Instructing Large Language Models to say "I don't know"

A survey on self-knowledge and replication of R-Tuning results

Politecnico di Milano
Filippo Wang, Leonardo Scappatura

10-06-2025

Abstract

Large language models (LLMs) have achieved remarkable performance by memorizing and synthesizing vast knowledge, but they still hallucinate generating confident answers not grounded in facts when faced with queries beyond their knowledge. Modern LLMs are typically trained to always produce an answer. Standard next-token prediction will continue a sentence or dialogue turn even if the correct answer is not in the models training data or context.

In this work we put together a comprehensive literature review on self-knowledge in LLMs, covering its definitions, methods for its evaluation, and strategies to improve such skill. We conclude with some thoughts about future research in this area. Moreover, we partially reproduce the results of "R-Tuning: Instructing Large Language Models to Say 'I Don't Know'" an article that received an Outstanding Paper Award at NAACL 2024

PhD course on Large Language Models
Politecnico di Milano

POLITECNICO
MILANO 1863

Table of Contents

1 Introduction

The concept of self-knowledge has emerged as an important skill in the development of large language models (LLMs), playing a critical role in aligning these systems with human values and expectations (Askell et al., 2021). Ideally, an LLM should be capable of recognizing its own limitations and refrain from producing misleading answers when presented with questions beyond its understanding. This capability becomes of paramount importance in high-stakes fields such as healthcare (Thirunavukarasu et al., 2023), law (Dahl et al., 2024), and finance (Li et al., 2024), where models that confidently state wrong facts can have serious consequences.

One of the challenges in this area is the lack of consensus on how self-knowledge should be defined in LLMs. The term is used variably across studies, such as negative rejection, truthfulness, confidence, syncopacy, model calibration, and so so on. Additionally, evaluating and improving honesty is inherently model-dependent, as each LLM has its own distribution of known and unknown information.

We present a comprehensive survey of literature related to self-knowledge in LLMs. We begin by clarifying commonly accepted definitions regarding self-knowledge, followed by an overview of current evaluation methodologies 2. We then explore recent advancements aimed at enhancing self-knowledge through both training-free and training-based approaches 3. The review concludes with a discussion of future research opportunities in this area 4.

2 Self-knowledge

Honesty broadly means that a models statements faithfully reflect what the model believes to be true, while truthfulness means those statements are actually true in the real world (Evans et al., 2021). An honest language model should never knowingly say something contrary to its own knowledge, whereas a truthful model should avoid saying any factual false statements. In practice, these properties can sometimes diverge a language model might be honest yet still say untruthful facts if it is mistaken in its internal beliefs. One of the most widely supported views in the literature is that an honest LLM should satisfy two key properties: self-knowledge and self-expression. **Self-knowledge** refers to the models capacity to understand the boundaries of its own competence, recognizing what information it can reliably provide and where its knowledge is limited or uncertain. A model with strong self-knowledge should be able to assess when it lacks sufficient information to answer a question accurately and, in such cases, signal uncertainty or refrain from answering altogether. This ability is especially important in avoiding misleading or fabricated responses, which are often referred to as hallucinations.

2.1 Binary classification

Many benchmarks were developed to evaluate the model's ability to classify correctly what it know and what it doesn't. We are able to categorize them in **model-agnostic** evaluations and **model-specific** evaluations. Some of the most notable model-agnostic

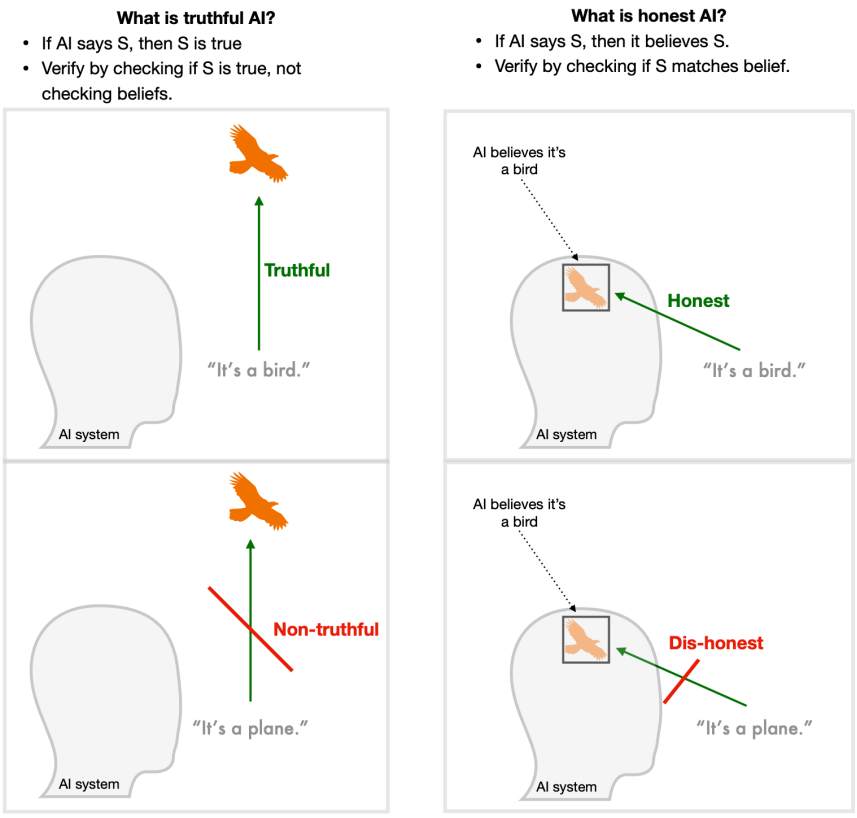

Figure 1: If the AI is honest, then the statement matches its belief. (Evans et al., 2021)

benchmarks are SelfAware (Yin et al., 2023), KUQ (Amayuelas et al., 2024), Unknown-Bench (Liu et al., 2024), HoneSet (Gao et al., 2024) and BeHonest (Chern et al., 2024). These benchmarks assume the knowledge of the models. Known questions are usually generated from sources like Wikipedia while unknown questions are created using some heuristics, for example "Latest news in the the stock market" to generate questions like "Show me recent the indexes with highest traded volume".

Model-specific evaluation benchmarks are usually built based on the models performance (e.g. accuracy) on a set of true {question, answer} pairs. A notable model-specific benchmark is Idk (Cheng et al., 2024), in particular they ask an LLM multiple times the same question, and if the accuracy of the output is consistent, the question is labeled as *known*. Another recent evaluation framework was developed from (Zhang et al., 2024), starting from any {Q,A} dataset $D = \{(q_1, a_1), (q_2, a_2), ...\}$ we can construct a certain dataset D_1 and an uncertain dataset D_0. Intuitively, the D_0 contains the (q, a) pairs that the model got wrong, while D_1 contains the ones that it got correctly. Answers in D_1 are concatenated with *'Are you sure you accurately answered the question based on your internal knowledge?* ***I am sure****'*, while answers in D_0 are concatenated with *'Are you sure you accurately answered the question based on your internal knowledge?* ***I am unsure****'*. Here are a couple of examples from refusal-aware dataset constructed starting from ParaRel (Elazar et al., 2021) with open_llama_3b (Grattafiori et al., 2024) :

> {"Question: What field does Max Weber work in?
> Answer: sociology. Are you sure you accurately answered the question based on your internal knowledge? I am sure."}

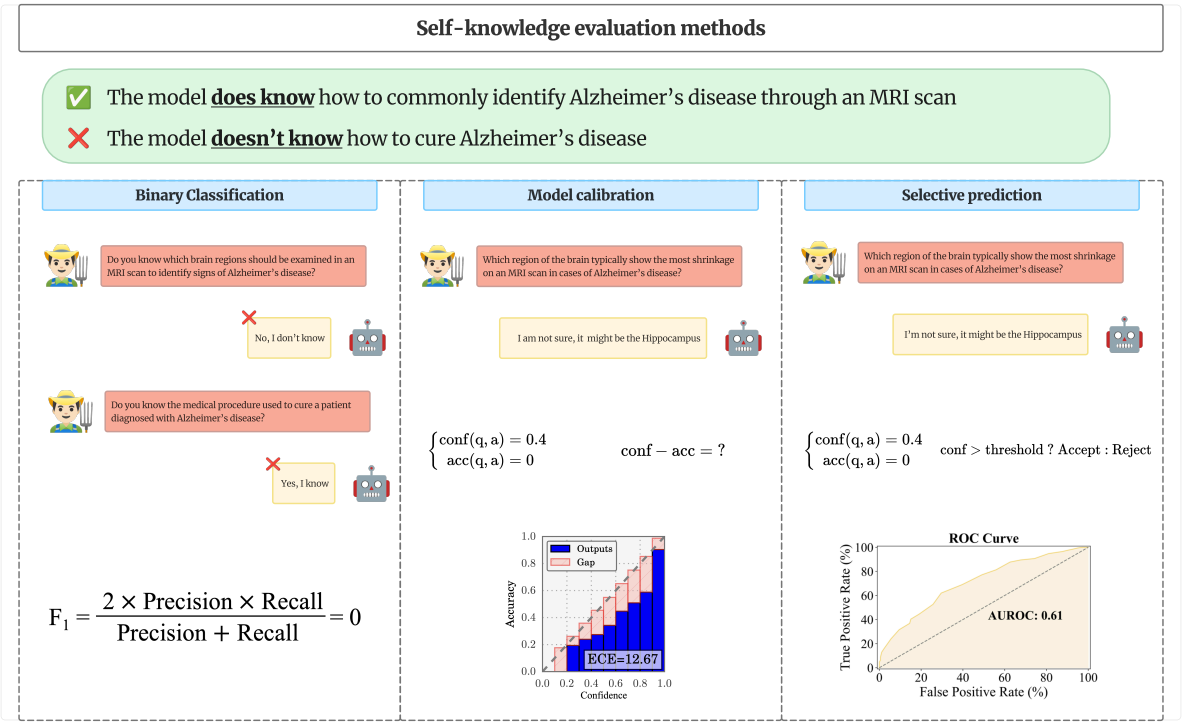

Figure 2: Illustrations of self-knowledge evaluation. 'conf' indicates the LLMs confidence score and 'acc' represents the accuracy of the response. Expected Calibration Error (ECE) plot courtesy of (Guo et al., 2017)

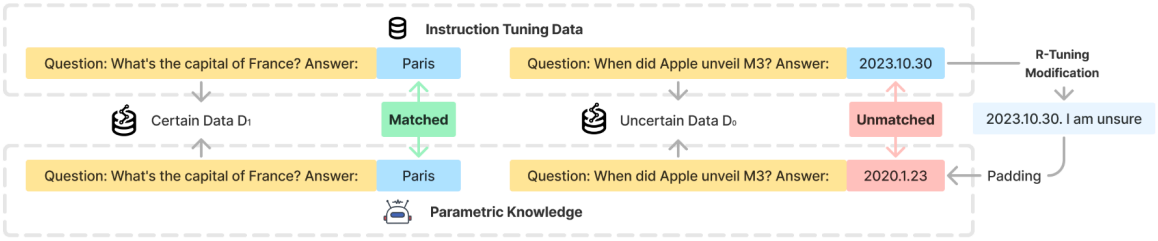

Figure 3: Illustration of R-Tuning to construct refusal-aware datasets D0 and D1. Courtesy of (Zhang et al., 2024)

> {"Question: Which country does Bonn serve as the capital of?
> Answer: Germany. Are you sure you accurately answered the question based on your internal knowledge? I am unsure."}

From these refusal-aware dataset we can fine-tune our chosen LLM with standard procedures. The model takes a sequence $t_1, t_2, \ldots, t_T$ consisting of the questions and answers, and predicts the answer part based on each question.

To assess the performance of these models, the typical evaluation metrics used are precision, recall, and F1 score.

2.2 Model calibration

In order to extract confidence scores $\text{conf}(q, \hat{a})$ from LLMs we need to apply some confidence elicitation methods (Geng et al., 2024). We say that a model is calibrated well

if the confidence score assigned to a prediction accurately reflects the likelihood that the prediction is correct, formally:

$$P(\hat{a} = a \mid \text{conf}(q, \hat{a}) = p) = p, \quad \forall p \in [0, 1]. \tag{1}$$

where a is the correct answer and $\hat{a}$ is the model's prediction.

Two of the most widely used metrics to assess the calibration of a model are the Brier Score and the Expected Calibration Error (ECE). The first one is just the mean squared error between accuracy and confidence level across a test {Q,A} dataset

$$\text{Brier Score} = \frac{1}{N} \sum_{i=1}^{N} \left(\text{acc}(a_i, \hat{a}_i) - \text{conf}(q_i, \hat{a}_i)\right)^2 \tag{2}$$

while the ECE uses a 'bucketing' strategy. Basically, it divides the model's confidence predictions $\text{conf}(q, \hat{a}) \in [0, 1]$ into M buckets B_m. (e.g. for M=10 the buckets are $\{(0, 0.1), (0.1, 0.2), \dots\}$

$$\text{ECE} = \sum_{m=1}^{M} \frac{|B_m|}{N} \left|\text{acc}(B_m) - \text{conf}(B_m)\right| \tag{3}$$

where $|B_m|$ is the number of test examples in bucket B_m, $\text{acc}(B_m)$ is the average accuracy, and $\text{conf}(B_m)$ is the average confidence in that bucket. Intuitively, we aim to minimize the ECE. Note that the difference between acc and conf for a given bin represents the **calibration gap** (red bars in reliability diagrams) but the final expected error is weighted by the number of samples in each bucket.

2.3 Selective prediction

A natural way to filter the answers from an LLM is to discard the predictions below a certain confidence score, but in order for this method to be effective, the model must learn to assign high confidence scores to correct predictions and low confidence scores to wrong answers. The difference with respect to model calibration is that selective prediction measures the difference between the confidence scores and ground truth (correct and incorrect) predictions, whilst calibration focuses on matching the accuracy of the model with its confidence.

3 Improving self-knowledge

Many studies aim to improve the self-knowledge capabilities of LLMs. One line of research teaches them to answer *"I don't know"*. Another line of research focuses on the probability that the responses are correct. We categorize existing methods into two broad groups: *training-free approaches*, which include Predictive Probability, Prompting, and Sampling and Aggregation, and *training-based approaches*, such as Supervised Fine-tuning, Reinforcement Learning, and Probing. Take an overview of these methods in Fig. 5.

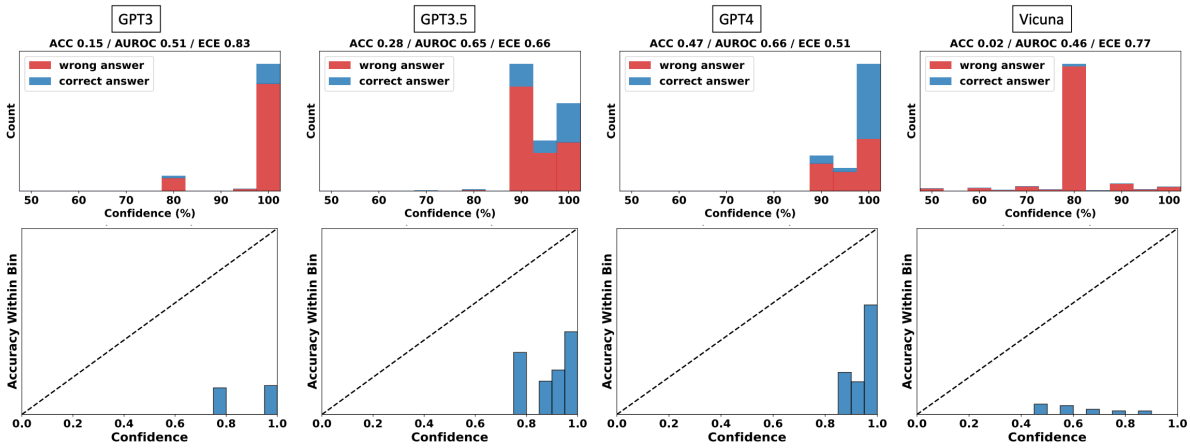

Figure 4: Empirical distribution (First row) and reliability diagram (Second row) of vanilla verbalized confidence across four models on GSM8K. It shows significant overconfidence. Figure courtesy of (Xiong et al., 2024)

3.1 Training-free approaches

Predictive Probability. Predictive probability, widely used in masked language model classification tasks (Xiao et al., 2022), is formalized in LLMs as:

$$\log p(y|x) = \sum_{t=1}^{T} \log p(y_t|y_{<t}, x) \tag{4}$$

where $\mathbf{x}$ is the prompt and $\mathbf{y}$ the output. Since this measure scales with output length T, a length-normalized version is commonly used.

Research shows that predictive probability is *well-calibrated* for multiple-choice tasks (T = 1), especially with more capable LLMs. However, this metric performs poorly for *free-form generation* (T > 1), as it captures lexical confidence rather than *semantic confidence*.To address this, some researchers reformulate free-form outputs as multiple-choice tasks, using sampled candidates and treating their predicted probabilities as confidence scores.

Prompting. A growing body of research explores prompting strategies to elicit self-knowledge from large language models (LLMs).

Self-evaluation approaches aim to estimate confidence by prompting the model to judge whether its answer is true or false. (Kadavath et al., 2022) introduce the P(True) method, which treats confidence estimation as a binary classification task. The model is prompted with a question and its answer, and the probability it assigns to "true" is interpreted as its confidence. Results show improved performance when P(True) is applied with multiple sampled answers. Building on this, (Zhao et al., 2024) propose a fact-and-reflection method, where the model first lists relevant facts, reasons through them, gives an answer, and then estimates confidence. While effective, these methods require extra inference steps, limiting efficiency, and recent studies suggest LLMs may struggle to judge their own answers accurately.

Inspired by conversational behavior, (Xiong et al., 2024) propose self-probing where

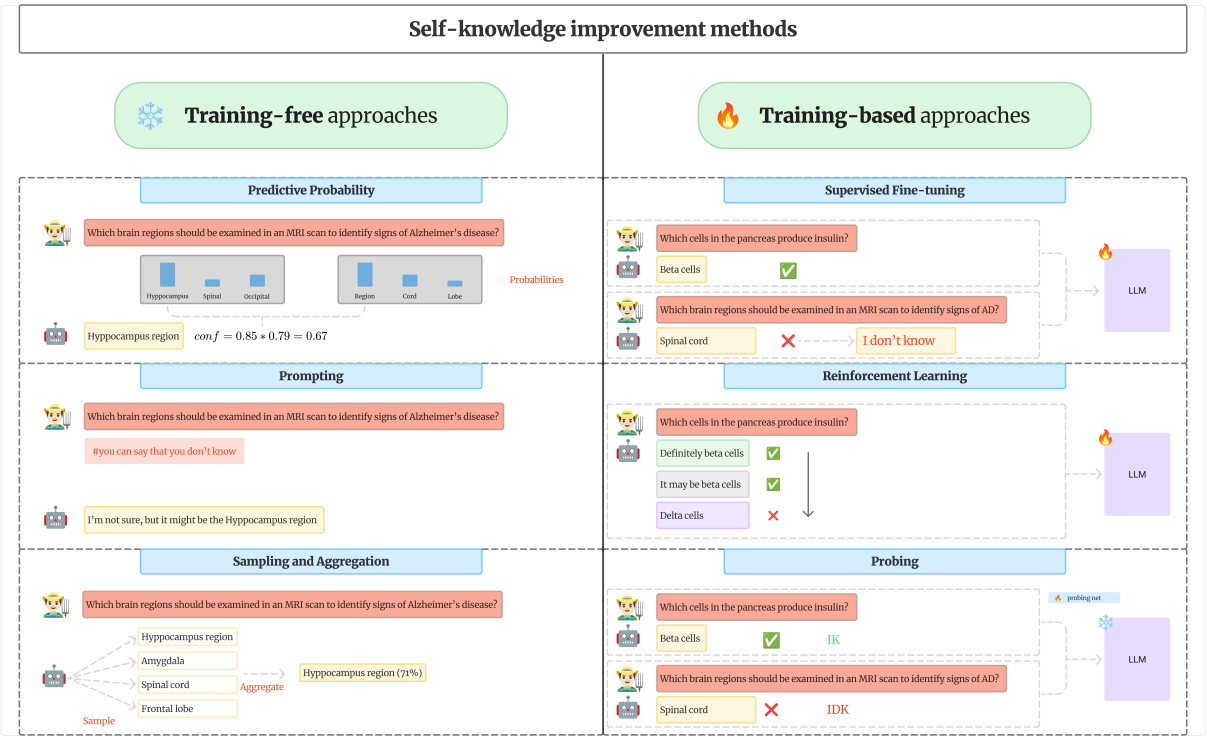

Figure 5: Illustration of different methods for improvement of self-knowledge. They are split in training-based and training-free approaches.

confidence is assessed in a multi-step, follow-up interaction, which breaks problems into parts with per-step confidence. However, the effectiveness of these techniques remains debated. LLMs often express misplaced certainty, while suggesting LLMs tend to *mimic human confidence expressions* rather than provide true self-assessment, often defaulting to high confidence as seen in training data.

Prompting strategies are appealing for their simplicity and performance, but some challenges remain. In particular, its unclear whether LLMs outward expressions of confidence genuinely reflect internal uncertainty or simply mirror human-like patterns from training data. Future work should focus on eliciting more faithful representations of model uncertainty and awareness.

Sampling and Aggregation. Many studies estimate confidence by analyzing the consistency of multiple model outputs. A common method is *temperature sampling*, which generates diverse outputs by adjusting randomness, others improve diversity by rephrasing prompts.

The main variation across works lies in *aggregation strategies* used to derive uncertainty. (Zhou et al., 2022) use *answer frequency* as a proxy for confidence, while others explore *entropy, confidence-weighted scores*, and related methods.

To capture *semantic* rather than *lexical* consistency, some approaches cluster outputs by entailment using a Natural Languange Inference (NLI) model and compute *semantic entropy*. Others extend this by exploring metrics like cluster *cluster count* and *degree matrices*, while some assess token-level *uncertainty* using NLI at each generation step.

For long-form outputs, recent methods propose decomposing responses into statements or claims, then using LLMs or semantic entropy to evaluate agreement across outputs.

While multi-sample approaches yield valuable confidence signals, they are computationally expensive and often rely on additional models for aggregation.

3.2 Training-based approaches

Supervised Fine-tuning. One research line trains models to say '*I don't know*' when lacking sufficient knowledge. The main challenge here is distinguishing between known and unknown questions. Some works address this by sampling multiple answers per question and checking their match with ground-truth; if the accuracy exceeds a threshold, the question is marked as known. Others adopt an unsupervised method using the models predictive probability.

However, these techniques struggle with evaluating long-form responses. To mitigate this, (Wan et al., 2024) reformulate instructions into multiple-choice questions to assess the models knowledge. In a different direction, others fine-tune models to predict the correctness likelihood of their own answers, experimenting with LoRA and probing techniques, and achieve strong results with as few as 1000 examples.

Reinforcement Learning from Human Feedback (RLHF). (Cheng et al., 2024) train models to abstain from answering questions they don't know, optimizing with DPO or PPO. They construct preference pairs based on model knowledge: correct answers are preferred over "I dont know," but when a model answers incorrectly, the preferred response is "*I dont know.*"

Other researchers simplify this by using LLMs to assess both honesty and helpfulness, resolving conflicts via DPO. (Xu et al., 2024) further enrich responses by training LLMs to provide numeric confidence scores and accompanying rationales, optimized with PPO to reward high confidence in correct answers and low confidence in incorrect ones.

Recent work also models human-AI interaction. One LLM ("speaker") is trained to calibrate its confidence so that another LLM ("listener") can decide whether to trust the response. Some approaches use direct preference optimization (DPO) to teach the speaker to express hedges or numeric confidence, rewarding both accepted correct answers and rejected incorrect ones. Others extend this by allowing the listener to ask follow-up questions based on the speakers long-form answer, using the log-likelihood of the listeners response to guide speaker training through reinforcement learning techniques like PPO.

RL methods significantly improve LLMs' ability to recognize and express uncertainty.

Probing. Instead of analyzing outputs, another research direction explores LLMs internal representations to assess self-knowledge. This is commonly done through *probing*, where a lightweight classifier is trained on the hidden states of a frozen LLM to perform tasks like truth detection or answerability .

(Kadavath et al., 2022) trained a value head to predict whether the model knows the answer to a free-form question, showing strong performance. Similarly, other used hidden states to distinguish true from false statements with accuracy ranging from 71% to 83%, suggesting LLMs internally encode truthfulness. Some findings also reveal a linear

separation between representations of true and false statements within the models embedding space. Additionally, probing query-token embeddings has been shown to forecast hallucinations even before generation begins.

A major limitation, however, is poor generalization to out-of-distribution inputs. Further gains are possible by combining probing and prompting approaches.

4 Challenges and future Research Directions

The rapid progress on LLM self-knowledge has uncovered several fundamental road-blocks that must be addressed before honest systems can be safely deployed at scale. Below, we distil the open questions most frequently highlighted in recent literature and outline promising avenues for future work.

4.1 Objective vs. Subjective Honesty

Current papers disagree on whether an LLM should be judged against **external truth** (objective honesty) or against the models **internal beliefs** (subjective honesty). The former better serves end-users who want factually correct answers, yet it requires extra supervision to label truth, especially when a models knowledge already exceeds human expertise. The latter is easier to optimiseone only needs to check whether the model believes its answer but risks letting a confidently wrong model appear honest. Reconciling these two perspectives, or designing hybrid benchmarks that report both scores, remains an open challenge.

4.2 Honesty in Instruction-Following and Long-Form Generation

Most honesty studies still focus on short factoid QA. Real applications, however, require multi-step reasoning, tool use, and dialogue. Designing benchmarks that measure whether a model admits ignorance mid-chain, calibrates confidence across paragraphs, and resists user pressure to fabricate during multi-turn conversations is a fertile research direction. Techniques such as hierarchical chain-of-thought, self-reflection prompts, or RLHF with conversation-level rewards could be leveraged here.

4.3 Honesty with In-Context Knowledge

Retrieval-augmented generation and long-context windows mean that answers often depend on **dynamic, external** documents rather than on static parameters. We still lack principled ways to measure whether a model faithfully grounds its claims in cited passages, or to make it refuse when the provided context is insufficient or contradictory. Developing grounding-aware confidence metrics and refusal strategies for RAG pipelines is therefore critical.

5 R-Tuning replication

5.1 Introduction

(Zhang et al., 2024) proposed a method to teach LLMs how to recognize when they dont know something and express uncertainty instead of guessing. In our project, we replicated this method and tested on ParaRel (Elazar et al., 2021) and MMLU (Hendrycks et al., 2021) datasets. Moreover, we tried to improve the evaluation process on the MMLU resulting in higher accuracy thanks to realistic guesses based on attention masks on tokens.

The idea behind R-Tuning (Refusal-Aware Fine-Tuning) is simple and can be broken down into the following four steps:

1. **Knowledge Identification.** Given a set of {question, answer} pairs, we first probe the model's knowledge. Data for which the model answers correctly is placed in the *certain dataset* D_1 , while data for which it answers incorrectly is placed in the *uncertain dataset* D_0
2. **Refusal-Aware Data.** This process is well explained in subsection 2.1
3. **Model Fine-Tuning.** The language model is fine-tuned on these new sequences. The goal is to teach it to generate not only the answer but also the subsequent, appropriate expression of confidence or uncertainty.
4. **Inference and Evaluation.** At test time, prompts follow the same conversational structure. Performance is evaluated based on both the accuracy of the answer and the validity of the model's self-reported certainty.

Our experiments were conducted using two prominent open-source language models: OpenLLaMA-3B (Touvron et al., 2023) and Qwen2.5-1.5B (Yang et al., 2024), the first one is present also in the original work, while the latter is a novel model we wanted to try. The fine-tuning process was orchestrated using the LMFlow framework (Diao et al., 2024), a versatile toolkit designed for large model customization. All training and inference tasks were executed on a single home server equipped with an NVIDIA RTX 4090 GPU.

5.2 Datasets

1. The **ParaRel** dataset is designed to evaluate whether language models can give consistent answers to paraphrased factual questions.

   For example, for the subject *Seinfeld*, the relation *original network* might include the paraphrased prompts: *"Seinfeld originally aired on [blank]"* and *"Seinfeld premiered on [blank]"*, both of which should be answered with *NBC*. The goal is to check whether the model provides consistent answers across these different formulations.

2. The **MMLU** (Massive Multitask Language Understanding) dataset tests how well language models perform on a wide range of academic and professional subjects.

   Each question comes with four answer choices, only one of which is correct. For instance, a sample question from the mathematics section could be:

   What is the derivative of $\sin(x)$*?*
   A. $\sin(x)$ B. $-\sin(x)$ C. $\cos(x)$ D. $-\cos(x)$

The correct answer is C: $\cos(x)$.

5.3 Fine Tuning

The fine-tuning process was managed using the LMFlow toolkit, following the methodology described in the R-Tuning paper. The models were trained on the refusal-aware datasets constructed from ParaRel and MMLU, with the primary goal of teaching them to append an uncertainty expression ("I am sure" or "I am unsure") after generating an answer. The training objective is a standard cross-entropy loss, selectively applied only to the answer and uncertainty tokens.

To improve memory efficiency this, we employed **QLoRA (Quantized Low-Rank Adaptation)**, a parameter-efficient fine-tuning technique.

LoRA works by freezing the large pre-trained weights of the model and injecting a pair of small, trainable rank-decomposition matrices into each target layer. During training, only these low-rank matrices are updated, reducing training time and drastically reducing the number of trainable parameters (and thus memory usage), compared to a full fine-tune. QLoRA enhances this efficiency by quantizing the frozen pre-trained weights to 4-bit precision, further minimizing the memory footprint.

Key hyperparameters for our QLoRA implementation were:

- **Quantization:** The base model was quantized to 4-bit.
- **LoRA Rank (r):** Set to 16. This defines the rank (and size) of the trainable matrices.
- **LoRA Alpha (α):** Set to 32. This is a scaling factor for the learned updates.
- **LoRA Dropout:** A dropout rate of 0.1 was applied to the LoRA layers to prevent overfitting.
- **Target Modules:** LoRA was applied to the attention mechanism's projection layers (q_proj, k_proj, v_proj, o_proj).
- **Fine tuning config:** We trained for 1 epoch, with a learning rate of 2e-4 and a batch size of 1.

5.4 Evaluation

The evaluation process measures how well the model can recognize when it knows an answer and when it does not. After fine-tuning is complete, the model is tested on a set of questions. For each question, the evaluation produces a tuple: $[y, \text{conf}(q, \hat{a}), \text{sure}(q, \hat{a})]$, where $y \in \{0, 1\}$ indicates whether the answer was correct, $\text{conf}(q, \hat{a})$ is the model's prediction confidence, and $\text{sure}(q, \hat{a})$ is the models self-reported confidence in its knowledge.

The primary metric used to evaluate performance is **Average Precisio**n (AP), which summarizes the precision-recall curve. This metric rewards models that assign higher confidence to correct answers while downweighting overconfident mistakes. AP is particularly useful for measuring the effectiveness of uncertainty-aware models, as it considers both accuracy and confidence calibration.

MMLU is a multiple-choice benchmark where each question has four possible answers (A, B, C, D). Prediction confidence is simply the probability assigned to the selected

option. However, to improve the robustness of the evaluation, a special handling is applied: if the models answer does not contain one of the four valid options, the models internal logits (i.e. the attention mask on the generated tokens) are used to select the option with the highest probability. This is not present in the original paper, and this change ensures that a prediction is always extracted and allows AP scores to reflect the models true internal belief, even when it fails to explicitly select an option. As a result, our AP scores on MMLU are much higher compared to the original paper.

5.5 Results

We report the AP (%) scores for R-Tuning and vanilla fine-tuning under single-task settings for both the ParaRel and MMLU datasets in Tables 1 and 2.

For the OpenLLaMA-3B model, our results closely match those reported in the original R-Tuning paper. R-Tuning consistently improves performance over vanilla fine-tuning in ParaRel (both ID and OOD), and slightly improves or remains competitive on MMLU.

Interestingly, results for the Qwen2.5-1.5B model show a different trend. While R-Tuning improves performance on the ParaRel dataset in both ID and OOD settings, it underperforms on MMLU ID compared to vanilla fine-tuning.

Dataset	Domain	Model	R-Tuning	Vanilla
ParaRel	ID	OpenLLaMA-3B	**93.23**	92.89
	OOD	OpenLLaMA-3B	**69.41**	68.42
MMLU	ID	OpenLLaMA-3B	**24.96**	24.19
	OOD	OpenLLaMA-3B	24.75	**26.08**

Table 1: **original.** Single-task experiments of R-Tuning and Vanilla on ParaRel and MMLU datasets with AP scores (%). ID and OOD denote in-domain and out-of-domain settings, respectively. Best results for each row are in **bold**.

Overall, our findings support the core effectiveness of R-Tuning while also demonstrating that evaluation choices especially in structured datasets like MMLU can significantly affect reported metrics.

Dataset	Domain	Model	R-Tuning	Vanilla
ParaRel	ID	OpenLLaMA-3B	**91.12**	81.11
		Qwen2.5-1.5B	**90.62**	63.86
	OOD	OpenLLaMA-3B	**61.16**	60.32
		Qwen2.5-1.5B	**66.11**	35.24
MMLU	ID	OpenLLaMA-3B	**32.44**	23.59
		Qwen2.5-1.5B	64.54	**80.28**
	OOD	OpenLLaMA-3B	**26.63**	25.38
		Qwen2.5-1.5B	65.41	**80.24**

Table 2: **ours.**

References

Amayuelas, Alfonso et al. (July 2024). *Knowledge of Knowledge: Exploring Known-Unknowns Uncertainty with Large Language Models*. DOI: 10.48550/arXiv.2305.13712.

Askell, Amanda et al. (Dec. 2021). *A General Language Assistant as a Laboratory for Alignment*. DOI: 10.48550/arXiv.2112.00861.

Cheng, Qinyuan et al. (Jan. 2024). *Can AI Assistants Know What They Don't Know?* DOI: 10.48550/arXiv.2401.13275.

Chern, Steffi et al. (July 2024). *BeHonest: Benchmarking Honesty in Large Language Models*. DOI: 10.48550/arXiv.2406.13261.

Dahl, Matthew et al. (Jan. 2024). "Large Legal Fictions: Profiling Legal Hallucinations in Large Language Models". In: *Journal of Legal Analysis* 16.1, pp. 64–93. ISSN: 2161-7201, 1946-5319. DOI: 10.1093/jla/laae003.

Diao, Shizhe et al. (May 2024). *LMFlow: An Extensible Toolkit for Finetuning and Inference of Large Foundation Models*. DOI: 10.48550/arXiv.2306.12420.

Elazar, Yanai et al. (May 2021). *Measuring and Improving Consistency in Pretrained Language Models*. DOI: 10.48550/arXiv.2102.01017.

Evans, Owain et al. (Oct. 2021). *Truthful AI: Developing and governing AI that does not lie*. DOI: 10.48550/arXiv.2110.06674.

Gao, Chujie et al. (Dec. 2024). *HonestLLM: Toward an Honest and Helpful Large Language Model*. DOI: 10.48550/arXiv.2406.00380.

Geng, Jiahui et al. (Mar. 2024). *A Survey of Confidence Estimation and Calibration in Large Language Models*. DOI: 10.48550/arXiv.2311.08298.

Grattafiori, Aaron et al. (Nov. 2024). *The Llama 3 Herd of Models*. DOI: 10.48550/arXiv.2407.21783.

Guo, Chuan et al. (Aug. 2017). *On Calibration of Modern Neural Networks*. DOI: 10.48550/arXiv.1706.04599.

Hendrycks, Dan et al. (Jan. 2021). *Measuring Massive Multitask Language Understanding*. DOI: 10.48550/arXiv.2009.03300.

Kadavath, Saurav et al. (Nov. 2022). *Language Models (Mostly) Know What They Know*. DOI: 10.48550/arXiv.2207.05221.

Li, Yinheng et al. (July 2024). *Large Language Models in Finance: A Survey*. DOI: 10.48550/arXiv.2311.10723.

Liu, Genglin et al. (Feb. 2024). *Examining LLMs' Uncertainty Expression Towards Questions Outside Parametric Knowledge*. DOI: 10.48550/arXiv.2311.09731.

Thirunavukarasu, Arun James et al. (Aug. 2023). "Large language models in medicine". In: *Nature Medicine* 29.8, pp. 1930–1940. ISSN: 1546-170X. DOI: 10.1038/s41591-023-02448-8.

Touvron, Hugo et al. (Feb. 2023). *LLaMA: Open and Efficient Foundation Language Models*. DOI: 10.48550/arXiv.2302.13971.

Wan, Fanqi et al. (Sept. 2024). *Knowledge Verification to Nip Hallucination in the Bud*. DOI: 10.48550/arXiv.2401.10768.

Xiao, Yuxin et al. (Oct. 2022). *Uncertainty Quantification with Pre-trained Language Models: A Large-Scale Empirical Analysis*. DOI: 10.48550/arXiv.2210.04714.

Xiong, Miao et al. (Mar. 2024). *Can LLMs Express Their Uncertainty? An Empirical Evaluation of Confidence Elicitation in LLMs*. DOI: 10.48550/arXiv.2306.13063.

Xu, Tianyang et al. (2024). "SaySelf: Teaching LLMs to Express Confidence with Self-Reflective Rationales". In: *Proceedings of the 2024 Conference on Empirical Methods in Natural Language Processing*. Miami, Florida, USA: Association for Computational Linguistics, pp. 5985–5998. DOI: 10.18653/v1/2024.emnlp-main.343.

Yang, An et al. (Sept. 2024). *Qwen2 Technical Report*. DOI: 10.48550/arXiv.2407.10671.

Yin, Zhangyue et al. (May 2023). *Do Large Language Models Know What They Don't Know?* DOI: 10.48550/arXiv.2305.18153.

Zhang, Hanning et al. (June 2024). *R-Tuning: Instructing Large Language Models to Say 'I Don't Know'*. DOI: 10.48550/arXiv.2311.09677.

Zhao, Xinran et al. (Sept. 2024). *Fact-and-Reflection (FaR) Improves Confidence Calibration of Large Language Models*. DOI: 10.48550/arXiv.2402.17124.

Zhou, Chunting et al. (Dec. 2022). *Prompt Consistency for Zero-Shot Task Generalization*. DOI: 10.48550/arXiv.2205.00049.