due to the uncertainty of generation caused by sampling (Lee et al., 2022) and different generation tendencies (Chuang et al., 2023). Multiple generation results can more faithfully reflect the knowledge state of the model.

In the presence of a correct answer, the consistency of the model's multiple generation with the correct answer is a reliable method for assessing knowledge state. The consistency of model generation with the correct answer can be computed using methods such as unigram overlap and cosine similarity of text representation.

However, the correct answer is hard to obtain in many scenarios, in which case self-consistency becomes a critical tool for assessing the validity of the generation. As evidenced by multiple research (Manakul et al. (2023), Agrawal et al. (2023), Hase et al. (2023), Elaraby et al. (2023)), there is a gen-

eral conclusion that higher consistency across multiple generations is often indicative of validity of the generation. Intuitively, if the model has the corresponding knowledge, multiple generation are likely to contain consistent facts, resulting in higher consistency. Whereas, the contents of the hallucinations often varies, leading to lower self-consistency. We evaluate the self-consistency of a certain generation by the average of the cosine similarity representations among other generated answers.

3.2 Internal perspective

Previous work (Azaria and Mitchell (2023), Kadavath et al. (2022), Li et al. (2023b)) prove that LLMs can discern the factual accuracy of certain statements, even when the false statements are self-generated. This supports the existence of state 2 in Figure 1 where the model has the corresponding knowledge but generates incorrect outputs. But are LLMs capable of discerning its own state of knowledge? The question can be rephrased as follows: for a given knowledge-related question, can a model discern its capability to output the correct answer before the actual generation of an answer? The following linear probing experiments on multiple models implies that the answer is yes.

We sample questions from the wiki-QA data, and LLM generates $k = 5$ answers for each question separately. We use the consistency method described earlier to pre-label the questions. The sum of these normalized consistency scores computed to derive the final score.

To categorize the questions, straightforward thresholds are utilized. The upper threshold is set at the 65th percentile score, and the lower at the 35th percentile score. Under this setup, responses with scores exceeding the upper threshold are labeled as correct, while those falling below the lower threshold are labeled as incorrect. If all of the k generated responses related to a specific question are deemed correct, the model is presumed to possess the relevant internal knowledge, and thus the question is labeled as 'Known'. Conversely, if all k responses are incorrect, the model is considered to lack the necessary internal knowledge, and hence the question is labeled as 'Unknown'.

A single linear layer classifier (probe) is trained on the internal representation corresponding to the last token of each question. Its task is to predict the corresponding Known/Unknown label.

For our experiments, we select three types of internal representations:

The attention output, which refers to the output of the dot product attention and before the attention linear layer in the decoder layer. This setup aligns with the probe's positioning within Li et al. (2023b); **The MLP output**, i.e., the feed-forward layer's output within the decoder layer, occurring prior to residual linkage; **The hidden states**, defined as each decoder layer's output.

The results of the internal knowledge probe experiment are shown in Figure 2, which presents the accuracy of the trained probes across different models with different internal representation and at different layers.

Comparative analysis of the experimental results across models of varying sizes yields consistent observations:

1. The linear probes of the internal state accurately predict the knowledge representation of the model. The probes' maximum accuracy surpasses 85% in most setups. This suggests that information about whether the model has the corresponding knowledge is linearly encoded in the internal representation of the model with high accuracy.

2. The accuracy of the probes increases rapidly in the early to middle layer, indicating that the model needs some layers of computation before it can determine its own knowledge states.

3. Hidden state probes exhibit the highest accuracy in discerning the knowledge state of the model, sustaining high accuracy from the middle layer to the output layer, which opens up the possibility of utilizing internal knowledge state when generating responses.

3.3 DreamCatcher

We integrated the above methods of knowledge state probing and consistency judgments to develop an automated hallucination annotation tool, DreamCatcher.

We start by collect the LLMs' generation for each question in the question set, in our case, the wiki-QA dataset. This process features two modes: normal generation and uncertainty generation. Normal generation is when the prompt contains only the question and model generates k responses, while uncertainty generation refers to where the prompt contains a request for the model to output answers that show uncertainty or lack of knowledge.

Subsequently, we assess the degree of hallucination of the generated responses using multiple

scorers using the methods described above. Concretely, we compute the following scores:

$$
\begin{aligned}
s_{s2g} &= \text{avg}_{ij}(\cos(\mathbf{r}_{G_i}, \mathbf{r}_{G_j})) \\
s_p &= \text{probe}(\mathbf{r}_Q) \\
s_{o2a} &= \text{count}(token_{overlap})/\text{count}(token_A) \\
s_{s2a} &= \cos(\mathbf{r}_G, \mathbf{r}_A)
\end{aligned}
$$

where Q denotes the question, A the correct answer, G the generation and $\mathbf{r}$ the embedding representation of text.

s_{s2g} **(Similarity to Generation Score)**: computes the cosine similarity between the embedding of certain generation (G_i) and other generated responses (G_j), using the bge-large model (Xiao et al., 2023) for text embedding.

s_p **(Probe Score)**: rates the questions by utilizing the probes trained in Section 3.2, which are intended to discern the model's knowledge state for the corresponding questions.

s_{o2a} **(Overlap with Answer Score)**: calculates the ratio of token overlap between the generated output and the correct answer (A).

s_{s2a} **(Similarity to Answer Score)**: computes the cosine similarity between the embedding of the generated response (G) and the correct answer (A), using the bge-large model for text embedding.

The scores are normalized and summed to provide an overall factuality score for each generation. The generations are then classified as "correct" or "incorrect" based on whether their total score is above or below the median score, respectively. Questions are categorized as "Known", "Unknown", or "Mixed" based on whether the responses are consistently correct, incorrect, or a combination of correct and incorrect across multiple generations, with "Mixed" being a less frequent occurrence.

The categories correspond to three ranking hierarchies as shown in Figure 3: Known (corresponding to state 1 in Fig.1): factual > uncertainty; Mixed (state 2): factual > uncertainty > hallucination; Unknown (state 4): uncertainty > hallucination. Here, "factual" refers to the generation with the highest factuality score, while "hallucination" denotes the generation with the lowest score.

We randomly sampled 200 entries, half Chinese and half English, from the DreamCatcher labeled data. Then the human annotator annotate the same data, without access to the labels of DreamCatcher. The consistency between DreamCatcher and human annotator is shown in Table 1, with an overall accuracy of 81%.

Language	Accuracy	Precision	Recall
All	81%	77%	86%
Chinese	77%	79%	76%
English	86%	76%	98%

Table 1: The consistency between DreamCatcher and human annotator. For precision and recall, we treat "correct" as a positive label and "incorrect" as negative.

4 Method

From the above knowledge-probing experiments, we discover that LLMs are capable of evaluating their own knowledge states in response to specific knowledge-based questions. This implies that LLMs demonstrate a self-awareness of their knowledge state, which does not consistently translate into their generation.

Frequently, when faced with questions outside of internal knowledge, LLMs tends to generate hallucinations. Additionally, even with questions within internal knowledge, LLMs may potentially generate incorrect responses due to other influences. One possible explanation could be that LLMs did not learn to generate with respect to the internal knowledge state during model training. Instead, the fine-tuning process often requires the model to generate seemingly reasonable answers to all factual questions.

We therefore emphasize on enhancing the model's utilization of internal knowledge state so that the model can choose to rely on internal knowledge to answer or honestly express its lack of relevant knowledge.[2]

Consequently, we propose the RLKF (Reinforce Learning from Knowledge Feedback) training framework. This introduces model knowledge state assessments into the reinforcement learning feedback mechanism, enhancing model honesty and factuality. The RLKF training process shares similarities with the standard RLHF (Reinforce Learning from Human Feedback), and can integrate smoothly with the existing RLHF framework, but reduces data collection costs by substituting

[2]This intuition could also be used for efficient RAG, enabling direct responses when the LLM possesses relevant internal knowledge, while relying on the retrieval tool in case of a knowledge gap.

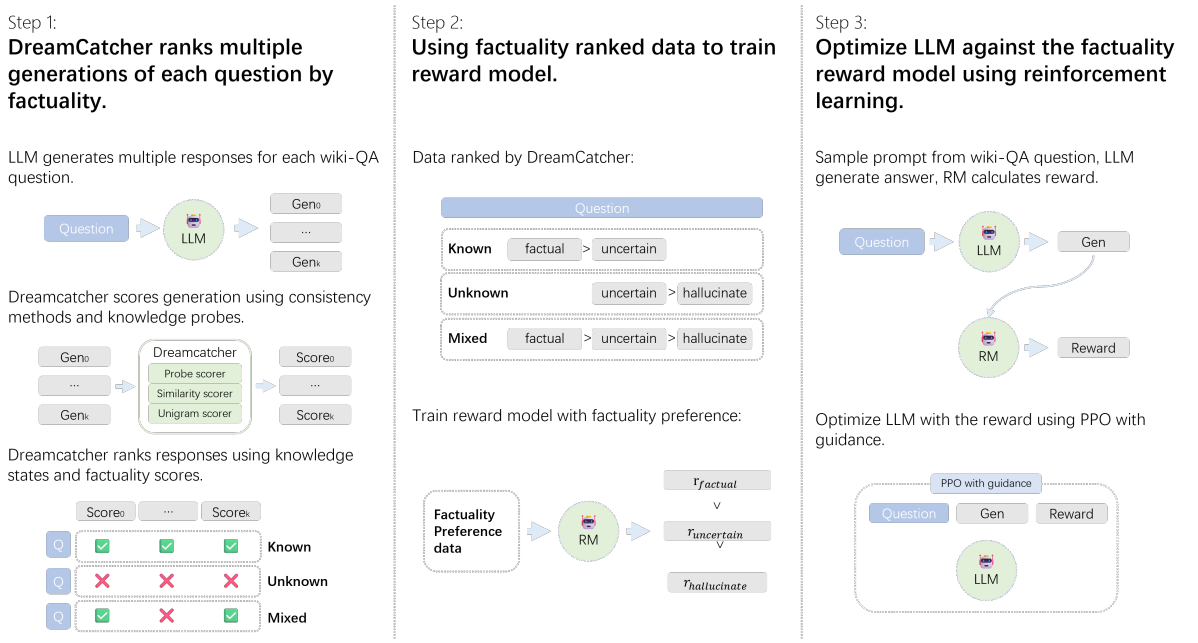


Figure 3: RLKF training framework

human labeling with automatic knowledge annotation.

The RLKF training framework consists of the following components, as shown in Figure 3.

Knowledge state annotation: We annotate factual preference data using the DreamCatcher tool.

Knowledge Feedback Modeling: Having obtained the factual preference data, we train the reward model following (Ouyang et al., 2022). The language modelling head in reward model is replaced with a linear layer to produce a scalar output, corresponding to the reward of the generated response. In line with (Köpf et al., 2023), an additional regularization parameter is introduced to prevents the predicted values from diverging too much.

By initiating the PPO Policy training and the reward model training from the same model, we can ensure that the reward model can leverage the same internal knowledge.

PPO Optimizing: Based on our factual reward model, we optimize the policy, i.e., the initial generative model, using the PPO algorithm once again following Ouyang et al., 2022. To improve the efficiency of model exploration towards honesty, we use guidance technique in reinforcement learning. Concretely, we concatenate the first few tokens of the preferred responses to the input prompts in a portion of the training data. The added tokens do not participate in the loss calculation, but can guide the model to generate desired responses, thus improving learning efficiency.

The core of the training framework is to establish the factual preference reward mechanism. The reinforcement learning algorithms in the RLKF framework can also be replaced by other optimization algorithms such as DPO (Rafailov et al., 2023), reject sampling, etc. We choose PPO to be consistent with the common practice in RLHF training.

5 Experiments

In the following experiments, We chose three different models of varying sizes: llama2-chat (13B and 7B); Qwen-chat (14B and 7B); and Ziya-reader (13B), which is consistent with the choice of models for the knowledge-probing experiments detailed in Section 3.

Model	Known	Unknown	Mixed
Qwen-chat-14B	82.7%	87.1%	77.8%
Qwen-chat-7B	65.7%	81.6%	61.1%
Llama2-chat-13B	85.4%	85.4%	60.0%
Llama2-chat-7B	78.9%	89.2%	57.6%
Ziya-reader-13B	93.5%	82.4%	64.5%

Table 2: Accuracy of trained reward model for each knowledge state category.

5.1 Data collection

We used the wiki-QA data collection method same as in Section 3, obtaining about 7,000 QA pairs

Models		MMLU	WinoGrande	ARC	BBH	GSM8K	MATH	C-Eval	CMMLU	Avg
Qwen-chat-14B	before	64.2%	53.8%	76.5%	34.5%	47.3%	18.9%	65.0%	64.1%	53.0%
	after	**64.5%**	**59.1%**	**87.2%**	**37.3%**	**49.9%**	**20.3%**	64.6%	**66.4%**	**56.2%**
Qwen-chat-7B	before	54.2%	49.6%	63.1%	28.8%	50.0%	12.6%	57.8%	58.1%	46.8%
	after	**55.3%**	**52.2%**	**75.4%**	28.1%	**50.9%**	12.5%	57.5%	56.0%	**48.5%**
Llama2-chat-13B	before	52.3%	51.9%	72.4%	21.7%	35.2%	3.2%	34.6%	34.5%	38.2%
	after	**52.8%**	**54.3%**	72.1%	**23.4%**	**35.6%**	3.1%	34.3%	**34.6%**	**38.8%**
Llama2-chat-7B	before	45.9%	51.5%	59.2%	23.3%	25.9%	1.6%	32.1%	31.6%	33.9%
	after	**46.2%**	**52.4%**	**61.1%**	**24.4%**	23.7%	**2.0%**	**34.0%**	**32.1%**	**34.5%**
Ziya-reader-13B	before	49.5%	50.8%	64.7%	44.7%	29.3%	4.3%	44.7%	46.1%	41.7%
	after	**50.3%**	**51.9%**	**67.9%**	42.6%	**33.2%**	3.8%	42.6%	45.1%	**42.2%**

Table 3: Evaluation of RLKF-trained models on various knowledge and reasoning related tasks: MMLU (Hendrycks et al., 2020), WinoGrande (Sakaguchi et al., 2021), ARC (Chollet, 2019), BBH (Suzgun et al., 2022), GSM8K (Cobbe et al., 2021), MATH (Hendrycks et al., 2021), C-Eval (Huang et al., 2023), CMMLU(Li et al., 2023a). Tasks are evaluated by the open-source evaluation tool TLEM (SUSTech, 2023), employing a 0-shot setting with greedy generation.

each for Chinese and English. To add variety to the questions, we have also modified the prompt to include multiple choice question types. Since our approach relies on the internal knowledge of the models and the boundaries of the internal knowledge are different for each model, we need to perform automatic annotation for each model individually. The generated responses are labeled using DreamCatcher to obtain factual preference data. The statistics of the factual preference data are shown in Table 7.

5.2 RLKF Training

We train the reward model using the factual preference data in Table 7. To maintain the generalization of the RM, we include same amount of general purpose data as the wiki-QA data in the training. Accuracy of the trained RM on factual preference data test set are shown in Table 2. Interestingly, the reward model is able to quickly achieve high accuracy for both known/unknown categories during training, suggesting that reward model may utilize the internal knowledge state of the initial model to determine whether the uncertainty response should be preferred.

Using the trained reward model, the RL process optimizes policy model using the PPO algorithm, where policy model is initialized from the same base model as reward model. The detailed training settings and hyper-parameters are described in A.4.

We conduct an evaluation of the trained model, focusing on its factuality and truthfulness. A comparative analysis of the models is performed between pre- and post- RLKF training on various tasks related to knowledge and reasoning as shown in Table 3. The RLKF-trained models demonstrate improvements on the majority of the benchmarks. While RLHF typically results in a reduction of benchmark performance, termed as 'alignment tax' (Askell et al., 2021), RLKF avoids this decline specifically on knowledge-related tasks, and even lead to improvements. Note that our training methodology does not employ any benchmark data, and the overall volume of training data utilized is small.

Models		TruthfulQA
Qwen-chat-14B	before	43.7%
	after	**49.1%**
Qwen-chat-7B	before	49.1%
	after	**50.3%**
Llama2-chat-13B	before	21.5%
	after	20.9%
Llama2-chat-7B	before	27.5%
	after	**28.3%**
Ziya-reader-13B	before	34.8%
	after	**37.9%**

Table 4: Evaluation of RLKF-trained models on TruthfulQA, again using TLEM (SUSTech, 2023), employing a 0-shot setting with greedy generation.

Regarding the truthfulness of trained models, we evaluated their performance using the widely recognized TruthfulQA task. Notably, all models, with the exception of llama2-chat-13B, show increase in honesty, as shown in Table 4.

6 Related Work

Hallucination in large language models (LLMs) has been the focal point of research, spanning its causes, detection, and mitigation. Our work relates to all three aspects.

Causes of hallucination: Studies have linked LLM hallucination to various causes. McKenna et al. (2023) ascribes it to memorization of training data, indicating a direct correlation between the training data and the resultant hallucination. Other works such as Schulman (2023) pinpoint improper model fine-tuning as contributive, and Perez et al. (2022) argues that RLHF induce model "sycophancy" which in turn degrades honesty.

Other studies link hallucinations to the generation process. For example, Lee et al. (2022) suggests that sampling-induced randomness could be responsible. One perspective provided by Chuang et al. (2023) proposes that "lower-level" prior layer information might overshadow factual information from subsequent layers. Furthermore, some works relate hallucinations to the overconfidence of LLMs (Ren et al., 2023).

Hallucination detecting: In terms of detecting hallucination, the consistency of multiple generations has been recognized as an effective indicator. SelfCheckGPT (Manakul et al., 2023) capitalizes on the consistent nature of internal knowledge-based generations compared to the variable nature of hallucination, propose several consistency checks to identify hallucinations. The idea is echoed by Agrawal et al. (2023), who suggest evaluating the generation consistency of generated references to spot hallucination. Similarly, Elaraby et al. (2023) proposes a metric involving the calculation of sentence-level entailment between response pairs as a measure of hallucination.

Employing large language models (LLMs) to recognize their own hallucinations has been suggested in Saunders et al. (2022), suggesting that discrimination is more accurate than generation for LLMs (G-D gap). This notion is furthered by Kadavath et al. (2022) and Agrawal et al. (2023) by directly prompting LLMs to assess the validity of their own output.

Another approach examines the factualness of statements by analyzing the model's internal representation. Studies Li et al. (2023b) and Burns et al. (2022) identify a "factualness" direction in the model's internal representation, with Li et al. (2023b) showcasing a high accuracy attention head through linear probing, and Burns et al. (2022) locating factualness direction through consistency of facts. Additionally, Kadavath et al. (2022) trains the model to predict the probability that it knows. Base on these works, we shifts focus onto the model's self-evaluation of knowledge state.

Hallucination mitigation: The common approach of hallucination mitigation involves enhancing the model with additional information. Elaraby et al. (2023) propose the use of larger models to provide additional information when hallucinations is detected.

Some research efforts focus on the optimization of decoding strategies to address hallucinations. Chuang et al. (2023) suggests that contrastive decoding can augment the factualness of model generation. Li et al. (2023b) enhances factualness by adjusting the output of attention heads along the direction of factualness during inference. Our work seeks to optimizes the utilization of the model's internal knowledge state, in line with the direction proposed by Schulman (2023) leveraging reinforcement learning to tackle hallucinations.

7 Conclusion

In our research, we thoroughly explore the capability of large language models (LLMs) to discern and express their internal knowledge, a key factor in mitigating factual hallucinations and ensuring reliable applications. Our research, manifested through a series of knowledge probing experiments, identifies the model's self-awareness of its knowledge state. We released the open-source tool DreamCatcher which scores and annotates the degree of hallucination in the LLM's response to knowledge-oriented question and rank responses based on their factuality.

We further validated our findings through the Reinforcement Learning from Knowledge Feedback (RLKF) training framework. Utilizing DreamCatcher to annotate factual preference data, we train a reward model and leveraging reinforcement learning to enhances LLM's factuality and truthfulness. Our results indicate RLKF's effectiveness in improving the model's utilization of its internal knowledge state, enhancing its performance in various knowledge and honesty related tasks. We posit that RLKF is a promising solution to address LLM's hallucination issues and, combined with RLHF, offers significant potential for enhancing the model's overall capabilities.

8 Limitations

Data limitation: Our Reinforcement Learning from Knowledge Feedback (RLKF) training relies on a relatively limited quantity and variety of data used. The factual question-answer data employed in our experiments predominantly resulted from using GPT3.5 to generate question-answer pairs from Wikipedia passages. Although this approach guarantees high factual precision and includes an extensive range of long-tail facts, it restricts diversity in writing style.

Given the time and cost considerations associated with the use of GPT api, the volume of data was also somewhat restricted. To enhance RLKF training, prospective research might contemplate compiling more intricate factual question-answer data that reflect real-world conditions.

Integration of Alternative Optimization Techniques: The essence of the RLKF framework lies in optimizing for factual preferences. After acquiring factual preference data, we opted for the Proximal Policy Optimization (PPO) method for optimization, given its demonstrated efficacy within the existing Reinforcement Learning from Human Feedback (RLHF) framework.

However, various other potential optimization methods exist, including reject sampling, DPO, mixed data supervised fine-tuning, among others. We anticipate future research will creatively incorporate factual preference data into their respective training frameworks, contributing to a comprehensive understanding of the LLM illusion phenomenon.

References

Ayush Agrawal, Lester Mackey, and Adam Tauman Kalai. 2023. Do language models know when they're hallucinating references? *arXiv preprint arXiv:2305.18248*.

Amanda Askell, Yuntao Bai, Anna Chen, Dawn Drain, Deep Ganguli, Tom Henighan, Andy Jones, Nicholas Joseph, Ben Mann, Nova DasSarma, et al. 2021. A general language assistant as a laboratory for alignment. *arXiv preprint arXiv:2112.00861*.

Amos Azaria and Tom Mitchell. 2023. The internal state of an llm knows when its lying. *arXiv preprint arXiv:2304.13734*.

Jinze Bai, Shuai Bai, Yunfei Chu, Zeyu Cui, Kai Dang, Xiaodong Deng, Yang Fan, Wenbin Ge, Yu Han, Fei Huang, et al. 2023. Qwen technical report. *arXiv preprint arXiv:2309.16609*.

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

Collin Burns, Haotian Ye, Dan Klein, and Jacob Steinhardt. 2022. Discovering latent knowledge in language models without supervision. *arXiv preprint arXiv:2212.03827*.

François Chollet. 2019. On the measure of intelligence. *arXiv preprint arXiv:1911.01547*.

Aakanksha Chowdhery, Sharan Narang, Jacob Devlin, Maarten Bosma, Gaurav Mishra, Adam Roberts, Paul Barham, Hyung Won Chung, Charles Sutton, Sebastian Gehrmann, et al. 2023. Palm: Scaling language modeling with pathways. *Journal of Machine Learning Research*, 24(240):1–113.

Yung-Sung Chuang, Yujia Xie, Hongyin Luo, Yoon Kim, James Glass, and Pengcheng He. 2023. Dola: Decoding by contrasting layers improves factuality in large language models. *arXiv preprint arXiv:2309.03883*.

Karl Cobbe, Vineet Kosaraju, Mohammad Bavarian, Mark Chen, Heewoo Jun, Lukasz Kaiser, Matthias Plappert, Jerry Tworek, Jacob Hilton, Reiichiro Nakano, et al. 2021. Training verifiers to solve math word problems. *arXiv preprint arXiv:2110.14168*.

Mohamed Elaraby, Mengyin Lu, Jacob Dunn, Xueying Zhang, Yu Wang, and Shizhu Liu. 2023. Halo: Estimation and reduction of hallucinations in open-source weak large language models. *arXiv preprint arXiv:2308.11764*.

Peter Hase, Mona Diab, Asli Celikyilmaz, Xian Li, Zornitsa Kozareva, Veselin Stoyanov, Mohit Bansal, and Srinivasan Iyer. 2023. Methods for measuring, updating, and visualizing factual beliefs in language

models. In *Proceedings of the 17th Conference of the European Chapter of the Association for Computational Linguistics*, pages 2706–2723.

Dan Hendrycks, Collin Burns, Steven Basart, Andy Zou, Mantas Mazeika, Dawn Song, and Jacob Steinhardt. 2020. Measuring massive multitask language understanding. *arXiv preprint arXiv:2009.03300*.

Dan Hendrycks, Collin Burns, Saurav Kadavath, Akul Arora, Steven Basart, Eric Tang, Dawn Song, and Jacob Steinhardt. 2021. Measuring mathematical problem solving with the math dataset. *arXiv preprint arXiv:2103.03874*.

Yuzhen Huang, Yuzhuo Bai, Zhihao Zhu, Junlei Zhang, Jinghan Zhang, Tangjun Su, Junteng Liu, Chuancheng Lv, Yikai Zhang, Jiayi Lei, et al. 2023. C-eval: A multi-level multi-discipline chinese evaluation suite for foundation models. *arXiv preprint arXiv:2305.08322*.

He Junqing, Pan Kunhao, Dong Xiaoqun, Song Zhuoyang, Liu Yibo, Liang Yuxin, Wang Hao, Sun Qianguo, Zhang Songxin, Xie Zejian, et al. 2023. Never lost in the middle: Improving large language models via attention strengthening question answering. *arXiv preprint arXiv:2311.09198*.

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

Andreas Köpf, Yannic Kilcher, Dimitri von Rütte, Sotiris Anagnostidis, Zhi-Rui Tam, Keith Stevens, Abdullah Barhoum, Nguyen Minh Duc, Oliver Stanley, Richárd Nagyfi, et al. 2023. Openassistant conversations–democratizing large language model alignment. *arXiv preprint arXiv:2304.07327*.

Nayeon Lee, Wei Ping, Peng Xu, Mostofa Patwary, Pascale N Fung, Mohammad Shoeybi, and Bryan Catanzaro. 2022. Factuality enhanced language models for open-ended text generation. *Advances in Neural Information Processing Systems*, 35:34586–34599.

Haonan Li, Yixuan Zhang, Fajri Koto, Yifei Yang, Hai Zhao, Yeyun Gong, Nan Duan, and Timothy Baldwin. 2023a. Cmmlu: Measuring massive multitask language understanding in chinese. *arXiv preprint arXiv:2306.09212*.

Kenneth Li, Oam Patel, Fernanda Viégas, Hanspeter Pfister, and Martin Wattenberg. 2023b. Inference-time intervention: Eliciting truthful answers from a language model. *arXiv preprint arXiv:2306.03341*.

Potsawee Manakul, Adian Liusie, and Mark JF Gales. 2023. Selfcheckgpt: Zero-resource black-box hallucination detection for generative large language models. *arXiv preprint arXiv:2303.08896*.

Nick McKenna, Tianyi Li, Liang Cheng, Mohammad Javad Hosseini, Mark Johnson, and Mark Steedman. 2023. Sources of hallucination by large language models on inference tasks. *arXiv preprint arXiv:2305.14552*.

Long Ouyang, Jeffrey Wu, Xu Jiang, Diogo Almeida, Carroll Wainwright, Pamela Mishkin, Chong Zhang, Sandhini Agarwal, Katarina Slama, Alex Ray, et al. 2022. Training language models to follow instructions with human feedback. *Advances in Neural Information Processing Systems*, 35:27730–27744.

Ethan Perez, Sam Ringer, Kamilė Lukošiūtė, Karina Nguyen, Edwin Chen, Scott Heiner, Craig Pettit, Catherine Olsson, Sandipan Kundu, Saurav Kadavath, et al. 2022. Discovering language model behaviors with model-written evaluations. *arXiv preprint arXiv:2212.09251*.

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. *arXiv preprint arXiv:2305.18290*.

Ruiyang Ren, Yuhao Wang, Yingqi Qu, Wayne Xin Zhao, Jing Liu, Hao Tian, Hua Wu, Ji-Rong Wen, and Haifeng Wang. 2023. Investigating the factual knowledge boundary of large language models with retrieval augmentation. *arXiv preprint arXiv:2307.11019*.

Keisuke Sakaguchi, Ronan Le Bras, Chandra Bhagavatula, and Yejin Choi. 2021. Winogrande: An adversarial winograd schema challenge at scale. *Communications of the ACM*, 64(9):99–106.

William Saunders, Catherine Yeh, Jeff Wu, Steven Bills, Long Ouyang, Jonathan Ward, and Jan Leike. 2022. Self-critiquing models for assisting human evaluators. *arXiv preprint arXiv:2206.05802*.

John Schulman. 2023. Reinforcement learning from human feedback: Progress and challenges, 2023.

SUSTech. 2023. Introducing tlem: The future of language model evaluation.

Mirac Suzgun, Nathan Scales, Nathanael Schärli, Sebastian Gehrmann, Yi Tay, Hyung Won Chung, Aakanksha Chowdhery, Quoc V Le, Ed H Chi, Denny Zhou, et al. 2022. Challenging big-bench tasks and whether chain-of-thought can solve them. *arXiv preprint arXiv:2210.09261*.

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

Hugo Touvron, Louis Martin, Kevin Stone, Peter Albert, Amjad Almahairi, Yasmine Babaei, Nikolay Bashlykov, Soumya Batra, Prajjwal Bhargava, Shruti

Bhosale, et al. 2023b. Llama 2: Open foundation and fine-tuned chat models. *arXiv preprint arXiv:2307.09288*.

Shitao Xiao, Zheng Liu, Peitian Zhang, and Niklas Muennighoff. 2023. C-pack: Packaged resources to advance general chinese embedding.

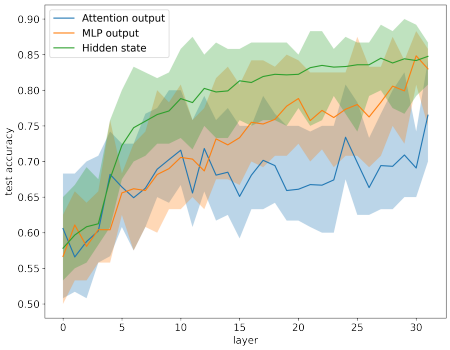
Muru Zhang, Ofir Press, William Merrill, Alisa Liu, and Noah A Smith. 2023a. How language model hallucinations can snowball. *arXiv preprint arXiv:2305.13534*.

Yue Zhang, Yafu Li, Leyang Cui, Deng Cai, Lemao Liu, Tingchen Fu, Xinting Huang, Enbo Zhao, Yu Zhang, Yulong Chen, et al. 2023b. Siren's song in the ai ocean: A survey on hallucination in large language models. *arXiv preprint arXiv:2309.01219*.

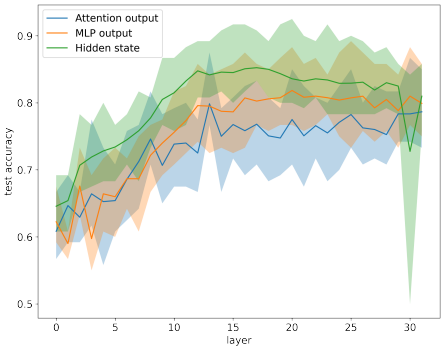
A Appendix

A.1 Example of wiki-QA Instruction

A.2 More probing results



(a) Llama2-chat-7B



(b) Qwen-chat-7B

Figure 4: Accuracy of knowledge state probing in 7B models. The light-colored area in the figure shows the range of accuracy for ten repetitions of the experiment, and the solid line shows the mean accuracy.

Instruction template:
Based on the following Wikipedia article snippet, ask a knowledge-based question and provide a corresponding answer.
Article snippet:
{Wikipedia passage}
Requirements:
1. there is a unique correct answer to the question, and the answer can be found in the given article fragment.
2. the question can be answered independently of the article fragment, i.e. the answer to the question cannot depend on contextual information, e.g. a question about a character in a literature needs to specify the work to which the character belongs, and a question such as "What is the article about?" cannot be asked.
3. Provide the question, answer, and category (e.g., literature, physics, etc.) at the same time, and reply in the following format: {"question":question,"answer":answer,"type":category}.
If you are unable to ask a question that meets the above requirements, you can simply reply "Unable to ask".
Reply:
Wikipedia passage:
House Arrest (1996 film) House Arrest is a 1996 American comedy film directed by Harry Winer, written by Michael Hitchcock, and starring Jamie Lee Curtis, Kevin Pollak, Jennifer Tilly, Christopher McDonald, Wallace Shawn, and Ray Walston with supporting roles done by Kyle Howard, Amy Sakasitz, Mooky Arizona, Russel Harper, and an up-and-coming Jennifer Love Hewitt. It tells the story of two children who trap their parents in their basement upon their plans for a separation as the other children they know get involved by trapping their respective problem parents as well. The film was released on August 14, 1996 and went on to gross just over $7 million at the box office. The film was panned by critics. The film was shot at various locations in the U.S. states of California and Ohio. Monrovia, California was the location for several exterior house scenes while most interior shots were done at the CBS/Radford lot in Studio City, California. The story was set in Defiance, Ohio, although another town, Chagrin Falls, Ohio, actually doubled for it.
GPT3.5 response:
{"question":"Who directed the film House Arrest?","answer":"Harry Winer","type":"film"}

Table 5: Example of instruction and corresponding GPT3.5 output of English wiki-QA.

Instruction template:
根据下面的维基百科文章片段，提出一个简短的知识型问题并给出对应回答，要求这个问题存在唯一正确答案，并且答案可以在给出的文章片段中找到。
文章片段：
{Wikipedia passage}
问题需要在脱离文章片段的情况下仍能够被回答，例如针对文学作品中人物提问需要指明所属的作品，以免引起歧义。问题的回答不能依赖于上下文的信息，不能提出类似“这篇文章的内容是什么？”的问题。
同时给出问题，回答和问题分类（比如文学类或物理类等），按如下格式回复：{"question":问题,"answer":回答,"type":分类}。如果无法提出满足上述要求的问题，可以直接回复“无法提问”。
回复：
Wikipedia passage:
M25
M25，也称为IC 4725，是一个由恒星组成，在南天人马座的疏散星团。Philippe Loys de Chéseaux在1745年对这个星团进行了第一次有记录的观测，查尔斯·梅西耶1764年将它收录进他的星云天体清单[6]。这个星团位于模糊的特征附近，因此有一条暗带通过中心附近[3]。
M25距离地球大约2,000光年，年龄约为6,760万岁[2]。这个星团在空间的维度大约是13光年，估计质量是1,937 M，其中大约24%是星际物质[4]。星团成员中的人马座U是一颗分类为造父变星的变星[7]，还有两颗红巨星，且其中一颗是联星系统[8]。
GPT3.5 response:
{"question":"M25是位于哪个星座的疏散星团？","answer":"南天人马座","type":"天文学"}

Table 6: Example of instruction and corresponding GPT3.5 output of Chinese wiki-QA.

A.3 Statistics of factual preference data

Model	Total	Known	Unknown	Mixed
Qwen-chat-14B	12799	49%	43%	8%
Qwen-chat-7B	7201	52%	40%	8%
Llama2-chat-13B	6600	48%	44%	8%
Llama2-chat-7B	6680	45%	45%	10%
Ziya-reader-13B	12558	49%	41%	10%

Table 7: Statistics of factual preference data and percentage of each knowledge state category used for reward modeling. The Llama2 models use English-only wiki-QA data, Qwen-chat-7B uses Chinese-only data, and Qwen-chat-14B and Ziya-reader-13B use a mixture of English and Chinese data.

A.4 RLKF Training details

We use the AdamW optimizer, with $\beta_1 = 0.9$, $\beta_2 = 0.99$, eps = $1e-5$ for all models. The learning rate for reward model training is $5e-6$ with 1% warmup and linear decay scheduler. The batch size is 16 for 13/14B models and 64 for 7B models. We train the reward model for 1 epoch. For PPO training, we use learning rate of $1e-6$ with cosine scheduler. The batch size is 32 for 13/14B models and 64 for 7B models. We set the KL penalty to 0 for all models.

A.5 More Observation

We observe that, some of the responses to the unknown questions are indicating uncertainty in RLHF-trained models, but there is also a significant percentage of responses that are hallucinations. This indicates an increase in model honesty achieved through RLHF, but there is still room for improvement.