A New Perspective on Factual Knowledge Extraction in Large Language Models: Combining Fine-Tuning and Inference

Anonymous ACL submission

Abstract

Factual knowledge extraction aims to explicitly extract knowledge parameterized in pre-trained language models for application in downstream tasks. Recent work has been investigating the impact of fine-tuning on the factuality of large language models (LLMs). In this paper, we thoroughly study this impact through systematic experiments, with a particular focus on the factuality gap caused by unknown and known knowledge. We find that this gap is essentially a discrepancy between attention patterns, which can be influenced by both fine-tuning and in-context learning (i.e., few-shot learning and Chain of Thought (CoT)). Appropriate prompt design during the inference stage can even mitigate the factuality gap caused by fine-tuning. Therefore, we argue that both stages play essential roles in factual knowledge extraction, and that they need to be studied in combination. Finally, we seek to provide explanations and offer novel insights into factual knowledge extraction through the integration of fine-tuning and inference in LLMs.

1 Introduction

Pre-trained large language models (LLMs) store extensive parameterized knowledge (Meng et al., 2022; Petroni et al., 2019a; Allen-Zhu and Li, 2024), which can be extracted and applied to various downstream tasks through different prompt designs (Chen et al., 2024; Wang et al., 2024). However, querying LLMs with naturally phrased questions may increase the likelihood of generating incorrect answers, leading to model hallucinations (Zhang et al., 2024; Huang et al., 2025). Previous research has shown that fine-tuning LLMs can enhance their factuality (Wei et al., 2022a), yet the impact varies significantly depending on the dataset. For instance, Gekhman et al. (2024) and Ghosal et al. (2024) indicate that fine-tuning on well-established or popular knowledge improves model performance, while fine-tuning on unknown or unpopular data can have the opposite effect.

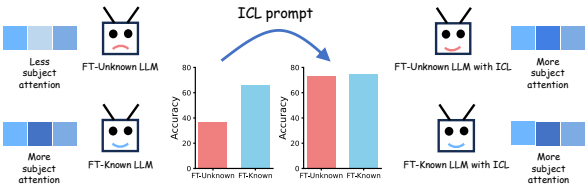

Figure 1: Overview: In-context learning (ICL) prompts can help reduce the factuality gap, as knowledge extraction is prompt-sensitive. Both fine-tuning and inference prompts are crucial for accurate knowledge retrieval.

Previous research has extensively explored how different fine-tuning datasets impact the factuality of LLMs. For example, Gekhman et al. (2024); Kazemi et al. (2023) draw empirical conclusions, summarizing and analyzing the factors of data characteristics, especially unknown knowledge, in fine-tuning. Joshi et al. (2024) propose a 'Persona' hypothesis to explain how fine-tuning affects factuality, while Ghosal et al. (2024) attempts to explain the cause of this factuality gap from the perspective of changes in attention distribution during the fine-tuning process. In this work, however, we find that this factuality gap is highly fragile. Modifying the inference-stage prompt, such as through few-shot examples (Brown et al., 2020) or chain-of-thought (CoT) (Wei et al., 2022b), can significantly reduce or even reverse the gap. Our work suggests that the factuality gap requires further investigation and a deeper understanding.

To gain a deeper understanding of the factuality gap caused by fine-tuning data, we pose the following three intriguing research questions: **RQ1:** *What is the impact of unknown knowledge on the factuality of large language models?* **RQ2:** *Does this factuality gap always exist?* **RQ3:** *Can the factuality gap be easily mitigated?* To address these questions, we design a series of experiments. We select two types of models, the LLama-3.1-8B

(Dubey et al., 2024) and Mistral-7B-v0.3 (Jiang et al., 2023), in both their *Base* and *Instruct* versions, and conduct experiments on two task categories: question answering (QA) and open-ended generation. These experiments allow us to answer the above questions. After that, we attempt offer insights to explain the related phenomena we have observed, including some hypotheses and theoretical derivations. Our main contributions can be summarized as follows:

- Through extensive experiments, we validate the factuality gap caused by fine-tuning on unknown knowledge and confirm it diminishes as the test set's distribution distance increases
- We identify an effective method to mitigate the factuality gap by carefully designing prompts during the inference stage. Our findings emphasize the crucial role of prompt design in the parameterized extraction of factual knowledge.
- Building on our observations, we conduct an in-depth analysis of the factuality gap and offer a deeper understanding from the perspective of attention patterns.

2 Related Works

2.1 Factual Knowledge Extraction in LLM

LLMs store extensive world knowledge within their parameters, and ineffective extraction is a major cause of model hallucinations (Kandpal et al., 2023; Mallen et al., 2023). Therefore, understanding knowledge extraction is crucial for improving LLM efficiency and performance. Allen-Zhu and Li (2024) integrates pretraining and fine-tuning to highlight the importance of data augmentation for extractable knowledge. Yin et al. (2024) introduces the concept of a *knowledge boundary*, where knowledge that cannot be correctly accessed under any expression is considered outside the model's boundary. While prior work focuses on either pretraining and fine-tuning phases or extraction during inference, our study combines both model fine-tuning and inference to offer a more comprehensive analysis of factual knowledge extraction.

2.2 Finetuning on Unknown Knowledge Encourage Hallucination

The unknown knowledge refers to information that is either unpopular or unfamiliar, indicating that the pre-trained model has limited exposure to it or struggles to extract it. Recent studies have explored the impact of fine-tuning on such knowledge and its effect on model factuality. Kang et al. (2024) suggest that unfamiliar examples in the fine-tuning dataset affect how the model handles unfamiliar test instances, but they do not address how these examples influence the overall factuality of the model. Gekhman et al. (2024) empirically demonstrate that fine-tuning on unknown knowledge negatively impacts factuality, attributing this to overfitting on such data during training. Ghosal et al. (2024) show that lesser-known facts, poorly stored during pretraining, lead to worse factuality compared to fine-tuning on well-known facts. They also provide a theoretical analysis, linking the effect to changes in entity attention during fine-tuning. Building on these studies, we further examine how inference-stage prompts affect attention and explore the nature of the factuality gap from a new perspective.

3 Preliminaries

3.1 Factual Knowledge

Definition of Factual Knowledge. We follow Ghosal et al. (2024); Petroni et al. (2019b) and decompose a verbalized piece of knowledge into three components: the subject entity, the relation type, and the answer. This structure aligns with the format used in many factual knowledge benchmarks. Thus, a piece of knowledge k can be simplified as a triplet: $k = (s, r, a)$, where $s \in \mathcal{S}$, $r \in \mathcal{R}$, and $a \in \mathcal{A}$, with $\mathcal{S}, \mathcal{R}, \mathcal{A}$ representing the sets of all subject entities, relations, and answers, respectively. For LLMs, if a model M has stored the knowledge k, then given s and r, the probability of the model outputting a should be close to 1: $P_M(a \mid s, r) \approx 1$.

Knowledge Extraction is Prompt-Sensitive. Factual knowledge is typically extracted in the form of QA queries. The knowledge $k = (s, r, a)$ corresponds to a set of question forms $Q(s, r) = \{f(s, r) \mid f \in \mathcal{F}\}$, where $\mathcal{F}$ denotes various prompt combinations. The question set Q is then input into the model to retrieve the answer a. However, experience shows that not all prompt inputs are able to retrieve a, as knowledge extraction is prompt-sensitive. Therefore, we hypothesize that if $k = (s, r, a)$ is stored in the model M, then

$$\forall 0 < \epsilon < 1, \exists q_1, q_2 \in Q(s, r), P_M(a \mid q_1) > \epsilon \\ \wedge P_M(a \mid q_2) < \epsilon. \quad (1)$$

3.2 Fine-tuning and Few-shot Learning

Given a set of knowledge triples $D_k = \{(s, r, a)_i^N\}$, we use a fine-tuning formatting function f_{ft} to construct the dataset $D_{\text{ft}} = \{f_{\text{ft}}(s, r), a \mid (s, r, a) \in D_k\}$. The model M_{ft} is trained on this dataset. During inference, the fine-tuning formatted query $q = f_{\text{ft}}(s, r)$ is used to extract knowledge. In the case of few-shot learning, the few-shot dataset is $D_{\text{few}} = \{(q, a)_i\}$, and in the case of few-shot CoT, the dataset is $D_{\text{cot}} = \{(f_{\text{cot}}(s, r), a)_i\}$.

Dai et al. (2023) has already mentioned the similarity between few-shot learning and fine-tuning, both of which leverage transformer attention for gradient descent. Let x be the input representation of a query token t, and W_Q, W_K, W_V are the projection matrices for computing the attention queries, keys, and values. We have $q = W_Q x$, and the form of the attention update after fine-tuning is given by:

$$\begin{aligned}\tilde{A}_{\text{FT}}(q) &= (W_V + \Delta W_V)XX^T(W_K + \Delta W_K)^T q \\ &= (W_V X(W_K X)^T q + \Delta W_V X(\Delta W_K X)^T q \\ &= (W_{\text{ZSL}} + \Delta W_{\text{FT}})q.\end{aligned} \tag{2}$$

Where X denotes the input representations of query tokens, ΔW_K and ΔW_V represent the parameter updates to W_K and W_V, respectively. $W_{\text{ZSL}}q$ denotes the attention in the zero-shot learning scenario.

Let X' denote the input representations of the demonstration tokens. The attention update form in few-shot learning is:

$$\begin{aligned}\tilde{A}_{\text{FSL}}(q) &= W_{\text{ZSL}}q + W_V X'(W_K X')^T q \\ &= W_{\text{ZSL}}q + \text{LinearAttn}(W_V X', W_K X', q) \\ &= W_{\text{ZSL}}q + \sum_i W_V x_i \left((W_K x_i)^T q\right) \\ &= W_{\text{ZSL}}q + \sum_i \left((W_V x_i) \otimes (W_K x_i)\right) q \\ &= (W_{\text{ZSL}} + \Delta W_{\text{FSL}})\, q.\end{aligned} \tag{3}$$

4 Impact of Unknown Knowledge on the Factuality of LLMs. (RQ1)

4.1 QA tasks

Settings. We fine-tune both the base and instruction-tuned versions of Llama3.1-8B [1] and Mistral-7B-v0.3[2] models on the known and unknown datasets constructed from Entity Questions (Sciavolino et al., 2021), PopQA (Mallen et al., 2023), and MMLU (Hendrycks et al., 2020), respectively. We use the exact match accuracy metric to evaluate performance. For the MMLU and Entity Questions datasets, we follow the approach of Gekhman et al. (2024), utilizing few-shot learning to split the dataset into known and unknown subsets. For PopQA, we adopt the methodology of Ghosal et al. (2024), where the dataset is partitioned into two parts based on the popularity of each data point. Both partitioning strategies ensure that the data distribution across question categories is balanced in both splits. Additionally, we randomly sample half of each question category from both splits to form a mixed dataset. More detailed settings can be found in Appendix A.1.

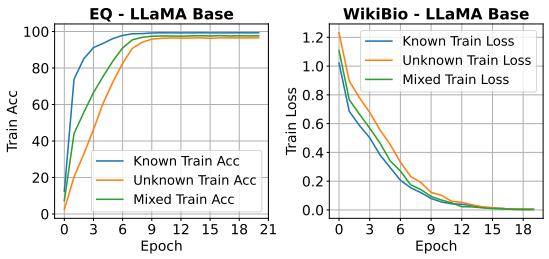

Figure 2: Training process of LLaMA Base on Entity Questions (EQ) and WikiBios.

Observation. The left subplot of Figure 2 shows the training process of LLaMA Base on the Entity Questions dataset, illustrating the general trend of the QA task training process. Table 1 presents the evaluation results on the evaluation dataset for models that reach early stopping and convergence. We observe that models trained on known knowledge converge more rapidly and exhibit fewer instances of factuality failure compared to those trained on unknown knowledge. Models trained on the mixed dataset converge at a rate between the two extremes, and their performance on the evaluation set is generally intermediate and closer to the known split.

The average factuality gap in early stopping settings and convergence settings are 6.49 and 8.19 respectively, which indicates that the gap becomes more pronounced when the model overfits to bad patterns. Furthermore, comparing the average performance gap between the Instruct and Base models reveals that, for both LLaMA (Base: 6.65, Instruct: 5.64) and Mistral (Base: 9.66, Instruct: 7.43), the gap for the Instruct models is smaller than that for the Base models. We hypothesize that this is due to the Instruct models having undergone

[1] https://huggingface.co/meta-llama/{Llama-3.1-8B, Llama-3.1-8B-Instruct}

[2] https://huggingface.co/mistralai/{Mistral-7B-v0.3, Mistral-7B-Instruct-v0.3}

Benchmark	Split	LLaMA		LLaMA-Instruct		Mistral		Mistral-Instruct	
		ES	Con.	ES	Con.	ES	Con.	ES	Con.
EQ	Unknow	28.25	24.80	28.75	25.00	21.15	18.00	26.00	20.90
	Mixed	40.70	35.85	39.65	34.70	35.90	31.15	36.25	31.20
	Known	40.30	38.50	39.20	37.70	36.05	34.45	35.40	34.50
PopQA	Unknown	32.45	29.55	31.80	28.90	28.15	23.80	27.30	21.85
	Mixed	35.65	34.70	34.75	33.70	32.10	29.45	32.20	27.75
	Known	36.05	34.35	35.15	32.60	32.85	29.90	32.60	30.00
MMLU	Unknown	34.94	33.90	33.64	33.51	28.09	26.52	31.61	25.87
	Mixed	36.32	35.08	34.94	34.03	33.51	31.29	33.57	30.89
	Known	37.49	37.10	35.92	34.88	35.60	34.81	33.44	32.14

Table 1: QA tasks evaluation accuracy. ES: Early Stop Con.: Convergence

some fine-tuning on QA tasks, allowing them to learn fundamental patterns that partially mitigate the gap. More details are in Appendix B.

Split	LLaMA		Mistral	
	ES	Con.	ES	Con.
Unknown	55.50	46.90	47.30	36.67
Mixed	59.49	48.32	50.59	38.62
Known	58.25	49.69	49.16	39.58

Table 2: FActScore of WikiBios task.

4.2 Open-ended generation task

Settings. We follow the approach outlined by Kang et al. (2024) using the WikiBios(Stranisci et al., 2023) dataset. To avoid instruction tuning disturbances, we use only the base versions of the two model families. The experimental setup is similar to that of the QA task, where the dataset is divided into three different splits. We use the FActScore (Min et al., 2023) metric to evaluate performance. For detailed implementation, please refer to Appendix A.2.

Observation. We observe a similar trend in the training curves and an increase in factuality failure when fine-tuning on unfamiliar data, as shown in the right subplot of Figure 2 and Table 2.

4.3 Toy Example

Settings. To further eliminate the potential impact of data filtering, we construct a Toy Example using manually created *Unknown* data that genuinely extends beyond the knowledge boundary of the LLM. We use the Llama3.3-70B-Instruct[3] model to extract data from the EntityQuestions dataset with a single query, without relying on few-shot examples. We then introduce fixed-format perturbations to entity tokens in the known set to create unknown knowledge set, ensuring that the model is unable to handle these perturbed examples. Additionally, we construct a mixed dataset combining known and unknown data in a 1:1 ratio. We fine-tune the models using LoRA, and evaluate their performance on the customized test set, which shares the same data type as the training set, i.e., normal (known) or perturbed (unknown). More experimental details can be found in Appendix A.3.

Observation. As shown in Table 6, we observe consistent gaps in factuality across models fine-tuned on known, mixed, and unknown knowledge sets. On the test set, the model fine-tuned on the known set achieves an accuracy of 91.9%, significantly outperforming the mixed set (66.5%) and unknown set (62%). This further confirms that unknown knowledge encourages factuality failure.

4.4 Our answer to RQ1

Fine-tuning on unknown knowledge encourages factuality failure, which is, in fact, a failure of in-domain generalization. The extraction pattern for unknown knowledge is a poor pattern, leading to poor generalization. Furthermore, the strength of this pattern is influenced by the training data and can be adjusted by the proportion of unknown data.

5 Does this Kind of Factuality Gap Always Exist? (RQ2)

5.1 Experimental Results

Settings. To better understand the impact of unknown data on model factuality, we categorize fac-

[3] https://huggingface.co/meta-llama/Llama-3.3-70B-Instruct

Dataset		Split	LLaMA		LLaMA-Instruct		Mistral		Mistral-Instruct	
			ES	Con.	ES	Con.	ES	Con.	ES	Con.
ID	eq_id	Unknow	28.25	24.80	28.75	25.00	21.15	18.00	26.00	20.90
		Known	40.30	38.50	39.20	37.70	36.05	34.45	35.40	34.50
NID	eq_ood	Unknown	30.00	28.93	31.67	30.43	32.17	23.73	30.43	24.13
		Known	39.03	36.60	38.17	37.03	34.83	33.00	34.17	32.43
	pop_ood	Unknown	28.17	23.79	19.00	19.42	23.13	20.19	25.89	22.74
		Known	32.58	32.05	27.54	25.47	28.69	27.40	29.71	28.06
OW	mmlu_ood	Unknown	66.11	66.70	69.23	69.30	62.63	62.46	62.25	62.53
		Known	67.05	67.09	69.51	69.47	62.98	63.54	60.74	60.70

Table 3: Generalization factuality. ID: in-distribution, NID: near in-distribution, OW: open world

tuality generalization into two types based on the distance between the test and training task data patterns: (1) *near in-distribution generalization* and (2) *open-world model factuality*. In the following, we examine the effects of unknown data on each type of factuality. We employ all-MiniLM-L6-v2[4] embedding model (Reimers and Gurevych, 2019) to extract and process data patterns from both out-of-distribution (OOD) and in-distribution (ID) test sets. By comparing the cosine similarity between these patterns, we are able to measure the distance between OOD and ID data.

We conduct validation experiments using models fine-tuned on the Entity Questions dataset from Section 4. For near in-distribution tasks, we sample non-overlapping data from the Entity Questions and PopQA datasets to create near in-distribution test sets, `eq_ood` and `pop_ood`. For the open-world task, we choose MMLU to create a complete `mmlu_ood` set, which provides more diverse data and significantly different question formats. The cosine similarities between `eq_ood`, `pop_ood`, `mmlu_ood` and the ID test set are 0.86, 0.82 and 0.55 respectively. More details can be found in Appendix A.4.

Observation. As shown in Table 3, Llama3.1-8B fine-tuned on known data consistently outperforms the model fine-tuned on unknown data for both `eq_ood` and `pop_ood` datasets. The performance gap is 9% on `eq_ood` and 4% on `pop_ood` with early stopping, and 7.5% and 8% at convergence, respectively. The factuality gap on `mmlu_ood` nearly disappears across all models. For instance, the Llama3.1-8B model fine-tuned on unknown data achieves a QA accuracy of 66.11% with early stopping, just 1% lower than the 67.05% achieved by the model trained on known data. At convergence, the performance gap narrows further to 0.4%, with 67.09% of the known data model and 66.7% of the unknown data model. These findings are consistent across other models.

[4] https://huggingface.co/sentence-transformers/all-MiniLM-L6-v2

Split	NID		OW
	eq_ood	pop_ood	mmlu_ood
Unknown	55.30	49.23	82.14
Mixed	54.87	49.34	81.72
Known	55.13	48.92	82.21

Table 4: Performance of Toy Example on OOD tasks at convergence.

However, we observe that this negative impact vanishes in our 70B Toy Examples, which are fine-tuned with limited data. As shown in Table 4, models trained on both known and unknown data perform similarly on `eq_ood`, `pop_ood` and `mmlu_ood`, with no significant differences. We hypothesize that this is due to the limited training data and the high capacity of the model, which prevents unknown data from substantially influencing the model's knowledge extraction.

5.2 Our answer to RQ2

This kind of factuality gap does not always exist. The negative impact of unknown knowledge on generalization decreases as the OOD data pattern becomes more distinct from the ID data.

6 Can the Factuality Gap be Easily Mitigated? (RQ3)

6.1 Proper prompt at inference stage may mitigate the gap

Settings. We select all the models and tasks from Section 4. For the QA tasks, we perform inference using few-shot or few-shot CoT approaches. The

Dataset		LLaMA		LLaMA-Instruct		Mistral		Mistral-Instruct	
		ES	Con.	ES	Con.	ES	Con.	ES	Con.
EQ	U	41.55$_{+13.3}$	38.95$_{+14.2}$	41.00$_{+12.3}$	37.40$_{+12.4}$	35.35$_{+14.2}$	32.95$_{+15.0}$	35.25$_{+9.25}$	30.05$_{+9.15}$
	K	43.45$_{+3.15}$	42.20$_{+3.70}$	41.20$_{+2.00}$	40.70$_{+3.00}$	38.25$_{+2.20}$	37.95$_{+3.50}$	33.15$_{-2.25}$	32.65$_{-1.85}$
PQ	U	39.45$_{+7.00}$	39.35$_{+9.80}$	35.55$_{+3.75}$	35.30$_{+6.40}$	33.90$_{+5.75}$	33.4$_{+9.60}$	32.80$_{+5.50}$	32.25$_{+10.4}$
	K	39.50$_{+3.45}$	39.15$_{+4.80}$	34.35$_{-0.80}$	35.80$_{+3.20}$	35.20$_{+2.35}$	34.10$_{+4.20}$	34.20$_{+1.60}$	32.50$_{+2.50}$
MU	U	54.80$_{+19.9}$	54.60$_{+20.7}$	64.99$_{+31.4}$	65.32$_{+31.8}$	55.39$_{+27.3}$	55.13$_{+28.6}$	58.00$_{+26.4}$	60.09$_{+34.2}$
	K	67.60$_{+30.1}$	67.86$_{+30.8}$	69.30$_{+33.4}$	68.84$_{+34.0}$	58.46$_{+22.9}$	58.39$_{+23.6}$	61.07$_{+27.6}$	60.94$_{+28.8}$
WB	U	55.20$_{-0.30}$	48.32$_{+1.42}$			47.93$_{+0.63}$	37.99$_{+1.32}$		
	K	58.20$_{-0.05}$	50.85$_{+1.16}$			50.58$_{+1.42}$	40.22$_{+0.64}$		

Table 5: Performance of the fine-tuned model with few-shot and few-shot CoT. EQ: Entity Questions, PQ: PopQA, MU: MMLU, WB: WikiBios. Exact Match Accuracy for QA tasks and FactScore for WikiBios, with underlined results for few-shot and non-underlined for few-shot CoT. The small number in the bottom right corner represents the improvement or decline in current performance relative to the performance without using few-shot learning.

Split	Original		With CoT	
	ES	Con.	ES	Con.
Unknown	44.73	41.70	84.08	82.81
Mixed	63.67	59.96	87.21	87.21
Known	83.11	82.81	86.72	87.60

Table 6: Performance of Toy Example.

few-shot examples are selected from the *Known* training data, after which GPT-4[5] generates an analysis of the question entity to construct the CoT for the given query. These examples are incorporated into the few-shot CoT format for inference. The box below is the few-shot CoT example format.

Question:{} Analysis:{} Answer:{}

We selected 3 sets of examples in total and considered two few-shot scenarios: one with CoT and one without. We ensure that the prompts input into the *Known* and *Unknown* models under the same conditions are exactly the same. The set with the best performance on the *Unknown* split was then chosen as the final outcome. For the generation task, we use only the few-shot learning approach, selecting examples in the same manner as in the previous case. Additionally, we also add special CoT to the Toy Example for verification. Detailed prompt design and Toy Example CoT are presented in Appendix C and Appendix A.3.

Observation. Table 5 presents a comparison of the results obtained through few-shot or few-shot CoT inference after training different models on various datasets. We can observe that, in most cases, after using few-shot learning, the performance on the *Unknown* split improves more significantly compared to the *Known* split. This suggests that the factuality gap can be mitigated or even fully eliminated. Additionally, we observe the following points: 1) The gap in models with early stopping is more easily mitigated. 2) The factuality gap of the *Instruct* model is easier to mitigate than *Base*, especially in the case of Convergence. 3) In MMLU and WikiBios, using few-shot learning sometimes even increases the performance gap. This may be due to the particularities of these two tasks compared to regular QA tasks. The former is a comprehensive dataset with complex and varied question formats, while the latter is an open-ended generation task, both of which result in a more complex factuality gap pattern.

Results of Toy Example are shown in Table 6. We observe that CoT effectively enhances model testing performance and narrows the factuality gap between the three 70B models.

6.2 Ablation study

To better understand the essence of how few-shot learning mitigates the factuality gap, we design the following ablation experiment on LLaMA Base model and Entity Questions dataset. Details of abalation studies can be found in Appendix D.

Prompt components We conduct an ablation study on the composition of the prompt, separately examining the selection of examples in few-shot prompts and the impact of CoT. We validated the effectiveness of *Known* examples and CoT, as shown in Figure 3.

[5] https://openai.com/index/gpt-4o-system-card/

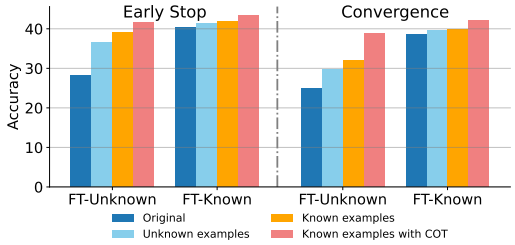

Figure 3: Ablation study of few-shot examples and CoT.

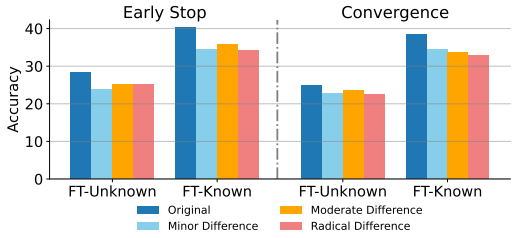

Figure 4: Ablation study of prompt formulation.

Prompt formulation We also study the impact of changing the prompt format on the factuality gap. We use GPT-4o to rephrase these questions in three different formats and find that the performance decline in all cases, and the factuality gap remains large.

6.3 Our answer to RQ3

For parameterized knowledge, both few-shot learning and supervised fine-tuning (SFT) are methods for extracting knowledge. SFT does not lead to the forgetting of prior knowledge, rather, it results in a suboptimal method for extracting knowledge. Moreover, in the case of unknown knowledge, the poor extraction patterns induced by SFT are fragile and can be adjusted back to a better extraction method using an appropriate prompt, such as the few-shot CoT approach.

7 Exploring New Insights into Knowledge Extraction in Large Language Models

7.1 Hypothesis

Based on the experimental observations and analysis above, we propose two hypotheses:

- **Hypothesis 1.** SFT does not cause forgetting by disrupting the knowledge storage of the LLM, but rather affects the model's factuality through attention patterns.
- **Hypothesis 2.** The attention patterns formed during the fine-tuning phase can be readjusted during the inference phase.

7.2 Explaination

We selected the LLaMA Base model trained on entity questions and the untrained LLaMA Base model for case studies. We follow the attention visualization method proposed by Ghosal et al. (2024), where the previous token of the generated answer is used as the query to attend to other tokens, allowing us to construct the attention map for each layer. More examples can be found in Appendix E.

Factuality gap. First, we observe the test results of *Unknown* knowledge and *Known* knowledge on the pre-trained model in Figure 5. It can be seen that the attention on the subject entity of *Known* knowledge is more prominent.

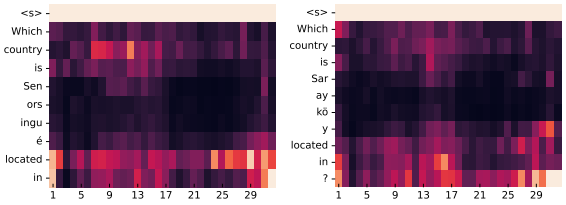

Figure 5: Attention maps of base model. Left: *Unknown* data, subject entity is "Senorsingué". Right: *Known* data, subject entity is "Sarayköy"

Based on **Hypothesis 1**, the attention pattern determines the model's factuality. Fine-tuning essentially reinforces the attention pattern of the base model, as shown in Figure 6. This hypothesis helps explain many phenomena observed in **RQ1** and **RQ2**.

Referring to Equation 4, the instruct model has already undergone a period of fine-tuning, so further fine-tuning on QA tasks can be viewed as an update to its attention:

$$\tilde{A}_{\text{FT}}(q) = (W_{\text{ZSL}} + \Delta W_{\text{instruct}} + \Delta W_{D_{\text{ft}}})q. \quad (4)$$

where $\Delta W_{\text{instruct}}$ represents the updates generated by instruction tuning, while $\Delta W_{D_{\text{ft}}}$ refers to the updates generated by SFT on the fine-tuning dataset D_{ft}. Due to the difference in dataset size, the former is generally larger than the latter. Therefore, the factuality gap of the instruct model is less affected. Similarly, the gap in the convergence model is larger than that in the early stop model for the same reason.

In **RQ2**, self-attention fundamentally performs semantic relevance computations (Vaswani et al., 2017) and does not disrupt the utilization of other

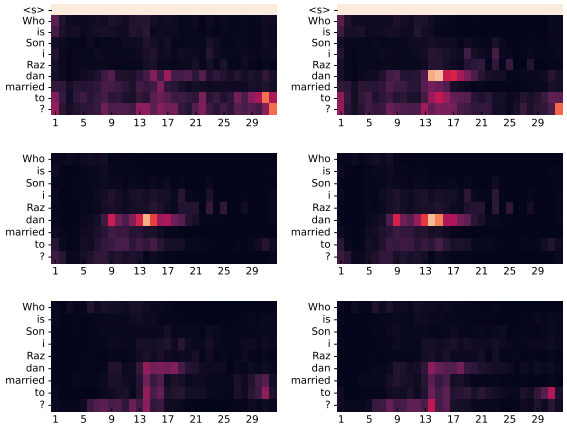

Figure 6: Attention maps of fine-tuned models. Top: Origin prompt. Middle: With few-shot learning. Bottom: With CoT. Left: Fine-tuned on *Unknown* data. Right: Fine-tuned on *Known* data. Subject entity is "Soni Razdan".

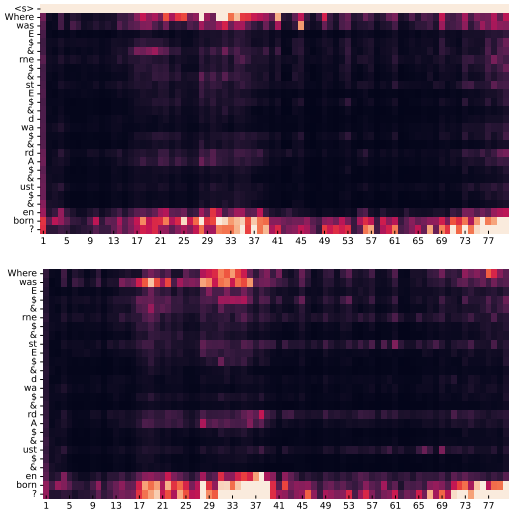

Figure 7: Attention maps of Toy Example. Top: Origin prompt. Bottom: With CoT. The subject entity without garbled text is "Ernest Edward Austen".

knowledge. The lower the semantic relevance with the training dataset, the less prominent this attention pattern becomes.

Based on the above, we can formalize the factuality gap referring to the definition in Section 3. Specifically, for a knowledge triple $k = (s, r, a)$, the LLM is trained on Unknown and Known data to obtain M_U and M_K, respectively. We can get:

$$\begin{aligned} \Delta_{\text{factuality}} &= |P_{M_U}(a \mid q) - P_{M_K}(a \mid q)| \\ &\propto |\text{Attn}_{M_U}(Q, K_s, V) - \text{Attn}_{M_K}(Q, K_s, V)| \end{aligned} \tag{5}$$

Attention reallocation In **RQ3**, we confirm that knowledge extraction is prompt-sensitive. By using carefully designed prompts, such as few-shot in-context learning or incorporating CoT, we can mitigate the pattern differences caused by the training data, as shown in Figure 6. For the Toy Example, using a specialized CoT can also correct the attention of the model trained on *Unknown* data, as shown in Figure 7.

For few-shot learning, its principle has been explained in 3 and can be described as:

$$\tilde{A}_{\text{FT}}(q) = (W_{\text{ZSL}} + \Delta W_{D_{\text{ft}}} + \Delta W_{D_{\text{few}}})q. \tag{6}$$

When $\Delta W_{D_{\text{few}}}$ can eliminate the effect of $\Delta W_{D_{\text{ft}}}$, the factuality gap can be mitigated.

For CoT, on one hand, the attention becomes sparser after adding CoT, where, in the demonstration, the answer tends to have stronger associations with tokens that are more strongly related, specifically the tokens corresponding to the subject entity. On the other hand, the analysis also shows an increased frequency of occurrence of the subject entity token. This can be explained from both aspects.

8 Conclusion

In this paper, we conduct an in-depth exploration of the factuality gap caused by fine-tuning. We study the factors that influence the emergence of the factuality gap, the generalization of the factuality gap, and methods for mitigating the factuality gap. Based on the analysis of these experimental phenomena, we find that the essence of the factuality gap is an attention pattern. This pattern, formed during the fine-tuning phase, can be modified through in-context learning, thereby influencing the factuality gap. In summary, this paper offers a new understanding of LLM factuality and provides novel insights into model reliability and the application of models in downstream tasks related to factual knowledge.

9 Limitations

Our work is primarily empirical in nature, with relatively underdeveloped theoretical proof aspects. Future work can delve deeper into the factuality gap from a theoretical perspective. Additionally, the explanation of the anomalous phenomena observed in the MMLU and WikiBios datasets in this work is somewhat vague, and there is a lack of further analysis regarding the differences in factuality between these datasets and typical QA tasks.

References

Zeyuan Allen-Zhu and Yuanzhi Li. 2024. Physics of language models: Part 3.1, knowledge storage and extraction. *Preprint*, arXiv:2309.14316.

Tom B. Brown, Benjamin Mann, Nick Ryder, Melanie Subbiah, Jared Kaplan, Prafulla Dhariwal, Arvind Neelakantan, Pranav Shyam, Girish Sastry, Amanda Askell, Sandhini Agarwal, Ariel Herbert-Voss, Gretchen Krueger, Tom Henighan, Rewon Child, Aditya Ramesh, Daniel M. Ziegler, Jeffrey Wu, Clemens Winter, Christopher Hesse, Mark Chen, Eric Sigler, Mateusz Litwin, Scott Gray, Benjamin Chess, Jack Clark, Christopher Berner, Sam McCandlish, Alec Radford, Ilya Sutskever, and Dario Amodei. 2020. Language models are few-shot learners. *Preprint*, arXiv:2005.14165.

Banghao Chen, Zhaofeng Zhang, Nicolas Langrené, and Shengxin Zhu. 2024. Unleashing the potential of prompt engineering in large language models: a comprehensive review. *Preprint*, arXiv:2310.14735.

Damai Dai, Yutao Sun, Li Dong, Yaru Hao, Shuming Ma, Zhifang Sui, and Furu Wei. 2023. Why can GPT learn in-context? language models implicitly perform gradient descent as meta-optimizers. In *ICLR 2023 Workshop on Mathematical and Empirical Understanding of Foundation Models*.

Abhimanyu Dubey, Abhinav Jauhri, Abhinav Pandey, Abhishek Kadian, Ahmad Al-Dahle, Aiesha Letman, Akhil Mathur, Alan Schelten, Amy Yang, Angela Fan, et al. 2024. The llama 3 herd of models. *arXiv preprint arXiv:2407.21783*.

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? In *Proceedings of the 2024 Conference on Empirical Methods in Natural Language Processing*, pages 7765–7784, Miami, Florida, USA. Association for Computational Linguistics.

Gaurav Rohit Ghosal, Tatsunori Hashimoto, and Aditi Raghunathan. 2024. Understanding finetuning for factual knowledge extraction. In *Forty-first International Conference on Machine Learning*.

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*.

Lei Huang, Weijiang Yu, Weitao Ma, Weihong Zhong, Zhangyin Feng, Haotian Wang, Qianglong Chen, Weihua Peng, Xiaocheng Feng, Bing Qin, and Ting Liu. 2025. A survey on hallucination in large language models: Principles, taxonomy, challenges, and open questions. *ACM Trans. Inf. Syst.*, 43(2).

Albert Q. Jiang, Alexandre Sablayrolles, Arthur Mensch, Chris Bamford, Devendra Singh Chaplot, Diego de las Casas, Florian Bressand, Gianna Lengyel, Guillaume Lample, Lucile Saulnier, Lélio Renard Lavaud, Marie-Anne Lachaux, Pierre Stock, Teven Le Scao, Thibaut Lavril, Thomas Wang, Timothée Lacroix, and William El Sayed. 2023. Mistral 7b. *Preprint*, arXiv:2310.06825.

Nitish Joshi, Javier Rando, Abulhair Saparov, Najoung Kim, and He He. 2024. Personas as a way to model truthfulness in language models. In *Proceedings of the 2024 Conference on Empirical Methods in Natural Language Processing*, pages 6346–6359, Miami, Florida, USA. Association for Computational Linguistics.

Nikhil Kandpal, Haikang Deng, Adam Roberts, Eric Wallace, and Colin Raffel. 2023. Large language models struggle to learn long-tail knowledge. In *Proceedings of the 40th International Conference on Machine Learning*, volume 202 of *Proceedings of Machine Learning Research*, pages 15696–15707. PMLR.

Katie Kang, Eric Wallace, Claire Tomlin, Aviral Kumar, and Sergey Levine. 2024. Unfamiliar finetuning examples control how language models hallucinate. In *Automated Reinforcement Learning: Exploring Meta-Learning, AutoML, and LLMs*.

Mehran Kazemi, Sid Mittal, and Deepak Ramachandran. 2023. Understanding finetuning for factual knowledge extraction from language models. *Preprint*, arXiv:2301.11293.

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.

Kevin Meng, David Bau, Alex J Andonian, and Yonatan Belinkov. 2022. Locating and editing factual associations in GPT. In *Advances in Neural Information Processing Systems*.

Sewon Min, Kalpesh Krishna, Xinxi Lyu, Mike Lewis, Wen-tau Yih, Pang Koh, Mohit Iyyer, Luke Zettlemoyer, and Hannaneh Hajishirzi. 2023. FActScore: Fine-grained atomic evaluation of factual precision in long form text generation. In *Proceedings of the 2023 Conference on Empirical Methods in Natural Language Processing*, pages 12076–12100, Singapore. Association for Computational Linguistics.

Fabio Petroni, Tim Rocktäschel, Sebastian Riedel, Patrick Lewis, Anton Bakhtin, Yuxiang Wu, and Alexander Miller. 2019a. Language models as knowledge bases? In *Proceedings of the 2019 Conference on Empirical Methods in Natural Language Processing and the 9th International Joint Conference on Natural Language Processing (EMNLP-IJCNLP)*, pages 2463–2473, Hong Kong, China. Association for Computational Linguistics.

Fabio Petroni, Tim Rocktäschel, Patrick Lewis, Anton Bakhtin, Yuxiang Wu, Alexander H. Miller, and Sebastian Riedel. 2019b. Language models as knowledge bases? *Preprint*, arXiv:1909.01066.

Nils Reimers and Iryna Gurevych. 2019. Sentence-bert: Sentence embeddings using siamese bert-networks. In *Proceedings of the 2019 Conference on Empirical Methods in Natural Language Processing*. Association for Computational Linguistics.

Christopher Sciavolino, Zexuan Zhong, Jinhyuk Lee, and Danqi Chen. 2021. Simple entity-centric questions challenge dense retrievers. *arXiv preprint arXiv:2109.08535*.

Marco Antonio Stranisci, Rossana Damiano, Enrico Mensa, Viviana Patti, Daniele Radicioni, and Tommaso Caselli. 2023. WikiBio: a semantic resource for the intersectional analysis of biographical events. In *Proceedings of the 61st Annual Meeting of the Association for Computational Linguistics (Volume 1: Long Papers)*, pages 12370–12384, Toronto, Canada. Association for Computational Linguistics.

Ashish Vaswani, Noam Shazeer, Niki Parmar, Jakob Uszkoreit, Llion Jones, Aidan N Gomez, Ł ukasz Kaiser, and Illia Polosukhin. 2017. Attention is all you need. In *Advances in Neural Information Processing Systems*, volume 30. Curran Associates, Inc.

Zhihu Wang, Shiwan Zhao, Yu Wang, Heyuan Huang, Sitao Xie, Yubo Zhang, Jiaxin Shi, Zhixing Wang, Hongyan Li, and Junchi Yan. 2024. Re-task: Revisiting llm tasks from capability, skill, and knowledge perspectives. *Preprint*, arXiv:2408.06904.

Jason Wei, Maarten Paul Bosma, Vincent Zhao, Kelvin Guu, Adams Wei Yu, Brian Lester, Nan Du, Andrew Mingbo Dai, and Quoc V. Le. 2022a. Finetuned language models are zero-shot learners.

Jason Wei, Xuezhi Wang, Dale Schuurmans, Maarten Bosma, brian ichter, Fei Xia, Ed Chi, Quoc V Le, and Denny Zhou. 2022b. Chain-of-thought prompting elicits reasoning in large language models. In *Advances in Neural Information Processing Systems*, volume 35, pages 24824–24837. Curran Associates, Inc.

Xunjian Yin, Xu Zhang, Jie Ruan, and Xiaojun Wan. 2024. Benchmarking knowledge boundary for large language models: A different perspective on model evaluation. In *Proceedings of the 62nd Annual Meeting of the Association for Computational Linguistics (Volume 1: Long Papers)*, pages 2270–2286, Bangkok, Thailand. Association for Computational Linguistics.

Tong Zhang, Peixin Qin, Yang Deng, Chen Huang, Wenqiang Lei, Junhong Liu, Dingnan Jin, Hongru Liang, and Tat-Seng Chua. 2024. CLAMBER: A benchmark of identifying and clarifying ambiguous information needs in large language models. In *Proceedings of the 62nd Annual Meeting of the Association for Computational Linguistics (Volume 1: Long Papers)*, pages 10746–10766, Bangkok, Thailand. Association for Computational Linguistics.

A Experiment Details

A.1 QA tasks

Data processing. For the Entity Questions task, we adopt the experimental framework outlined by Gekhman et al. (2024). Specifically, we select train split and dev split data from the following relation subsets: `P131`, `P136`, `P17`, `P19`, `P26`, `P264`, `P36`, `P40`, `P495`, `P69`, `P740`, and `P800` for both training and evaluation purposes. The remaining relation subsets are reserved for out-of-distribution (OOD) testing, as described in Section 5. We employ a few-shot learning approach to classify the Unknown and Known datasets. Within the dev split, we randomly select 10 sets, each containing 4 examples, and apply both greedy and random sampling decoding methods. For random sampling, the following parameters are used: `temperature`=0.5, `top_p`=1.0, `top_k`=40, and 16 answers are sampled. The data is classified as either Unknown or Known based on the accuracy of the greedy search and random sample. If at least one correct answer is obtained from either the greedy search or random sampling, the data is classified as Known.

We perform this filtering procedure for each relation subset and subsequently use the filtered Unknown and Known splits to balance the data across categories. After filtering, the number of Unknown and Known samples for each of the four models is as follows: LLaMA Base: 28,337, LLaMA Instruct: 31,226, Mistral Base: 30,952, and Mistral Instruct: 31,335. For evaluation, we randomly select 2,000 samples from the development dataset corresponding to the relation subsets used in the training dataset.

For PopQA, we follow the approach of Ghosal et al. (2024) and divide the dataset into two parts based on the popularity value of the subject entity in each data point, denoted as "s_pop". Similar to Entity Questions, we perform the splitting for each question type individually. First, each subclass dataset is randomly divided into a training set and an evaluation set in a 4:1 ratio. Then, the training set is further split into two halves to ensure equal distribution of each type of question. Finally, the Unknown and Known datasets contain 5,704 samples, while the evaluation dataset consists of 2,858 samples.

For MMLU, we also adopt a few-shot learning approach, but with some simplifications. We directly select 5 data points from the MMLU dev split as a group of few-shot examples. Apart from changing the number of random samples to 4, the other model hyperparameters are set the same as in Entity Questions. We use the test split of MMLU as the training data and the val split as the evaluation data. For the training data, we ensure that the Unknown and Known datasets have the same number of samples by taking the smaller size from each class. Finally, the number of Unknown and Known samples for the four models is as follows: LLaMA Base: 2,724, LLaMA Instruct: 2,730, Mistral Base: 2,994, Mistral Instruct: 4,128. The length of the evaluation dataset is 1,531.

The mixed training datasets for the three models are constructed by randomly selecting half of the data from both the Unknown and Known subsets of each class to form new datasets.

Training Details. We divide all the training into 12 groups based on the dataset and model, with each group containing training on the Unknown, Known, and Mixed subsets. We ensure that the training parameters are exactly the same within each group.

For all the three datasets, the training hyperparameters are set as follows: the batch size is 128, and we use a fixed learning rate. Specifically, the learning rates for LLaMA Base and LLaMA Instruct are set to 1e-5, while for Mistral Base and Mistral Instruct, the learning rate for Entity Questions is 5e-6, and for the other datasets, it is set to 1e-6. No additional regularization methods are used during training. The training for all three datasets used the model with the best accuracy on the evaluation set as the early stop model, and the model whose loss converged after completing all epochs is considered the Convergence model.

For the Entity Questions dataset, all models are trained for 20 epochs. For PopQA, the LLaMA models are trained for 15 epochs, and the Mistral Base and Mistral Instruct models are trained for 30 and 35 epochs, respectively. For MMLU, the LLaMA models are trained for 15 epochs, and the Mistral models are trained for 30 epochs.

Additionally, for the SFT process prompt, the PopQA dataset use the original questions and answers, while the question prompt format for the Entity Questions dataset is as follows:

> Answer the following question.\n Who is Caitlin Thomas married to?

The question prompt format for the MMLU

dataset is as follows:

> The following is a multiple choice question, paired with choices. Answer the question in format: 'Choice:content'.\n\n### Question:\nThe cyclic subgroup of Z_24 generated by 18 has order\n\n### Choices:\nA) 0 B) 4 C) 2 D) 6 \n\n### Answer:\n

Evaluation Details. We use Exact Match as the metric to measure the model's evaluation accuracy. During testing, the prompt format of the questions is the same as during training. The model during testing uses the greedy search decoding method with a `max_token` value of 10.

A.2 Open-ended generation tasks

Data processing. We utilize the WikiBios (Kang et al., 2024) data directly, randomly selecting 2,000 entries as the training set and 500 entries as the evaluation dataset. For the training set partition, we also employ a few-shot learning approach. In the evaluation set, we select 4 examples and used the random sample decoding method to sample two answers, with `max_token=32`. The remaining decoding parameters are the same as in Entity Questions. To assess the accuracy of the answers, we employed the **FActScore** metric. The GPT model used for this task is `gpt-3.5-turbo-0125`, with raw scores and no penalties applied for the `num_fact` parameter. Each data point is evaluated individually, and the average of the two sampled answers is taken. Based on the resulting FActScore, the training set is then divided into two parts: the higher-scoring subset is classified as Known, while the lower-scoring subset is classified as Unknown.

Training Details. The dataset is trained only on LLaMA Base and Mistral Base, with a batch size of 128 and a fixed learning rate of 1e-5. No additional regularization methods are used. Training stops when the loss converged to below 0.01, and this model is considered the Convergence Model. The model with the lowest evaluation loss is selected as the early stop model.

Evaluation Details. We used **FActScore** as the evaluation metric, with the same data processing settings as described above.

A.3 Toy Example

For our Toy Example, we utilized the Llama3.3-70B-Instruct[6] model, incorporating data sampled from the EntityQuestions dataset.

Data processing. We employ the Llama3.3-70B model to construct the *Known* knowledge set by querying the model with the original questions. To each question, we append the phrase *"Answer the following question."* before the question itself to form a complete query, without relying on additional few-shot examples. Specifically, we apply a greedy sampling method, limiting the model's output to a maximum of 10 tokens, and verified whether the ground truth answer is present in the model's response. If the ground truth answer is included, we identifiy the subject words in the question. For each subject word longer than two letters, we introduce a fixed perturbation, "$&". For subject words of three letters, the perturbation is inserted after the first letter. For subject words longer than three letters, the perturbation is applied before the second letter. The modified question is then re-entered into the model to ensure that the resulting response did not contain the answer to the original question, and regarded as the *Unknown knowledge*.

Below is an example of our known and unknown set consturction, using the real question from relation `P26`. The question in this case is "Who is Caitlin Thomas married to?", and the ground truth answer is "Dylan Thomas". The subject words in the question is "Caitlin Thomas".

> Q: Answer the following question.\n Who is Caitlin Thomas married to?
> A: Caitlin Thomas.
> Modified: Answer the following question.\n Who is C$&aitl$&in T$&hom$&as married to?
> A: Rio de Janeiro.

We combine the following relations from the EntityQuestion dataset: `P131`, `P136`, `P17`, `P19`, `P26`, `P264`, `P36`, `P40`, `P495`, `P69`, `P740`, and `P800`, resulting in a training set of 2,000 data entries and a test set of 1,000 for the *Known*, *Unknown* and *Mixed* dataset. We ensure that the ratio of known to unknown data in the *Mixed* dataset is 1:1, with the Unknown data derived from the Known data. Notably, the data in the *Mixed* dataset does not overlap

[6] https://huggingface.co/meta-llama/Llama-3.3-70B-Instruct

with the *Known* or *Unknown* datasets.

Training Details. During the training of the Toy Example, we use a learning rate of 2e-5, a batch size of 128, and a weight decay of 0. We apply a cosine learning rate scheduler with a warm-up of 64 steps. We use the training data template detailed in Appendix A.1, and trained the model for a total of 50 epochs on an 8×6000 Ada 48G setup.

Toy Example CoT prompt. To mitigate the performance gap caused by fine-tuning on different data filters, we employ the following Chain-of-Thought (CoT) prompt to guide the model in reasoning and answering the questions.

> Ignore all the special characters in the following question. Think step by step. First, clean all special characters in the question. In this step, you might see some unicode characters in foreign languages. Next, rethink the cleaned question. Finally, give the detailed answer of the cleaned question with short explanation.

A.4 Generalization

For near in-distribution tasks, We follow Gekhman et al. (2024) and sample non-overlapping data from the remaining relation subsets of the Entity Questions with 3000 data points to create near in-distribution test set `eq_ood`.We use the entire PopQA evaluation dataset as near in-distribution test sets `pop_ood`. The cosine similarities between `eq_ood`, `pop_ood`, and the ID test set are 0.86 and 0.82, respectively. For the open-world task, we choose MMLU, which provides more diverse data and significantly different question formats. We select 50 samples from each of the 57 MMLU tasks to create a complete `mmlu_ood` set. After embedding, the cosine similarity between `mmlu_ood` and the ID test set is 0.55.

B Train Acc Curves

The training accuracy curve for all QA tasks is shown in Figure 9, while the training loss curve for the generation task is shown in Figure 8.

C Prompt Design Details

For few-shot learning, we select examples from the Known split. Considering the length and effectiveness of the examples, 4 examples were selected from PopQA and Entity Questions, while 3 examples were selected from MMLU. We used GPT-4 to

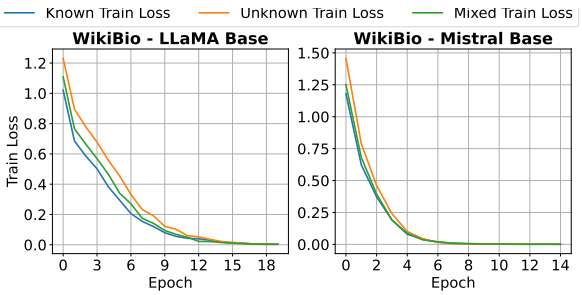

Figure 8: Training loss of generation task

generate the CoT prompts for each type of task. For each dataset, we input the few-shot learning examples and generate the CoT instructions according to the question type, thus obtaining the corresponding few-shot CoT prompt for each question type. The instructions for each dataset are as follows:

> **Entity Questions, PopQA:** Follow the few shot Chain of Thought example format: Question:{} Analysis:{} Answer:{} to modify the format and generate analysis of the entity in each question of the QA pairs below. The analysis should describe the related information of the entity shortly in the question in order to lead to the answer:

> **MMLU:** 'Follow the few-shot Chain of Thought example format: Question:{} Choices:{} Analysis:{} Answer:{} to modify the format and generate analysis of the critical entity in each multiple choice question below. The analysis should describe the related information of the entity in the question shortly in order to lead to the answer:\n

D Abalation Study Details

For the selection of few-shot learning examples, Table 7 shows the test results for all *Unknown* examples. The testing of *Unknown* examples is the same as for *Known* examples, where 3 sets are randomly selected from the corresponding dataset, with each set containing 4 examples. The set with the best performance is then chosen. As for the results using only Known examples in Table 8, it can be observed that for most models, the factuality improves when using Known examples.

For the ablation experiment of CoT, the results using only few-shot learning and those with the addition of CoT are shown in Table 8 and Table

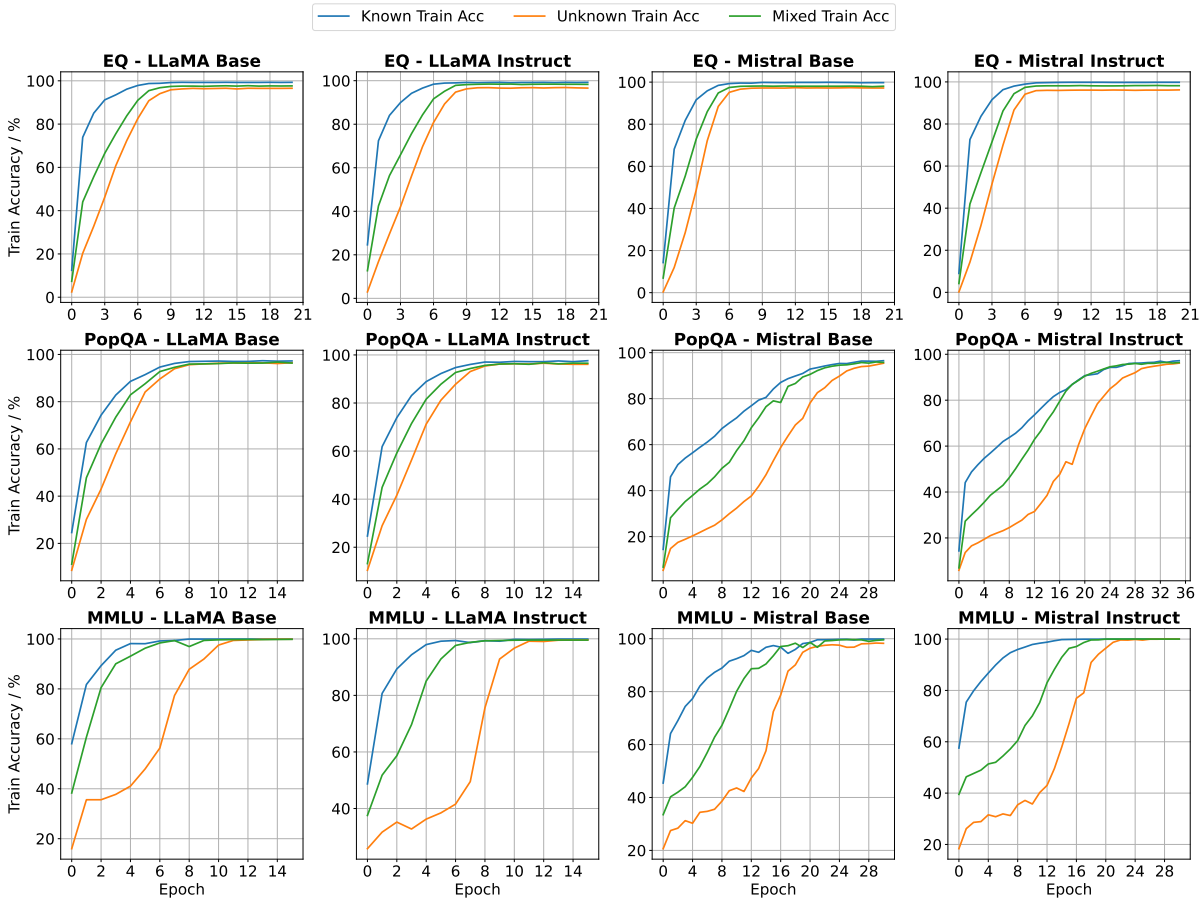

Figure 9: Training accuracy of QA tasks

9, respectively. By comparing the results, we can observe the differences between the models with and without CoT. We find that the factuality of the models trained on PopQA and Entity Questions improves, while the results on MMLU are more unstable and sometimes do not show any improvement with the addition of CoT. We hypothesize that this may be due to CoT causing the text to become too long, leading to a performance degradation.

For the ablation experiment on the variation of question formats, we used GPT-4 to rephrase 2,000 data points from the Entity Questions evaluation dataset three times. The instructions for the three rephrasings are as follows:

> Please rephrase this question with Minor Difference. Just return the rephrased question without additional word.
> Please rephrase this question with Moderate Difference. Just return the rephrased question without additional word.
> Please rephrase this question with Radical Difference. Just return the rephrased question without additional word.

E Attention Visualization

Two additional questions are added to visualize attention in three different cases. The two questions are: "Which country is Valea Coacăzei River located in?" and "Where was Margaret Mwanakatwe born?". The attention maps are shown in Figures 10 and 11, respectively.

Benchmark	Split	LLaMA		LLaMA-Instruct		Mistral		Mistral-Instruct	
		ES	Con.	ES	Con.	ES	Con.	ES	Con.
EQ	Unknow	36.60	29.60	33.25	26.20	26.55	18.50	30.95	19.20
	Known	41.45	39.55	39.45	37.55	33.80	33.75	32.55	32.80
PopQA	Unknown	30.95	29.55	28.85	27.30	31.25	30.55	31.10	30.40
	Known	34.50	33.15	32.05	31.80	33.60	33.10	32.75	31.45
MMLU	Unknown	54.02	53.43	64.34	64.14	54.02	53.63	55.26	55.45
	Known	66.62	66.69	66.95	66.75	56.89	57.09	59.70	59.96
WikiBios	Unknown	54.18	48.62			48.24	38.18		
	Known	54.81	50.63			48.54	36.48		

Table 7: Few-shot learning with Unknown examples

Benchmark	Split	LLaMA		LLaMA-Instruct		Mistral		Mistral-Instruct	
		ES	Con.	ES	Con.	ES	Con.	ES	Con.
EQ	Unknow	39.10	32.10	37.65	34.40	22.85	17.60	32.05	21.25
	Known	41.75	39.90	39.80	37.80	31.40	30.15	33.05	33.90
PopQA	Unknown	33.60	32.25	31.80	29.05	33.90	33.25	32.80	31.60
	Known	36.10	34.75	32.10	31.80	35.20	34.50	34.20	33.35
MMLU	Unknown	54.80	54.60	64.99	65.32	55.39	55.13	56.24	56.43
	Known	67.60	67.86	69.30	68.84	58.46	58.39	60.48	60.74
WikiBios	Unknown	53.72	47.03			47.93	35.53		
	Known	55.61	50.09			50.58	38.97		

Table 8: Few-shot learning with Known examples

Benchmark	Split	LLaMA		LLaMA-Instruct		Mistral		Mistral-Instruct	
		ES	Con.	ES	Con.	ES	Con.	ES	Con.
EQ	Unknow	41.55	38.95	41.00	37.40	35.35	32.95	35.25	30.05
	Known	43.45	42.20	41.20	40.70	38.25	37.95	33.15	32.65
PopQA	Unknown	39.45	39.35	35.55	35.30	33.05	33.40	32.55	32.25
	Known	39.50	39.15	34.35	35.80	34.70	34.10	33.95	32.50
MMLU	Unknown	45.79	47.35	64.34	64.01	53.04	53.49	58.00	60.09
	Known	56.56	56.83	65.12	65.45	56.50	58.13	61.07	60.94

Table 9: Few-shot learning with CoT

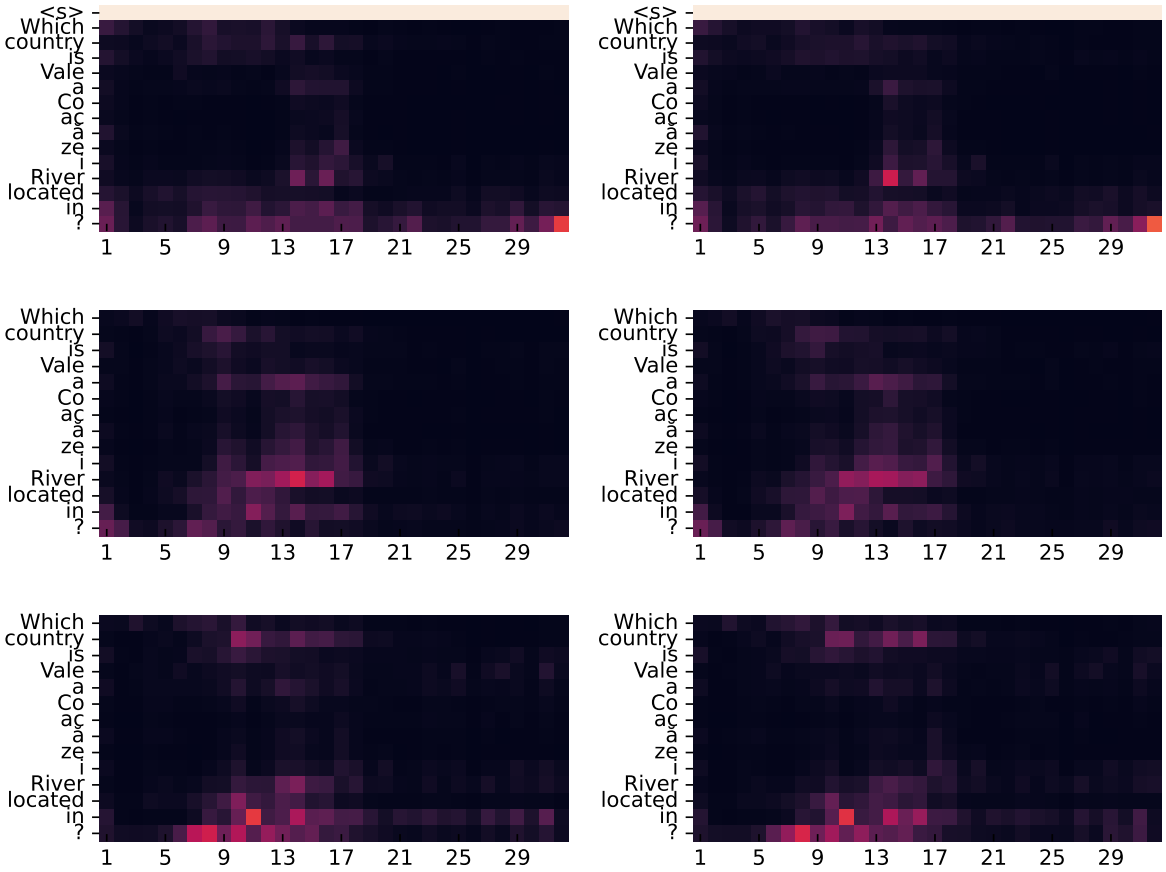

Figure 10: Attention maps of fine-tuned models. Top: Origin prompt. Middle: With few-shot learning Bottom: With CoT. Left: Fine-tuned on *Unknown* data. Right: Fine-tuned on *Known* data. Subject entity is "Valea Coacăzei River".

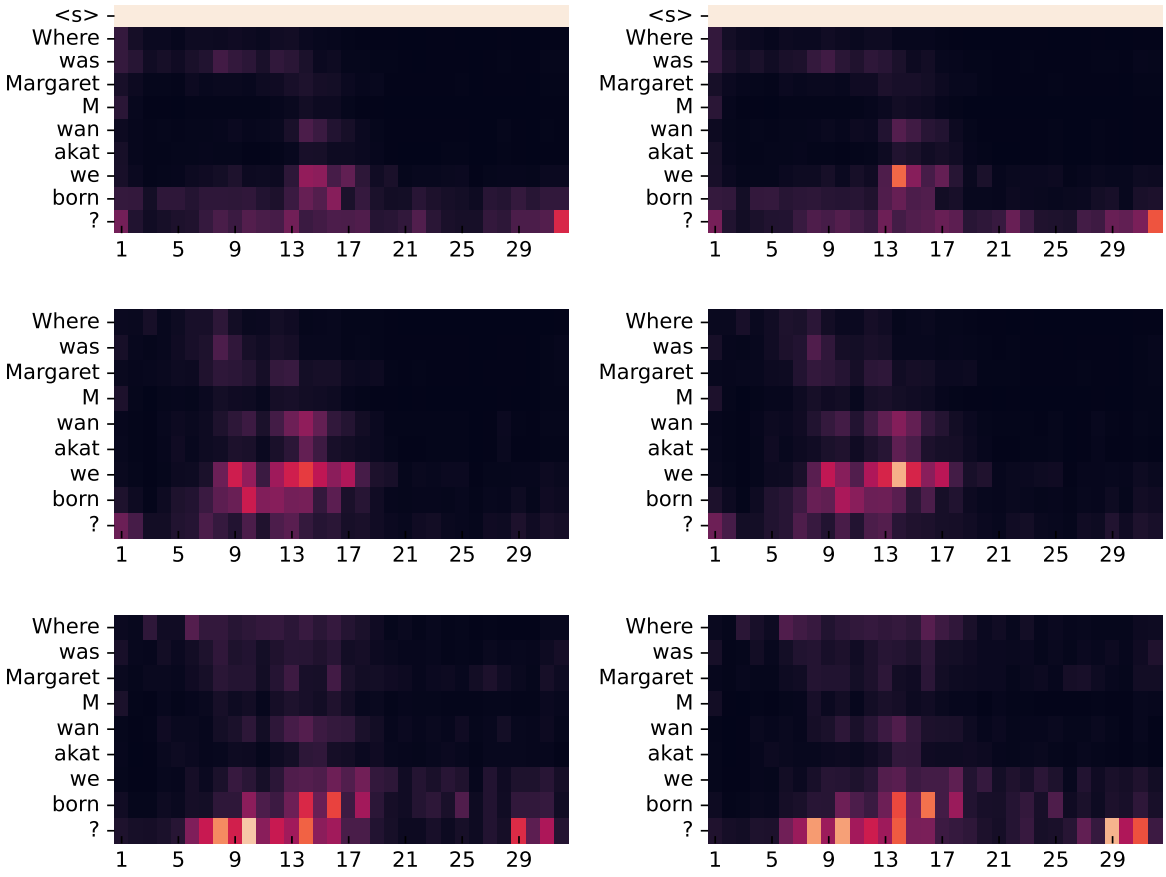

Figure 11: Attention maps of fine-tuned models. Top: Origin prompt. Middle: With few-shot learning Bottom: With CoT. Left: Fine-tuned on *Unknown* data. Right: Fine-tuned on *Known* data. Subject entity is "Margaret Mwanakatwe".