From Parameters to Prompts: Understanding and Mitigating the Factuality Gap between Fine-Tuned LLMs

Anonymous ACL submission

Abstract

Factual knowledge extraction aims to explicitly extract knowledge parameterized in pre-trained language models for application in downstream tasks. While prior work has been investigating the impact of supervised fine-tuning data on the factuality of large language models (LLMs), its mechanism remains poorly understood. We revisit this impact through systematic experiments, with a particular focus on the factuality gap that arises when fine-tuning on known versus unknown knowledge. Our findings show that this gap can be mitigated at the inference stage, either under out-of-distribution (OOD) settings or by using appropriate in-context learning (ICL) prompts (i.e., few-shot learning and Chain of Thought (CoT)). We prove this phenomenon theoretically from the perspective of knowledge graphs, showing that the test-time prompt may diminish or even overshadow the impact of fine-tuning data and play a dominant role in knowledge extraction. Ultimately, our results shed light on the interaction between finetuning data and test-time prompt, demonstrating that ICL can effectively compensate for shortcomings in fine-tuning data, and highlighting the need to reconsider the use of ICL prompting as a means to evaluate the effectiveness of fine-tuning data selection methods.

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., 2024b). 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

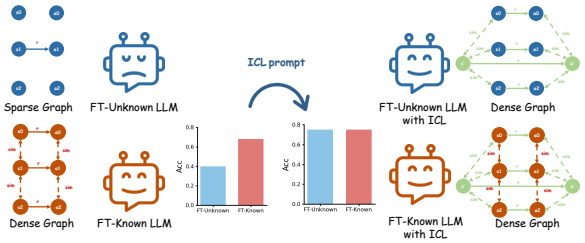

Figure 1: Overview: In-context learning (ICL) prompts can help reduce the factuality gap, as they enhance the connectivity of the graph of the FT-Unknown LLM by incorporating demonstrations like (s', a'), thereby narrowing the factuality gap. FT-Unknown LLM and FT-Known LLM refer to LLM fine-tuned on unknown and known knowledge, respectively.

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.

Previous research has extensively explored how different fine-tuning datasets impact the factuality of LLMs(Gekhman et al., 2024; Kazemi et al., 2023; Joshi et al., 2024; Ghosal et al., 2024). In this work, however, we find that this factuality gap caused by finetuning data is highly fragile. Modifying the test-time 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 caused by fine-tuning data can be understood from a novel perspective of knowledge graph modeling.

To gain deeper insight into the nature of this factuality gap, we pose the following three intriguing research questions: **RQ1:** *How to understand the factuality gap caused by finetuning data?* **RQ2:** *Can the factuality gap be easily mitigated?* **RQ3:** *What can we do to utilize this finding in knowledge extraction?* 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. In this paper, our main contributions can be summarized as follows:

- Through extensive experiments, we validate the existence of a factuality gap introduced by fine-tuning data and demonstrate that this gap diminishes as the distributional distance of the test set increases. Furthermore, we identify in-context learning at inference time as an effective approach to mitigate this gap.
- We conduct an in-depth analysis of the factuality gap and offer a deeper understanding from the perspective of knowledge graphs. To the best of our knowledge, we are the first to prove this phenomenon theoretically through the lens of graph modeling.
- Building on our empirical and theoretical work, we leverage this finding to explore its potential applications, especially introduce novel insights into the evaluation of data selection algorithms.

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, we study the interaction between model fine-tuning and inference to offer a more comprehensive analysis of factual knowledge extraction.

2.2 Finetuning Data and Model Factuality

Recent studies have explored the impact of fine-tuning data on model factuality. Kang et al. (2024) suggests 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) shows that finetuning on lesser-known facts leads to worse factuality because of less attention on the entity tokens during training. Lin et al. (2024b); Liu et al. (2024b) attempt to improve the factuality of the model by refining the data used for fine-tuning. Extending prior work, we examine the impact of fine-tuning data on model factuality from the graph modeling angle, and propose a method to reduce its adverse effects.

2.3 In-context Learning and Model Factuality

As a test-time method, ICL plays an important role in LLM knowledge extraction capabilities, and many studies have explored how ICL affects model factuality. Some works focus on using ICL for knowledge editing (Zheng et al., 2023), while others investigate how the construction of ICL examples influences knowledge extraction (Wang et al., 2024a; Wu et al., 2025; Yang et al., 2024; Lin et al., 2024a). In contrast, our work steps further to study the impact of ICL prompts on the factuality of fine-tuned models, rather than improving the ICL construction itself.

3 Preliminaries and Setup

3.1 Factual Knowledge

Following prior work on factual knowledge in LLMs (Ghosal et al., 2024; Petroni et al., 2019b), we represent each factual statement as a triplet (s, r, a), where s is the subject entity, r is the relation type, and a is the answer. This triplet structure is widely used in benchmarks such as LAMA (Petroni et al., 2019b), KILT (Petroni et al., 2021), and TruthfulQA (Lin et al., 2022). Formally, we denote a piece of knowledge as $k = (s, r, a) \in \mathcal{S} \times \mathcal{R} \times \mathcal{A}$, where $\mathcal{S}$, $\mathcal{R}$, and $\mathcal{A}$ represent the sets of all subject entities, relation types, and answers, respectively. This abstraction provides a unified format for evaluating whether a language model contains and can retrieve specific facts.

3.2 Knowledge Extraction in LLMs

To analyze the mechanism of knowledge extraction, we consider a simplified one-layer transformer architecture, with fixed non-orthogonal embed-

dings $E \in \mathbb{R}^{|\mathcal{T}| \times d}$ and the vocabulary $\mathcal{T}$. An input sequence of n tokens is written as $X = (x_1, \ldots, x_n) \in \mathcal{T}$. The model computes its outputs as

$$f(X; W^{KQ}, W^V) = \sigma(\text{Att}(E(X); W^{KQ}, W^V)),$$

where $W^{KQ}, W^V \in \mathbb{R}^{d \times d}$ are learnable parameters, Att() is the self-attention function and $\sigma()$ is the function that predict next token from the probability distribution. In this paper, we focus on the prediction for the next token, given by the final output vector $f_{[:,-1]}(X; W, V)$. For more detailed model settings, please refer to Appendix A.1.

Given a factual triplet $(s, r, a) \in \mathcal{T}^3$, we ask whether the model can retrieve the answer a when provided with an appropriate context. Specifically, we allow any context sequence $\{x_1, \ldots, x_{n-1}\} \in \mathcal{T} \setminus \{s, r, a\}$, with s and r. If the model predicts a as the next token, we say that the knowledge has been successfully extracted.

Definition 3.1 (Unknown Knowledge)**.** *A token triple $(s, r, a) \in \mathcal{T}^3$ is said to be an **unknown knowledge** if, for all contexts $\{x_1, \ldots, x_{n-1}\} \subset \mathcal{T} \setminus \{s, r, a\}$, $f_{[:,-1]}(x_1, \ldots, x_{n-1}, s, r) \neq a$.*

Definition 3.2 (Known Knowledge)**.** *A token triple $(s, r, a) \in \mathcal{T}^3$ is said to be a **known knowledge** if there exists a context $\{x_1, \ldots, x_{n-1}\} \subset \mathcal{T} \setminus \{s, r, a\}$ such that $f_{[:,-1]}(x_1, \ldots, x_{n-1}, s, r) = a$.*

In practice, we approximate this distinction using few-shot prompting. A triplet is considered known if the model produces the correct answer in at least one prompt. Otherwise, it is treated as unknown. This empirical definition enables generalization across prompt templates while preserving alignment with the formal setting above.

4 Understanding the Factuality Gap from Finetuning on Known vs Unknown Knowledge (RQ1)

In this section, we examine the factuality gap in models fine-tuned on known versus unknown knowledge, under both in-distribution and out-of-distribution scenarios. We present comprehensive experimental observations and support them with corresponding theoretical analysis.

4.1 Factuality Gap under In-distribution Generalization

Settings. We evaluate the impact of fine-tuning on known versus unknown factual knowledge across two task settings: QA and open-ended generation. For QA task, we follow the experimental protocol of Gekhman et al. (2024), fine-tuning both base and instruction-tuned variants of LLaMA3.1-8B [1] and Mistral-7B-v0.3 [2] on known and unknown subsets derived from EntityQuestions (Sciavolino et al., 2021), PopQA (Mallen et al., 2023), and MMLU (Hendrycks et al., 2020). Exact match accuracy is used as the evaluation metric. For open-ended generation task, we follow Kang et al. (2024) using the WikiBios dataset (Stranisci et al., 2023). The dataset is split analogously into known and unknown subsets, and performance is measured using the FActScore metric (Min et al., 2023). All models are evaluated under both early stopping and full convergence conditions. Implementation details are provided in Appendix B.

Obs. 1: Factuality gaps widen with training but are consistently smaller in instruction-tuned models. Table 1 reports results for models at early stopping and full convergence. We observe that the average factuality gap increases as training progresses. Across both Llama and Mistral architectures, instruction-tuned models consistently exhibit smaller gaps than their base counterparts. This pattern also holds on the WikiBios dataset.

4.2 Factuality Gap under Out-of-distribution Generalization

Settings. Beyond in-distribution (ID) generalization, we extend factuality generalization into other two out-of-distribution (OOD) types based on the distance between the test and training 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[3] embedding model (Reimers and Gurevych, 2019) to extract and process data patterns from both OOD and 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 in Section 4.1. For near in-distribution tasks, we sample non-overlapping data from the Entity Questions

[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}

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

Dataset	Split	Llama		Llama-Instruct		Mistral		Mistral-Instruct	
		ES	Con.	ES	Con.	ES	Con.	ES	Con.
EQ	Unknown	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
PopQA	Unknown	31.28	26.98	30.09	27.33	26.94	20.54	25.26	19.59
	Known	36.81	35.55	35.86	35.09	33.00	31.67	32.26	31.63
MMLU	Unknown	34.94	33.90	33.64	33.51	28.09	26.52	31.61	25.87
	Known	37.49	37.10	35.92	34.88	35.60	34.81	33.44	32.14
WikiBios	Unknown	55.50	46.90			47.30	36.67		
	Known	58.25	49.69			49.16	39.58		

Table 1: QA tasks exact match accuracy and WikiBios FActScore evaluation. ES: Early Stop, Con.: Convergence. Llama: Llama-3.1-8B, Llama-Instrcut: Llama-3.1-8B-Instruct, Mistral: Mistral-7B-v0.3, Mistral-Instruct: Mistral-7B-Instruct-v0.3

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 about experiments can be found in Appendix B.4.

Obs. 2: Factuality gaps persist on the OOD data but vanish under strong distribution shifts. As shown in Table 2, Llama3.1-8B fine-tuned on known data consistently outperforms its unknown-trained counterpart on both `eq_ood` and `pop_ood`, with gaps of 9% and 4% at early stopping, and 7.5% and 8% at convergence. Similar trends hold for Mistral. However, on the `mmlu_ood` dataset, which is more semantically distant, the factuality gap nearly disappears across all models.

4.3 A Graph-Theoretic Understanding of Factuality Gap

Theoretical Insight. We present a formal graph-theoretic framework for analyzing factuality in LLMs. Prior work has explored knowledge extraction empirically using graphs (Tang et al., 2024; Liu et al., 2024a), but lacks a principled account of generalization. We show that fine-tuning induces an edge-completion process over a latent knowledge graph, where one-hop connectivity captures factual prediction. This explains why known knowledge enables stronger generalization and why the factuality gap vanishes under semantic shift. Our analysis provides the first theoretical explanation of factuality emergence and decay in LLMs.

Let $\mathcal{G}_r = (\mathcal{V}, \mathcal{E}_r, \mathcal{E}^{\text{sim}})$ be a directed graph defined under a specific relation r. The node set $\mathcal{V} = \{t_1, t_2, \ldots, t_{|\mathcal{V}|}\}$ consists of entity tokens drawn from the LLM's token space. The edge set $\mathcal{E}_r = \{(v_s, v_a) \in \mathcal{V}^2 \mid f_{[:,-1]}(s, r) = a\}$ captures explicit relational knowledge: an edge from v_s to v_a exists if the model, when given the input token sequence (s, r), predicts a as the next token. The similarity edge set $\mathcal{E}^{\text{sim}}_t = \{(v_t, v_{t'}) \in \{t\} \times \mathcal{V} \mid t' \neq t \text{ and } \|t - t'\|_2 \leq \epsilon\}$ represents implicit connections based on embedding similarity: an edge from t to t' exists if the distance between their embeddings is less than ϵ. A more detailed definition is provided in Appendix A.1.

Memorizing Knowledge as an Edge-Completion Process. Under the graph-theoretic formulation, SFT can be viewed as an edge-completion process through which the LLM acquires new **one-hop** knowledge. Formally, this corresponds to augmenting the internal knowledge graph by adding edges that connect previously disjoint or weakly connected subgraphs, thereby encoding new relational facts into the model.

Lemma 4.1 (Memorizing Knowledge as an Edge–Completion Process)**.** *Let $\mathcal{D}_r$ be the training dataset for relation r. For a knowledge triple $k = (s, r, a) \in \mathcal{D}_r$, let $\mathcal{G}_s = (\mathcal{V}_s, \mathcal{E}^{sim}_s)$ and $\mathcal{G}_a = (\mathcal{V}_a, \mathcal{E}^{sim}_a)$ be the subgraphs connected to s and a via similarity edges. Then after memorizing k, the relation graph $\mathcal{G}_r$ is updated as $\mathcal{G}_r \leftarrow \mathcal{G}_r \cup \{(v_i, v_j) \mid (v_i, v_j) \in \mathcal{V}_s \times \mathcal{V}_a,\ f_{[:,-1]}(i, r) = j\}$.*

Remark 1. *We interpret the memorization of knowledge in LLMs as an edge-completion process on the relation graph. The formal justification is provided in Appendix A.2. Notably, while the update considers all candidate pairs in $\mathcal{V}_s \times \mathcal{V}_a$, only a subset of edges,specifically those satisfying $f_{[:,-1]}(i, r) = j$, are actually added. In particular, there always exists at least one edge (v_s, v_a) added to the graph.*

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

The generalization capability of SFT is reflected in the emergence of new connections between previously unlinked subgraphs. Let $\mathcal{G}'_r = (\mathcal{V}, \mathcal{E}'_r, \mathcal{E}^{\text{sim}})$ denote the relation graph internal to the LLM after fine-tuning. If there exists a pair $(s', a') \in \mathcal{V}_s \times \mathcal{V}_a$ such that the corresponding knowledge triple $(s', r, a') \notin \mathcal{D}_r$, and $(v_{s'}, v_{a'}) \notin \mathcal{E}_r$ but $(v_{s'}, v_{a'}) \in \mathcal{E}'_r$, then the model has successfully generalized beyond the training data by inferring the unseen triple (s', r, a').

Factuality Gap Explained via Differential Connectivity. If a knowledge triple (s, r, a) is present in the training set $\mathcal{D}_r$, few-shot prompting can be viewed as temporarily injecting edges $\{(v_{s'_i}, v_{a'_i})\}$, where each (s'_i, r, a'_i) is a support triple, to connect $\mathcal{V}_s$ and $\mathcal{V}_a$. This mechanism will be presented in Section 5.3. Unknown knowledge under few-shot prompting typically arises when the connectivity between v_s and $v_{s'_i}$, or between v_a and $v_{a'_i}$, is weak, which is often due to sparsity in the induced subgraphs, particularly when s or a corresponds to a low-degree entity. As a result, fine-tuning on such unknown knowledge induces only limited updates to the relation graph, thereby reducing the model's capacity to generalize across the domain r. Figure 2 illustrates the process of adding edges when LLM finetuned on different types of knowledge.

Theorem 4.1 (Factuality Gap as a Connectivity Gap in Knowledge Graphs)**.** *Let $\mathcal{G}_{kn} = (\mathcal{V}, \mathcal{E}_{kn}, \mathcal{E}^{sim})$ and $\mathcal{G}_{unk} = (\mathcal{V}, \mathcal{E}_{unk}, \mathcal{E}^{sim})$ be knowledge graphs induced by LLMs fine-tuned on known and unknown knowledge, respectively. Let (s, r, a) be a test triple sampled uniformly at random from a fixed test set. Define indicator variables $Z_{kn} = \mathbf{1}\{(v_s, v_a) \in \mathcal{E}_{kn}\}$ and $Z_{unk} = \mathbf{1}\{(v_s, v_a) \in \mathcal{E}_{unk}\}$. Assume edges under relation r are uniformly distributed and test triples are uniformly sampled over their support. Then the expected factuality gap satisfies $\mathbb{E}[Z_{kn} - Z_{unk}] = \Delta_{fact} \propto |\mathcal{E}_{kn}| - |\mathcal{E}_{unk}| > 0$.*

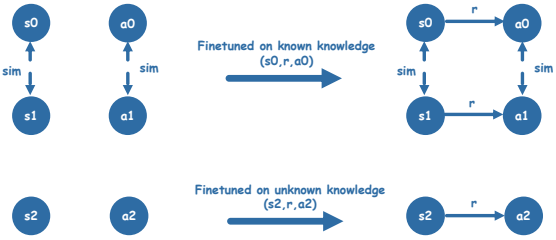

Figure 2: Memorizing a known knowledge triple (s_0, r, a_0) generalizes to memorizing (s_1, r, a_1) but memorizing an unknown knowledge triple (s_2, r, a_2) can not generalize.

Remark 2. *Theorem 4.1 interprets the factuality gap as a direct consequence of differences in one-hop connectivity induced by fine-tuning. Under uniform sampling assumptions, the expected gap in one-hop accuracy reflects the difference in the number of factual edges established in the graph. A detailed proof is provided in Appendix A.3.*

OOD Generalization and the Vanishing Gap. As the test distribution diverges from the training graph structure, both known and unknown knowledge graph $\mathcal{G}_{\text{kn}}, \mathcal{G}_{\text{unk}}$ become equally less overlapped to the OOD knowledge graph. Consequently, as the knowledge graph corresponding to the training data domain becomes nearly disjoint from that of the test data domain, the factuality gap approaches zero.

Theorem 4.2 (Decay of Factuality Gap Under Distributional Shift)**.** *Let $\cos\langle \mathcal{D}_{test}, \mathcal{D}_{train} \rangle := \mathbb{E}_{x \sim \mathcal{D}_{test},\, x' \sim \mathcal{D}_{train}} \langle x, x' \rangle$ denote the semantic similarity between the test and training distributions, where x, x' are unit-normalized representations. Then, the factuality gap Δ_{fact} under OOD evaluation decreases as the semantic similarity vanishes: if $\cos\langle \mathcal{D}_{test}, \mathcal{D}_{train} \rangle \to 0$, the factuality gap $\Delta_{fact} \to 0$.*

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	39.82$_{+8.54}$	37.89$_{+10.91}$	35.06$_{+4.97}$	34.01$_{+6.68}$	35.93$_{+8.99}$	35.76$_{+15.22}$	31.46$_{+6.20}$	31.32$_{+11.73}$
	K	38.77$_{+1.96}$	38.66$_{+3.11}$	35.55$_{-0.31}$	36.18$_{+1.09}$	35.93$_{+2.93}$	35.90$_{+4.23}$	31.63$_{-0.63}$	31.84$_{+0.21}$
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$_{+33.40}$	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 3: 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.

Remark 3. *In practice, we compute the semantic similarity using an external embedding model. We assume that the resulting scores closely approximate those that would be obtained using the internal representations of the LLM. A formal proof of Theorem 4.2 is provided in Appendix A.4.*

5 Can Fatcuality Gap be Easily Mitigated? (RQ2)

5.1 ICL Mitigates the Factuality Gap

Settings. We evaluate all models and tasks from Section 4 using few-shot and few-shot CoT prompting. Few-shot examples are selected from the *Known* training data. For CoT, GPT-4o[4] generates entity-level analyses to construct reasoning chains, which are integrated into the CoT prompts. The format is shown below.

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

We construct three prompt sets and evaluate two prompting variants: with and without CoT. All models, including *Known* and *Unknown*, are evaluated using the same prompts. The prompt set yielding the highest performance on the *Unknown* model is reported. For generation tasks, we use few-shot prompting only, following the same example selection strategy. Full prompt details are provided in Appendix C.

Obs. 3: In-context learning narrows the factuality gap, especially with few-shot CoT. Table 3 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* model, especially in the case of Convergence. 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.

5.2 Ablation study

To better understand how in-context learning mitigates the factuality gap, we conduct ablation experiments using the Llama-3.1-8B model on the Entity Questions dataset. Full details are provided in Appendix D.

Obs. 4: All of example source, CoT reasoning and question format critically affect factual generalization. We conduct an ablation study on the composition of the prompt, separately examining the source 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. 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,

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

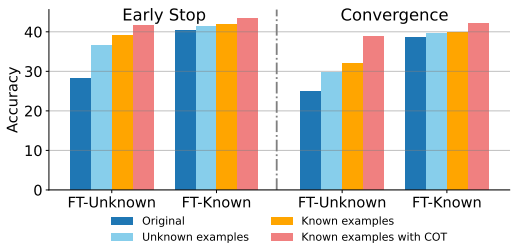

Figure 3: Ablation study of few-shot examples and CoT.

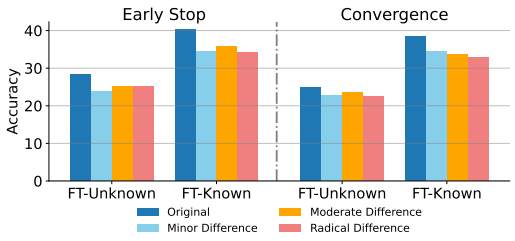

Figure 4: Ablation study of prompt formulation. We use three levels of rephrasing: Minor, Moderate, Radical.

and the factuality gap remains large, which is illustrated in Figure 4.

5.3 Understanding the Role of ICL

ICL prompts act as subgraph injections that reduce the factuality gap. We present a new theoretical perspective on ICL: few-shot examples and CoT rationales can be interpreted as *prompt-induced subgraphs* that augment the model's internal knowledge graph during inference. Given a prompt Π containing n support triples (s_i, r, a_i), we treat it as an auxiliary knowledge graph $\mathcal{G}_\Pi = (\mathcal{V}_\Pi, \mathcal{E}_{r_\Pi}, \mathcal{E}_\Pi^{\text{sim}})$. For CoT prompting, a target triple (s, r, a) is supported by a structured reasoning chain $C = \{(s, r_i, a_i) \mid 1 \le i \le n,\ a_n = a\}$, where all steps share the subject s, and relations r_i may differ. This defines an additional support graph $\mathcal{G}_C$. At inference time, the model operates on an augmented graph $\mathcal{G}^\star = \mathcal{G} \cup \mathcal{G}_\Pi \cup \mathcal{G}_C$, where $\mathcal{G}$ is the base knowledge graph encoded by the fine-tuned model, and $\mathcal{G}_\Pi \cup \mathcal{G}_C$ are injected through the ICL prompt.

Theorem 5.1 (ICL Prompt Can Mitigate the Factuality Gap)**.** *Let $\mathcal{G}$ be the knowledge graph induced by an LLM after fine-tuning, and let $\mathcal{P}$ be a valid in-context prompt represented as an auxiliary graph $\mathcal{G}_\mathcal{P}$. The augmented graph at inference time is $\mathcal{G}^\star = \mathcal{G} \cup \mathcal{G}_\mathcal{P}$. Then, the factuality gap under prompt-augmented inference satisfies $\Delta^\star_{fact} < \Delta_{fact}$.*

Remark 4. *This result provides a structural explanation for why in-context prompting improves factuality: it temporarily densifies connectivity between relevant subgraphs, effectively compensating for missing fine-tuned edges. Please refer to Appendix A.5 for the proof of Theorem 5.1.*

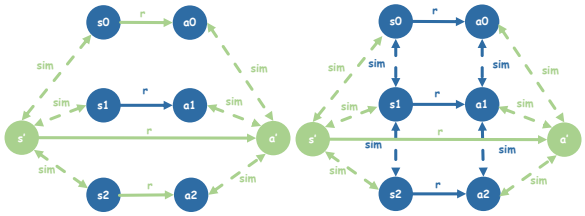

Figure 5: In an LLM fine-tuned on unknown knowledge (left), the demonstration (s', r, a') introduces new edges (s_0, a_0) and (s_2, a_2). In contrast, for the LLM fine-tuned on known knowledge (right), these edges already exist and thus are not newly added. Consequently, the factuality gap narrows as the difference in the number of edges between the two graphs decreases.

6 Leveraging ICL for Knowledge Extraction (RQ3)

6.1 Improving Generalization under Limited or Noisy Supervision

Building on our theoretical insights, we hypothesize that ICL can improve factual generalization not only in the presence of low-quality fine-tuning data, but also when the available data is limited. To test this, we conduct an experiment on the PopQA dataset, comparing two conditions: (1) applying ICL after fine-tuning on a random 5% subset of the training data, and (2) applying ICL after full-data fine-tuning. As shown in Figure 6, the 5%-trained model achieves performance comparable to the full-data model when combined with ICL. Full details are provided in Appendix B.5. These findings suggest that well-designed ICL prompts can effectively compensate for limited or low-quality supervision in the knowledge extraction of LLM.

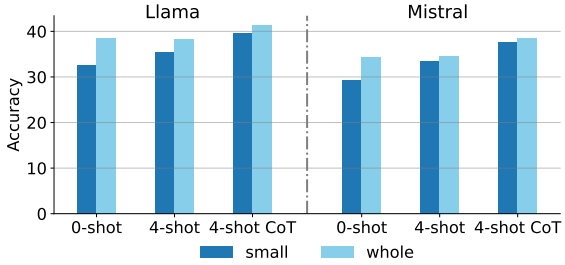

Figure 6: Comparison between Llama-3.1-8B and Mistral-7B-v0.3 models fine-tuned on 5% of dataset and the whole dataset.

6.2 Rethinking the Metric for Finetuning Data Selection Method

Recent studies on data selection algorithms for finetuning commonly evaluate performance using few-shot prompting (Liu et al., 2024b; Xia et al., 2024). However, our theoretical and empirical findings suggest that in-context learning can significantly reduce, and in some cases even eliminate, the performance differences arising from variations in training data. Consequently, evaluations based solely on few-shot prompting may mask the true effectiveness of data selection methods. We therefore argue that a more comprehensive evaluation framework is necessary to reliably assess the performance of data selection algorithms.

7 Discussion: How Far Can CoT Go?

Toy Example Setup. 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[5] 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. We fine-tune the models using LoRA, and evaluate their performance on the test set, which shares the same data type as the training set, i.e., normal (known) or perturbed (unknown). Additionally, we also add special CoT to the Toy Example for verification. Detailed prompt design is shown in the box below. Experiment details are presented in Appendix B.3.

CoT for Toy Example

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 a short explanation.

Obs 5: Under controlled perturbations, the factuality gap remains large, but is substantially reduced by CoT prompting. As shown in Table 4, we observe consistent gaps in factuality across models fine-tuned on known and unknown knowledge sets. The results further confirm that unknown knowledge encourages factuality failure. We also observe that CoT effectively enhances model testing performance and narrows the factuality gap between the two 70B models.

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

Table 4: Performance of Toy Example

Discussion. For more powerful 70B models, fine-tuning on both known and unknown knowledge can still lead to a factuality gap. However, the way these models mitigate the gap through in-context learning differs significantly from the approach discussed above. This mitigation is achieved by using instructions to directly establish a connection between perturbed entities and normal entities, which then enables correct knowledge extraction. These results demonstrate that CoT is powerful enough to bypass the mapping established during the fine-tuning stage, allowing the model to respond based on the new mapping defined within the CoT prompt. This highlights the effectiveness of prompt-based reasoning in decoupling model behavior from parameter-level modifications.

8 Conclusion

This work provides both theoretical and empirical investigations of the factuality gap introduced by fine-tuning LLMs on known versus unknown knowledge. Based on the analysis of experimental phenomena, we further attempt to explain and investigate this gap from a graph-theoretic perspective, viewing the process of knowledge extraction as a problem of graph connectivity and structural completeness. This theoretical framework reveals the interaction mechanism between fine-tuning and test-time ICL prompts, uncovering how prompt-based reasoning compensates for parameter-induced limitations. In summary, in this paper, we offer a new perspective on the factual behavior of LLMs, providing foundational insights into factual generalization that can inform data selection strategies, prompt design, model interpretability, and the deployment of models in knowledge-intensive tasks.

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

Limitations

The proposed framework is derived from empirical observations and may lack full formal generality. Some underlying assumptions may not fully capture model behavior across diverse domains, architectures, or prompt formats. In particular, this work does not fully explain the anomalous behavior observed on datasets such as MMLU and WikiBios, which may involve more complex or multimodal factual structures. We hope this work encourages future efforts to refine the theoretical framework, extend it to broader task types, and develop more robust explanations for these challenging settings.

References

Ekin Akyürek, Dale Schuurmans, Jacob Andreas, Tengyu Ma, and Denny Zhou. 2023. What learning algorithm is in-context learning? investigations with linear models. In *The Eleventh International Conference on Learning Representations*.

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.

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.

Bill Yuchen Lin, Abhilasha Ravichander, Ximing Lu, Nouha Dziri, Melanie Sclar, Khyathi Chandu, Chandra Bhagavatula, and Yejin Choi. 2024a. The unlocking spell on base LLMs: Rethinking alignment via in-context learning. In *The Twelfth International Conference on Learning Representations*.

Sheng-Chieh Lin, Luyu Gao, Barlas Oguz, Wenhan Xiong, Jimmy Lin, Wen-tau Yih, and Xilun Chen. 2024b. Flame: Factuality-aware alignment for large language models. *arXiv preprint arXiv:2405.01525*.

Stephanie Lin, Jacob Hilton, and Owain Evans. 2022. TruthfulQA: Measuring how models mimic human falsehoods. In *Proceedings of the 60th Annual Meeting of the Association for Computational Linguistics (Volume 1: Long Papers)*, pages 3214–3252, Dublin, Ireland. Association for Computational Linguistics.

Haochen Liu, Song Wang, Yaochen Zhu, Yushun Dong, and Jundong Li. 2024a. Knowledge graph-enhanced large language models via path selection. In *Findings of the Association for Computational Linguistics: ACL 2024*, pages 6311–6321, Bangkok, Thailand. Association for Computational Linguistics.

Zifan Liu, Amin Karbasi, and Theodoros Rekatsinas. 2024b. TSDS: Data selection for task-specific model finetuning. In *The Thirty-eighth Annual Conference on Neural Information Processing Systems*.

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, Aleksandra Piktus, Angela Fan, Patrick Lewis, Majid Yazdani, Nicola De Cao, James Thorne, Yacine Jernite, Vladimir Karpukhin, Jean Maillard, Vassilis Plachouras, Tim Rocktäschel, and Sebastian Riedel. 2021. KILT: a benchmark for knowledge intensive language tasks. In *Proceedings of the 2021 Conference of the North American Chapter of the Association for Computational Linguistics: Human Language Technologies*, pages 2523–2544, Online. 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.

Jiabin Tang, Yuhao Yang, Wei Wei, Lei Shi, Lixin Su, Suqi Cheng, Dawei Yin, and Chao Huang. 2024. Graphgpt: Graph instruction tuning for large language models. In *SIGIR*, pages 491–500.

Johannes Von Oswald, Eyvind Niklasson, Ettore Randazzo, Joao Sacramento, Alexander Mordvintsev, Andrey Zhmoginov, and Max Vladymyrov. 2023. Transformers learn in-context by gradient descent. In *Proceedings of the 40th International Conference on Machine Learning*, volume 202 of *Proceedings of Machine Learning Research*, pages 35151–35174. PMLR.

Jianing Wang, Chengyu Wang, Chuanqi Tan, Jun Huang, and Ming Gao. 2024a. Knowledgeable in-context tuning: Exploring and exploiting factual knowledge for in-context learning. In *Findings of the Association for Computational Linguistics: NAACL 2024*, pages 3261–3280, Mexico City, Mexico. Association for Computational Linguistics.

Zhihu Wang, Shiwan Zhao, Yu Wang, Heyuan Huang, Sitao Xie, Yubo Zhang, Jiaxin Shi, Zhixing Wang, Hongyan Li, and Junchi Yan. 2024b. 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.

Qinyuan Wu, Mohammad Aflah Khan, Soumi Das, Vedant Nanda, Bishwamittra Ghosh, Camila Kolling, Till Speicher, Laurent Bindschaedler, Krishna Gummadi, and Evimaria Terzi. 2025. Towards reliable latent knowledge estimation in llms: Zero-prompt many-shot based factual knowledge extraction. In *Proceedings of the Eighteenth ACM International Conference on Web Search and Data Mining*, WSDM '25, page 754–763. ACM.

Mengzhou Xia, Sadhika Malladi, Suchin Gururangan, Sanjeev Arora, and Danqi Chen. 2024. LESS: Selecting influential data for targeted instruction tuning. In *International Conference on Machine Learning (ICML)*.

Linyi Yang, Shuibai Zhang, Zhuohao Yu, Guangsheng Bao, Yidong Wang, Jindong Wang, Ruochen Xu, Wei Ye, Xing Xie, Weizhu Chen, and Yue Zhang. 2024. Supervised knowledge makes large language models better in-context learners. In *The Twelfth International Conference on Learning Representations*.

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.

Ce Zheng, Lei Li, Qingxiu Dong, Yuxuan Fan, Zhiyong Wu, Jingjing Xu, and Baobao Chang. 2023. Can we edit factual knowledge by in-context learning? In *Proceedings of the 2023 Conference on Empirical Methods in Natural Language Processing*, pages 4862–4876, Singapore. Association for Computational Linguistics.

A Theory Work

A.1 Notation and Setup

Embedding Layer We define the embedding matrix $E \in \mathbb{R}^{|\mathcal{T}|\times d}$, where the i-th row $E[i] = E_{t_i}$ is the (non-orthogonal) embedding vector of token t_i, and the un-embedding matrix is $U \in \mathbb{R}^{d\times|\mathcal{T}|}$. The matrices E and U are weight-tied and are learned during pretraining.

We assume the embeddings are non-orthogonal and fixed during finetuning, this setting reflects realistic language model behavior and allows us to define local neighborhoods over tokens via euclidean distance similarity in the embedding space. Specifically, we define:

$$t_i \sim_\epsilon t_j \iff \|E[i] - E[j]\| < \epsilon,$$

which enables generalization from seen tokens to nearby tokens in the semantic space.

One-Layer Transformer Architecture We consider a one-headed, one-layer transformer with untied, learned embedding $E \in \mathbb{R}^{|\mathcal{T}|\times d}$ and un-embedding $U \in \mathbb{R}^{d\times|\mathcal{T}|}$ matrices. For a prompt (s, r), the input embedding matrix is $X = [E_s, E_r] \in \mathbb{R}^{d\times 2}$, where E_s, E_r are the continuous token embeddings.

Let $W^{KQ} = (W^K)^\top W^Q \in \mathbb{R}^{d\times d}$ and $W^V \in \mathbb{R}^{d\times d}$. The attention weights are:

$$\begin{aligned}\alpha &= \text{softmax}(X^\top W^{KQ} X_{:,-1}) \\ &= \text{softmax}(\begin{bmatrix} (W^{KQ})_{s,r} \\ (W^{KQ})_{r,r} \end{bmatrix}) \\ &= \begin{bmatrix} \alpha_s \\ \alpha_r \end{bmatrix}.\end{aligned}$$

where the subscript $:, -1$ denotes the last column of the matrix. Thus, we take the softmax of the (post-self-attention) embedding of the last input token to predict the next token. The hidden state is:

$$h(s, r) = W^V X\alpha = \alpha_s W^V E_s + \alpha_r W^V E_r.$$

The logits for token i are computed via:

$$z_i(s, r) = U_{:,i}^\top h(s, r),$$

and the output distribution is:

$$p_\theta(i \mid s, r) = \text{softmax}_i(z(s, r)).$$

A.2 Proof of Lemma 4.1

In Lemma 4.1, we characterize the SFT process as adding edges between the connected subgraph of v_s and the connected subgraph of v_a in the LLM's knowledge graph. We now provide a proof of this statement.

Proof. We assume a standard cross-entropy loss on the output, and we perform a gradient update (SGD step) on the model parameters using the example (s, r, a). Let $p_\theta(x \mid s, r)$ denote the model's predicted probability for token x as the answer given (s, r). The cross-entropy loss for the correct answer a is

$$\mathcal{L} = -\log p_\theta(a \mid s, r).$$

This loss pushes the model to increase the probability of a while decreasing the probability of other tokens for the input (s, r). Then, we examine the gradients with respect to various components. The gradient of $\mathcal{L}$ with respect to the hidden states for any token x is

$$\begin{aligned}\delta_h &= \frac{\partial \mathcal{L}}{\partial h(s, r)} \\ &= \sum_x \frac{\partial \mathcal{L}}{\partial z_x(s, r)} \cdot \frac{\partial z_x(s, r)}{\partial h(s, r)} \\ &= \sum_x (p_\theta(x \mid s, r) - \mathbb{I}\{x = a\}) \cdot U_{:,x}\end{aligned}$$

where $\mathbb{I}\{x = a\}$ is 1 for $x = a$ and 0 otherwise.δ_h points in the direction that increases the logit for a and decreases logits for others. Then, SGD update (with learning rate η) for W^V, and the new value vector for any other token $i \in V_s$ after updated is:

$$\begin{aligned}v_i^{\text{new}} &= (W^V + \Delta W^V)E[i] \\ &= W^V E[i] + \eta \frac{\partial \mathcal{L}}{\partial W^V} \\ &= W^V E[i] + \eta\alpha_s\, \delta_h\, (E[s]^\top E[i]) \\ &\quad + \eta\alpha_r\, \delta_h\, (E[r]^\top E[i])\end{aligned}$$

Because $E[i] \approx E[s]$, the inner product $E[s]^\top E[i]$ will be close to $|E[s]|^2$ (and $E[r]^\top E[i]$ is presumably small unless r happened to be similar to s in embedding). Thus v_i gets a nearly identical adjustment in the δ_h direction. For any token $i \in V_s$, consider its key after the update:

$$\begin{aligned}k_i^{\text{new}} &= (W^K + \Delta W^K)E[i] \\ &= W^K E[i] \\ &\quad + \eta\, \alpha_s(1 - \alpha_s) \cdot \delta_h^\top W^V (E[s] - E[r]) \\ &\quad \cdot q_r(E[s]^\top - E[r]^\top)E[i]\end{aligned}$$

W^K is being adjusted so that the keys of s and all similar tokens i move closer in the direction of the relation's query q_r. This increases $q_r \cdot k_i$ for each such i, thus increasing the attention weight $\alpha_{r \to i}$ when the model processes (i, r) in the future.

The W^Q update also specifically adjusted $q_r = W^Q E[r]$ to better align with k_s. This change benefits any input where the key is similar to k_s. In particular, q_r will now have higher dot-product with k_i for any i in V_s (since $k_i^{new} \approx k_s^{new}$). Thus, both W^K and W^Q updates reinforce the attention to any subject token similar to s.

Now consider the forward pass for a new input (i, r) after the update. The new hidden state for (i, r) is then:

$$\alpha_i^{\text{new}} = \frac{\exp\left((q_r^{\text{new}})^\top k_i^{\text{new}}\right)}{\exp\left((q_r^{\text{new}})^\top k_i^{\text{new}}\right) + \exp\left((q_r^{\text{new}})^\top k_r^{\text{new}}\right)}$$

Given our analysis, $(q_r^{new})^\top k_i^{new}$ is significantly larger than the old $(q_r^{old})^\top k_i^{old}$, and also larger relative to $(q_r^{new})^\top k_r^{new}$. Since the update was based on s vs. r, we expect $(q_r^{new})^\top k_i^{new} \approx (q_r^{new})^\top k_s^{new}$ which was boosted. Thus α_i^{new} will be close to the α_s^{new} achieved for the training pair, which is likely near 1 if the model learned to almost fully attend to the subject. So the relation r will heavily attend to i:

$$\begin{aligned} h(i, r) &\approx \alpha_i^{\text{new}} v_i^{\text{new}} + \alpha_r^{\text{new}} v_r^{\text{new}} \\ &\approx v_i^{\text{new}} + (\text{small residual}). \end{aligned}$$

Because v_i^{new} was updated to be nearly v_s^{new} in the δ_h direction, and v_s^{new} was tuned to align with u_a, it follows that $h(i, r)$ points toward $U_{:,a}$ as well. In other words, the hidden representation the model computes from (i, r) is now oriented in a way that favors the answer a and similar tokens.

Since $h(i, r) \approx v_i^{new}$ and $v_i^{new} \approx v_s^{new}$ and v_s^{new} was pushed toward $U_{:,a}$, we have $z_a(i, r)$ greatly increased. The probabilities $P(x|i, r) = \text{softmax}(z(i, r))$ will assign much more mass to a and its neighbors. Therefore, the model's predicted answer token $j = f_{[:,-1]}(i, r)$ will lie in the neighborhood V_a. Symbolically, $f_{[:,-1]}(i, r) = j$ with $j \in V_a$. □

A.3 Proof of Theorem 4.1

In this section, we prove Theorem 4.1 and analyze why the unknown knowledge identified by few-shot prompting tends to correspond to nodes with lower degrees. Based on this observation, we further show that performing SFT on unknown knowledge results in a graph with fewer associated explicit edges, compared to the graph formed by fine-tuning on known knowledge.

Assumption A.1. *We assume that, when using few-shot prompting, the attention mechanism guides the query (s, r) to follow the patterns observed in the demonstrations (s'_i, r, a'_i) when predicting the answer.*

This assumption is reasonable based on prior work (Brown et al., 2020; Von Oswald et al., 2023; Akyürek et al., 2023), which demonstrates that language models can imitate demonstrated patterns via in-context learning.

Proof. In the transformer's attention mechanism, the weight placed on any key–value pair is

$$\alpha_t = \frac{\exp\left((W^Q E[r])^\top (W^K E[t])\right)}{\sum_{u \in \{s, r, s'_i, \dots\}} \exp\left((W^Q E[r])^\top (W^K E[u])\right)}.$$

Here, $q = W^Q E[r]$ is the query vector for the relation token, and each key vector $k_t = W^K E[t]$ corresponds to token t.

If a demonstration subject s'_i has an embedding $E[s'_i]$ so close to $E[s]$ that

$$\|E[s'_i] - E[s]\| < \epsilon,$$

then applying the same linear map W^K yields

$$k_{s'_i} = W^K E[s'_i] \approx W^K E[s] = k_s.$$

Because the two key vectors are nearly identical, their dot products with the query vector are also nearly the same:

$$q^\top k_{s'_i} \approx q^\top k_s.$$

According to Assumption A.1, the prediction for (s, r) follows the pattern established by the demonstrations (s'_i, r, a'_i). Based on the derivation in Appendix A.2, when

$$q^\top k_{s'_i} \approx q^\top k_s,$$

the resulting distribution $p_\theta(x \mid \dots, s, r)$ will place most of its mass near a'_i. In this case, if $a \sim_\epsilon a'_i$, then the probability of correctly predicting the target answer a increases significantly.

Therefore, for known knowledge where few-shot prompting successfully leads to correct predictions, the pairs (s, a) are typically close in the embedding space to many demonstration pairs (s'_i, a'_i). This

implies that $|\mathcal{V}_s|$, the size of the similarity neighborhood in the constructed graph $\mathcal{G}_s = (\mathcal{V}_s, \mathcal{E}^{\text{sim}})$, is relatively large. Similarly, $|\mathcal{V}_a|$ is also larger.

According to Lemma 4.1, this means that fine-tuning on known knowledge typically results in more than one new edge, while fine-tuning on unknown knowledge generally adds only one new edge. Thus, we have:

$$|\mathcal{E}_{\text{kn}}| > |\mathcal{E}_{\text{unk}}|.$$

Let D_{test} be a random sample of test triples (s, r, a). Under the *uniform-edge assumption* (i.e., every possible pair in $\mathcal{V} \times \mathcal{V}$ is equally likely to be included in $\mathcal{E}_r$), the probability that a test triple (s, r, a) is "in" the graph (i.e., can be answered correctly in one hop) is

$$\Pr\left((v_s, v_a) \in \mathcal{E}_r\right) = \frac{|\mathcal{E}_r|}{|\mathcal{V}|^2}.$$

Hence, the expected number of correctly answered test triples is

$$|D_{\text{test}}| \times \frac{|\mathcal{E}_r|}{|\mathcal{V}|^2}.$$

Define the *factuality gap* between known- and unknown-fine-tuning as

$$\begin{aligned}\Delta_{\text{fact}} = &\left|\{(v_s, v_a) \in E_{\text{kn}} \mid (s, r, a) \in D_{\text{test}}\}\right| \\ &- \left|\{(v_s, v_a) \in E_{\text{unk}} \mid (s, r, a) \in D_{\text{test}}\}\right|.\end{aligned}$$

Taking expectations under random sampling, we have:

$$\begin{aligned}\mathbb{E}[\Delta_{\text{fact}}] &= |D_{\text{test}}| \cdot \left(\frac{|\mathcal{E}_{\text{kn}}| - |\mathcal{E}_{\text{unk}}|}{|\mathcal{V}|^2}\right) \\ &\propto |\mathcal{E}_{\text{kn}}| - |\mathcal{E}_{\text{unk}}| \\ &> 0.\end{aligned}$$

That is,

$$\Delta_{\text{fact}} \propto |\mathcal{E}_{\text{kn}}| - |\mathcal{E}_{\text{unk}}| > 0,$$

which is exactly the statement of Theorem 4.1. □

A.4 Proof of Theorem 4.2

We make several foundational proofs and attempt to provide a graph-theoretic analysis showing that the greater the semantic distance between the test set and the training set, the smaller the observed factuality gap on the test set.

We begin by proving the relationship between cosine similarity and the edge connectivity of the knowledge graph associated with the dataset.

Proof. First, we assume that all token embeddings are unit-normalized, so for any two tokens i, j

$$||e_i|| = ||e_j|| = 1.$$

Their Euclidean distance and cosine similarity are related, and under the neighborhood condition there is

$$||e_i - e_j|| < \epsilon \iff \cos(e_i, e_j) > 1 - \frac{\epsilon^2}{2}.$$

Let

$$\gamma = \mathbb{E}_{(s_{\text{test}}, s_{\text{train}})}\left[\cos\left(e_{s_{\text{test}}}, e_{s_{\text{train}}}\right)\right]$$

be the average cosine similarity between a random test subject embedding and a random training subject embedding. Define the threshold

$$\tau = 1 - \frac{\epsilon^2}{2}.$$

Then, by Markov's inequality, the fraction of test–training pairs whose cosine exceeds τ is bounded above by

$$\Pr\left(\cos(e_{s_{\text{test}}}, e_{s_{\text{train}}}) > \tau\right) \leq \frac{\gamma}{\tau}.$$

In particular, as γ decreases, so does the probability that a random test subject lies within an ϵ-ball of a random training subject. The same argument applies to object embeddings a.

Since embedding neighborhoods are **independent** for the subject and the object, the joint probability that a given training triple implants the correct test edge is bounded by

$$\begin{aligned}&\Pr\left(s_{\text{test}} \in \mathcal{V}_{s_{\text{train}}} \ \wedge \ a_{\text{test}} \in \mathcal{V}_{a_{\text{train}}}\right) \\ &= \Pr\left(\cos(e_{s_{\text{test}}}, e_{s_{\text{train}}}) > \tau\right) \\ &\quad \times \Pr\left(\cos(e_{a_{\text{test}}}, e_{a_{\text{train}}}) > \tau\right) \\ &\leq \left(\frac{\gamma}{\tau}\right)^2.\end{aligned}$$

Thus, each training triple contributes to the test-set edge-coverage only with probability at most $\left(\frac{\gamma}{\tau}\right)^2$. Then the factuality gap scales at most like

$$\left(\frac{\gamma}{\tau}\right)^2 \cdot D_{\text{kn}} - \left(\frac{\gamma}{\tau}\right)^2 \cdot D_{\text{unk}} = \left(\frac{\gamma}{\tau}\right)^2 \cdot \left(D_{\text{kn}} - D_{\text{unk}}\right).$$

In particular, as the average test–train cosine similarity γ decreases, the factor $\left(\frac{\gamma}{\tau}\right)^2$ becomes smaller, thereby reducing the factuality gap proportionally. □

A.5 Proof of Theorem 5.1

In this work, the term ICL prompt refers to two specific types of prompts: few-shot prompts and CoT prompts. In the following, we consider these two types separately to provide the theoretical analysis.

Proof. Let $\Pi = \{(s_i', r, a_i')\}$ be the few-shot prompt that provided to the LLM together with the input pair (s, r). This prompt can be interpreted as an auxiliary knowledge graph $\mathcal{G}_\Pi = (\mathcal{V}_\Pi, \mathcal{E}_{r_\Pi}, \mathcal{E}_\Pi^{\text{sim}})$. The graph includes not only the triples (s, r, a) from the demonstrations, but also the edges connecting them to semantically similar nodes that are implicitly related within the same domain.

With graph $\mathcal{G}_\Pi$, the updated knowledge graph becomes

$$\mathcal{G}^\star = \mathcal{G}_{\text{unk/kn}} \cup \mathcal{G}_\Pi.$$

Since $\mathcal{G}_\Pi, \mathcal{G}_{\text{unk}}, \mathcal{G}_{\text{kn}} \subseteq \mathcal{G}_r$, with a knowledge prompt that has enough semantic connection with the in-distribution data, there exists a sufficiently large subgraph $\mathcal{G}_\mathcal{P}$ such that

$$|\mathcal{E}_\Pi \cap \mathcal{E}_{\text{kn}}| > |\mathcal{E}_\Pi \cap \mathcal{E}_{\text{unk}}|.$$

Then, the factuality gap is

$$\begin{aligned}
\Delta_{\text{fact}}^\star &= \lambda \left(|\mathcal{E}_{\text{kn}} \cup \mathcal{E}_\mathcal{P}| - |\mathcal{E}_{\text{unk}} \cup \mathcal{E}_\mathcal{P}|\right) \\
&= \lambda \left[(|\mathcal{E}_{\text{kn}}| + |\mathcal{E}_\mathcal{P}| - |\mathcal{E}_{\text{kn}} \cap \mathcal{E}_\mathcal{P}|)\right. \\
&\quad \left. - (|\mathcal{E}_{\text{unk}}| + |\mathcal{E}_\mathcal{P}| - |\mathcal{E}_{\text{unk}} \cap \mathcal{E}_\mathcal{P}|)\right] \\
&= \lambda(|\mathcal{E}_{\text{kn}}| - |\mathcal{E}_{\text{unk}}| \\
&\quad - (|\mathcal{E}_{\text{kn}} \cap \mathcal{E}_\mathcal{P}| - |\mathcal{E}_{\text{unk}} \cap \mathcal{E}_\mathcal{P}|)) \\
&< \lambda \left(|\mathcal{E}_{\text{kn}}| - |\mathcal{E}_{\text{unk}}|\right) \\
&= \Delta_{\text{fact}}.
\end{aligned}$$

According to Appendix A.3, there $\lambda = \frac{|\mathcal{D}_{\text{test}}|}{|\mathcal{V}|^2}$. Therefore, we can get

$$\Delta_{\text{fact}}^\star < \Delta_{\text{fact}}.$$

Let $C = (s, r_1, a_1, r_2, a_2, \ldots, r_k, a)$ be the CoT prompt that provided to the LLM together with the input pair (s, r). This prompt can be interpreted as an auxiliary knowledge graph $\mathcal{G}_C = (\mathcal{V}_C, \mathcal{E}_C)$. The graph consists of the complete set of nodes and edges that lie along the reasoning path from the subject s to the object a.

With graph $\mathcal{G}_\Pi$, the updated knowledge graph becomes

$$\mathcal{G}_{\text{unk/kn}}^\star = \mathcal{G}_{\text{unk/kn}} \cup \mathcal{G}_C.$$

The new factuality gap is defined as

$$\Delta_{\text{fact}}^\star = |\{\text{covered by } \mathcal{G}_{\text{kn}}^\star\}| - |\{\text{covered by } \mathcal{G}_{\text{unk}}^\star\}|.$$

But for every test triple (s, r, a) that is explained by the CoT prompt, it is covered by both augmented graphs. Therefore, its contribution to the gap is $1 - 1 = 0$. Any remaining gap can only come from test triples not supported by CoT.

In the extreme case where CoT covers the entire test set, we have:

$$\Delta_{\text{fact}}^\star = 0.$$

More generally, since the same CoT subgraph is added to both graphs, the only remaining difference in coverage comes from test triples outside the scope of CoT. Thus, we have:

$$\Delta_{\text{fact}}^\star \leq \Delta_{\text{fact}}.$$

□

B Experiment Details

B.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. For each relation, we take the smaller data size between the Known split and the Unknown split as the final data size, in order to ensure that the Known and Unknown splits have equal amounts of data under

each relation. 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, 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 an equal distribution of each type of question. We also use few-shot prompting to filter the Unknown and Known splits. The difference is that, considering the smaller size of the PopQA dataset, we randomly select only 3 few-shot groups from the evaluation set, while keeping the other filtering parameters consistent with those used for Entity Questions. Finally, the number of Known and Unknown samples used for each of the four models is as follows: LLaMA Base: 3,659; LLaMA Instruct: 3,589; Mistral Base: 3,488; and Mistral Instruct: 3,421. 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.

Training Details. We divide all the training into 12 groups based on the 3 datasets and 4 models, with each group containing training on the Unknown and Known subsets. We ensure that the training parameters are exactly the same within each group.

For all the 12 groups, 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 12 groups uses 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 and PopQA dataset, all models are trained for 20 epochs. For MMLU, the Llama models are trained for 15 epochs, while the Mistral models are trained for 30 epochs. All of the models are trained on an 8× RTX 6000 Ada Generation 48G setup.

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_new_token` value of 10.

B.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.

B.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* and *Unknown* dataset.

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 B.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.

B.4 OOD 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 se-

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

lect 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.5 Finetuned on Small Dataset

In this section, we introduce the experimental setup for evaluating the factuality gap resulting from fine-tuning on a small subset versus the whole dataset. The construction of the PopQA training and evaluation sets has already been described in Appendix B. We fine-tune both the LLaMA Base and Mistral Base models on two dataset settings: (1) the full PopQA training set, consisting of 11,409 samples and (2) a randomly selected 5% subset of the full data, consisting of 561 samples. The training hyper-parameters follow those specified in Appendix B for the corresponding PopQA experiments. The only difference is that here we train for 10 epochs and select the final model based on early stopping.

The evaluation settings remain the same as in previous experiments, including the reuse of the original ICL prompt design.

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 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 5 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 6, 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 6 and Table 7, 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.

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	32.61	29.46	27.64	26.77	28.87	28.45	29.01	28.20
	Known	35.97	34.71	32.68	31.18	31.39	31.42	30.06	29.92
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 5: 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	31.70	25.05	32.05	21.25
	Known	41.75	39.90	39.80	37.80	31.40	30.15	33.05	33.90
PopQA	Unknown	37.05	33.80	31.07	28.52	28.97	28.55	29.25	28.38
	Known	36.91	36.00	34.29	33.00	31.42	31.60	30.27	30.13
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 6: 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	39.82	37.89	35.06	34.00	35.93	35.76	31.46	31.32
	Known	38.77	38.66	35.55	36.18	35.93	35.90	31.63	31.84
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 7: Few-shot learning with CoT