Understanding Finetuning for Factual Knowledge Extraction

Gaurav Ghosal [1] **Tatsunori Hashimoto** [2] **Aditi Raghunathan** [1]

Abstract

In this work, we study the impact of QA fine-tuning data on downstream factuality. We show that fine-tuning on lesser-known facts that are poorly stored during pretraining yields significantly worse factuality than fine-tuning on well-known facts, even when all facts are seen during pretraining. We prove this phenomenon theoretically, showing that training on lesser-known facts can lead the model to ignore subject entity names and instead output a generic plausible response *even when the relevant factual knowledge is encoded in the model*. On three question answering benchmarks (PopQA, Entity Questions, and MMLU) and two language models (Llama-2-7B and Mistral-7B), we find that (i) finetuning on a completely factual but lesser-known subset of the data deteriorates downstream factuality (5-10%) and (ii) finetuning on a subset of better-known examples matches or outperforms finetuning on the entire dataset. Ultimately, our results shed light on the interaction between pretrained knowledge and finetuning data and demonstrate the importance of taking into account *how* facts are stored in the pretrained model when fine-tuning for knowledge-intensive tasks.

1. Introduction

Large language models store large amounts of factual knowledge in their weights during pretraining (Jiang et al., 2020; Petroni et al., 2019; Mallen et al., 2023). As a result, they have shown promise on a variety of knowledge intensive tasks, including factual question-answering (Roberts et al., 2020; Radford et al., 2019). However, these abilities are unreliable and language models are prone to generate plausible, but incorrect responses to queries (Huang et al., 2023).

A natural avenue to improve factuality is via fine-tuning, as studied in several recent works (Kazemi et al., 2023; Joshi et al., 2023; Ouyang et al., 2022; Tian et al., 2023a; Yang et al., 2023). Multiple works, however, have shown that language models answer questions incorrectly even when they know the right answer, suggesting that current approaches to fine-tuning may be suboptimal (Burns et al., 2022; Li et al., 2023a; Liu et al., 2023b). In order to achieve better fine-tuning or uncover the ceiling of such approaches, we need to understand what factors determine the performance of fine-tuning. What is the mechanism by which fine-tuning improves factuality?

We can distill prior understanding of this question into three factors. Joshi et al. (2023) posits that fine-tuning on truthful data influences the model to adopt a credible *persona*. This theory suggests that ensuring the *factual accuracy* of the finetuning data is sufficient for downstream factuality. Another view from Kazemi et al. (2023) and Allen-Zhu & Li (2023) is that fine-tuning familiarizes the pretrained model with the QA format, which varies from the way that facts are observed during pretraining. This implies that finetuning examples should cover question formats likely to be seen during testing. Finally, Schulman (2023) and Yang et al. (2023) hypothesize that fine-tuning examples must be drawn from facts that the model sees during pretraining.

In this work, we find that the impact of fine-tuning examples depends on *how well* they are stored in the model, beyond simply their factuality or whether they are grounded in the pretraining corpus. Concretely, fine-tuning on QA examples about facts that the pretrained model knows well significantly improves factuality. Conversely, fine-tuning on QA examples regarding less well-encoded facts *actively harms* downstream factuality, causing the model to incorrectly respond to questions it could otherwise get right. We make this finding in a synthetic setting, after ensuring that all QA examples are factually accurate, representative of the downstream task, and seen during pretraining.

Why does the encoding of facts seen in finetuning affect factuality downstream? We propose the following intuitive mechanism. When presented with a factual question, a model can either respond using relevant memorized knowledge or leverage more general "shortcuts" that enable it to propose a plausible, but incorrect response. For example,

[1]Department of Machine Learning, Carnegie Mellon University, Pittsburgh, USA [2]Department of Computer Science, Stanford University, Stanford, USA. Correspondence to: Gaurav Ghosal <gghosal@andrew.cmu.edu>.

Proceedings of the 41^{st} International Conference on Machine Learning, Vienna, Austria. PMLR 235, 2024.

when asked about a person's occupation, a language model could potentially take the shortcut of responding with a word that is generally associated with occupations (i.e. actor). If shortcut usage is reinforced during fine-tuning, this can drown out the influence of memorized knowledge, causing the model to behave less factually on test data. Our observations suggest that the composition of the fine-tuning data controls which mechanism is amplified: less well-known facts can lead to more aggressive use of shortcuts. We conceptually illustrate our hypothesis in Figure 1.

In Section 4, we prove this intuition in a one-layer transformer. We introduce a quantity termed *factual salience* that measures how well a fact is learned by the one-layer transformer. Next, we demonstrate that a one-layer transformer can resort to using shortcuts through *attention imbalance*: attending only to more general tokens (for example those that specify the question type) rather than the specific entities in the question. We prove that *fine-tuning gradients on less salient facts contribute to the formation of attention imbalance*, while those on more salient facts counteract it. Furthermore, we show the effect of attention imbalance is amplified when looking at downstream performance on less well-known facts. Our results have a counterintuitive consequence: for less well-known facts, it is *worse* to fine-tune on similar less well-known facts and better to fine-tune on a different distribution of more well-known facts.

We test the implications of our analysis on three real-world QA datasets (PopQA, MMLU, and Entity Questions) and two LLM models (Llama-2-7B and Mistral-7B). As predicted by our theory, we find that fine-tuning on well-known knowledge (top 50%) outperforms fine-tuning on less well-known knowledge (bottom 50%) by 7% on MMLU, 6% on PopQA, and 4% on EntityQuestions. Moreover, we can match the performance of fine-tuning on the entire dataset by finetuning on just the top 50%. On MMLU, we find that finetuning on the top 30% well-known facts *outperforms* finetuning on the entire dataset by up to 2.5%.

To summarize, via theory and experiments, we uncover an important factor that determines the effect of finetuning on downstream factuality—how well the finetuned facts are encoded in the pretrained model. Beyond a conceptual understanding, our findings have immediate practical considerations for finetuning data curation: it can suffice to focus on a smaller number of well-known facts even when trying to improve factuality on less well-known facts.

2. Preliminaries and Setup

Language models are presented with large quantities of factual knowledge during pretraining, for example in books and articles in the pretraining corpora (Jiang et al., 2020; Petroni et al., 2019; Mallen et al., 2023). When users interact with

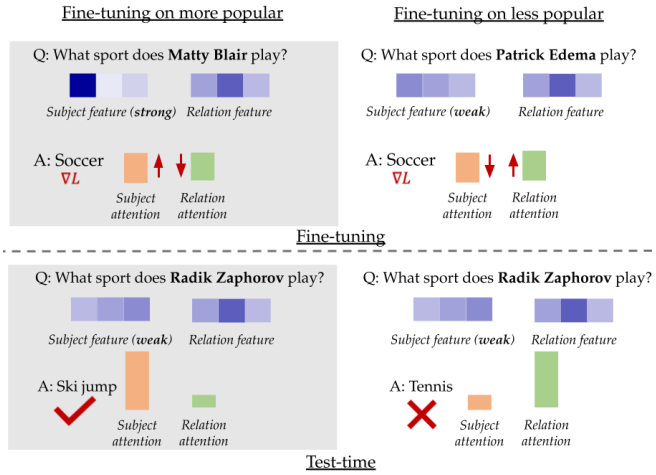


Figure 1. **Conceptual Mechanism of Finetuning on Popular versus Unpopular Knowledge**. When finetuning on less popular knowledge, the model can learn to heavily upweight relation features which enables it to make a plausible guess about the correct answer. However, training on popular, well-encoded facts discourages this imbalance. At testing time, heavy reliance on relation features can result in less popular knowledge being overwritten.

language models, however, it is most desirable for them to extract knowledge in a QA format, which varies from how facts are seen in pretraining. As a result, LLMs must undergo finetuning to learn to apply their pretrained knowledge to these downstream QA tasks. Here, we introduce a formalization of this process which guides our synthetic experiments (Section 3) and theory (Section 4).

Definition of Factual Knowledge Following prior works (Petroni et al., 2019; Elsahar et al., 2018), we abstractly represent a fact as the mapping from a subject-entity s and relation-type r to an answer a. We can represent these mappings as *knowledge triples* (s, r, a) where $s \in \mathcal{S}, r \in \mathcal{R}$, and $a \in \mathcal{A}$ and $\mathcal{S}$, $\mathcal{R}$, $\mathcal{A}$ are the set of all subject entities, relations, and answers, respectively. Importantly a single (s, r, a) triple can be expressed in multiple ways in natural language. Here, we model a natural language as the set of sequences of tokens lying in a token set $\mathcal{T}$.

Knowledge Formatting Functions Previously, we presented a definition of factual knowledge but observed that a fact can be presented textually in many formats. We formalize this intuition by introducing the notion of a *formatting function* $g : \mathcal{S} \times \mathcal{R} \times \mathcal{A} \to \mathcal{T}^k$ which maps an (s, r, a) triple to a series of tokens lying in the set $\mathcal{T}$. One such g, for example, could map the knowledge triple (USA, capital, Washington D.C.) to the tokenization of the sentence "The capital of the USA is Washington D.C."

Pretraining and Fine-tuning Now, we are ready to formalize the interaction of the pretraining and finetuning stages. Given a set of knowledge triples $D_{\text{k}} = \{(s, r, a)_{i=1}^{N}\}$ and a

pretraining formatting function, we generate a pretraining corpus $D_{\text{pre}} = \{g_{\text{pre}}(s,r,a) | (s,r,a) \in D_{\text{k}}\}$. Next, for a *downstream formatting function* g_{down}, we generate a downstream dataset $D_{\text{down}} = \{g_{\text{down}}(s,r,a) | (s,r,a) \in D_{\text{k}}\}$. In practice, the finetuning dataset is often limited relative to pretraining so we partition D_{down} into D_{ft} and D_{eval} and use D_{ft} for finetuning and D_{eval} as a held-out test set.

In QA settings, g_{pre} presents facts as they would be seen in books and articles, while g_{down} presents facts as question-answer pairs (i.e. "What is the capital of the USA? Washington D.C."). The goal of QA finetuning is thus to enable facts observed in the pretraining format to be extracted by prompting in question-answering (QA) format.

3. Synthetic Experiments

In this section, we study the role of fine-tuning data on factuality in a synthetic setup. This setup allows us to investigate the role of the pretraining process, which would be impractical to do in real large language models.

3.1. Synthetic Setup

We consider the following simulated setup based on the formalism introduced in Section 2. We consider that there is a single token for each subject, relation, and answer. We take $g_{\text{pre}}(s,r,a) = (s,r,a)$ (i.e. the pretraining formatting function simply maps to the sequence of subject, relation, and answer tokens). To simulate the change in formatting that occurs in downstream tasks, we introduce a *QA-prompt* token p_r for each relation type. The QA-prompt tokens are unseen during pretraining but used in the downstream formatting function: $g_{\text{down}}(s,r,a) = (s,p_r,a)$. Thus, during finetuning, the language model must learn to respond to a prompt (s,p_r) as if it had been prompted with (s,r). Our token space is thus $\mathcal{T} = \mathcal{S} \cup \mathcal{R} \cup \mathcal{A} \cup \{p_r | r \in \mathcal{R}\}$.

During pretraining, (s,r,a) triples are sampled i.i.d. from the distribution $s \sim \text{Zipf}(\mathcal{S}), r \sim \text{Unif}(\mathcal{R})$ at each step. This modeling choice simulates the fact that pretraining corpora often contain both very popular entities as well as many obscure, rarely seen ones. During fine-tuning, however, we perform standard batch based training on D_{ft}. We assume that all knowledge sequences presented to the model (in both pretraining and downstream formats) are consistent with the ground truth (s,r,a) triples in D_{k}. This allows us to study the role of finetuning data *beyond factual correctness* as is the focus of prior work (Joshi et al., 2023).

Finally, we emphasize that all facts in the downstream finetuning (D_{ft}) and test datasets (D_{eval}) are present in D_{pre}. As a result, our simulation results do not arise from the impact of finetuning on new knowledge as has been hypothesized in prior works (Schulman, 2023).

3.2. Observations in Simulation

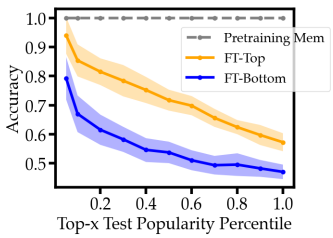
Main Finding: Fine-tuning Fact Popularity Impacts Downstream Performance In Figure 2(a), we plot the accuracy of training on the most popular (`FT-Top`) and least popular (`FT-Bottom`) entities in the finetuning dataset. We find that the choice of finetuning dataset *significantly* impacts downstream QA factuality. Concretely, fine-tuning on examples corresponding to the most popular facts in pretraining results in a 10% improvement in factuality. This difference is amplified as we include relatively less popular data in the test set. For example, the difference between `FT-Top` and `FT-Bottom` doubles when we extend our test set from the top 5% to the top 10% most popular entities and persists as we include increasingly unpopular entities.

Impact of Long-Tailedness in Pretraining Corpus In Figure 2(b), we examine the impact of the Zipf α parameter on this phenomena. Intuitively, as α increases, the difference in frequency between more and less popular facts increases. On the other hand, lower α values result in a more uniform distribution over facts. We observe that increasing α exacerbates the differences between the fine-tuning datasets, whereas lowering α largely closes the gap. These findings suggest that differing impacts of the fine-tuning datasets is correlated with how significantly facts differ in their pretraining frequency.

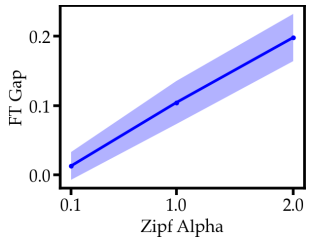
Impact of the Number of Pretraining Steps Previously, we observed that the *long-tailedness* of the pretraining distribution controls sensitivity to the fine-tuning dataset. One hypothesis to explain this could be that less frequent facts might not be stored in the model, but we observe that the gap between more and less popular facts is present even when all facts can be extracted in (s,r) form, as evidenced by the pretraining memorization accuracy in Figure 2(a). This suggests that the gap is driven primarily by differences in the internal *fact-storage*. In Figure 2(c), we investigate this by plotting the gap between `FT-Top` and `FT-Bottom` as a function of pretraining steps. We find that with more pretraining steps, the gap decreases, indicating that these internal differences in fact storage disappear as all facts are seen a sufficient number of times. However, achieving this in real large language models would likely be impractical due to the large scale of pretraining data, and the regime of practical interest shows vast difference in downstream performance depending on the finetuning distribution.

3.3. Conceptual Model: Factual Salience

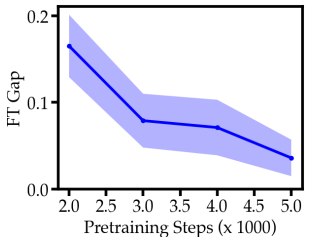
Our findings in simulation suggest a *continuous progression* in whether a model "knows" a particular fact. This controls how well a fact can be extracted in downstream settings, as seen in the decline of downstream accuracy with popularity in Figure 2(a). Moreover, the extent to which the model



(a) Impact of Finetuning Dataset



(b) Effect of Zipf Alpha



(c) Effect of Pretraining Steps

Figure 2. **Simulation Study of Finetuning for Knowledge Extraction** (a) We plot the downstream factuality of finetuning on more versus less popular facts, finding that finetuning on more popular facts improves downstream factuality (b) We plot the difference between finetuning on `FT-Top` and `FT-Bottom` as a function of the subject Zipf parameter. We find that on increasingly *long-tailed* datasets, the impact of finetuning dataset is amplified. (c) We plot the difference between finetuning on `FT-Top` and `FT-Bottom` as a function of pretraining steps, finding that the difference between the finetuning datasets is mitigated with additional training.

knows a fact also determines its behavior in finetuning, as evidenced by the gap between `FT-Top` and `FT-Bottom` in Figure 2(a). We refer to this intuitive quantity of how well a model knows a fact as the *fact salience* and provide a formal analysis in Section 4.

Our simulated results indicate that fact salience is related to the frequency of facts in the pretraining corpus. In particular, we see that differences in the salience are exacerbated as the pretraining distribution becomes more *long-tailed*. However, we also find that these differences are mitigated with additional pretraining, suggesting that they are driven primarily by facts that have been seen only a few times. Importantly, this matches the regime of typical language model training, where roughly *single-epoch* training is performed over a diverse and long-tailed pretraining corpus. In this setting, many facts are likely to be seen only a few times, since multiple passes are not performed.

4. Theoretical Analysis of Factual Salience

In the previous section, we intuitively introduced *fact salience* and hypothesized that it plays a central role in factuality. We now formalize this intuition in a one-layer transformer. We give a quantitative definition of fact salience in this simplified setting (Section 4.2) and justify its relationship to downstream fact extractability (Theorem 4.2). Next, we demonstrate that fine-tuning on less salient facts can suppress pretrained knowledge (Theorem 4.5). Finally, we prove that the factual salience increases as a fact is seen during pretraining, justifying the use of pretraining frequency as a proxy (Section 4.4). In Appendix C.1, we validate our theory numerically.

Simplified Model We analyze a one-layer, single-headed transformer model with fixed, orthogonal token embeddings (denoted as $\phi(t)$ for token $t \in \mathcal{T}$). Our model has two learnable parameters: the value matrix, W_{V}, and the key-query matrix W_{KQ} and we assume that $W_{\text{V}}, W_{\text{KQ}} \in \mathbb{R}^{|\mathcal{T}| \times |\mathcal{T}|}$ (i.e. the value, key and query projections preserve the dimensions of the embeddings). We fix the output head to be the identity operation. The forward pass of our model on the input sequence (s, r) can thus be decomposed as follows:

$$X = \begin{bmatrix} \phi(s) & \phi(r) \end{bmatrix}$$

In the first step, the token sequence is embedded and concatenated to an input embedding matrix X.

$$\text{Self-Att}(X; W_{\text{V}}, W_{\text{KQ}}) = W_{\text{V}} X \sigma(X W_{\text{KQ}} X_{-1})$$

Next, the input embedding matrix passes through a single head of self-attention to compute the last-token activation.

$$f(s, r; W_{\text{V}}, W_{\text{KQ}}) = \sigma(\text{Self-Att}(X; W_{\text{V}}, W_{\text{KQ}}))$$

Finally, we compute a probability distribution over the *next token* $f(s, r; W_{\text{V}}, W_{\text{KQ}})$ as a softmax over the last output of the self-attention layer. An extended analysis of this model is provided in Appendix A.1.

Remark. Since W_{V} is full rank, the parameterization above is sufficiently expressive to achieve 100% argmax decoding accuracy (as described in Appendix A.1) on any pretraining dataset where every (s, r) has a unique answer (see Appendix A.2 for proof).

4.1. Quantifying Factual Salience

In Section 3, we hypothesized that facts are stored with different strengths in the model weights after pretraining, impacting both their extractability and their behavior in finetuning. In this section, we explicitly quantify this strength in a one-layer transformer.

Definition 4.1 (Fact Salience)**.** For a fact (s, r, a), we define the fact salience $S(s, r, a)$ as $\phi(a)^\top W_{\text{V}} \phi(s)$.

Since we fix the model's output transformation to be the identity, $W_{\text{V}}\phi(s)$ can be viewed as encoding an unnormalized probability distribution over the next token, *conditioned only on* s. Thus, $S(s, r, a)$ measures how well the

model "stores" the correct answer in relation to s. Intuitively, for the fact to be extractable downstream (when prompted by (s, p_r)), the model can only rely on information stored in s because p_r is unseen during pretraining. Additionally, we observe that $S(s, r, a)$ does not depend on the attention parameters as all memorization is implemented by W_{V} (as demonstrated in Appendix A.2). In the next section, we demonstrate the role played by W_{KQ} in modulating the contribution of this stored knowledge to the model's output.

4.2. Attention Imbalance

In the previous section, we defined a measure of how well knowledge is internally stored in a one-layer transformer. In this section, we study the role of the attention mechanism in *controlling* how this stored knowledge contributes to the model's output. In particular, we show that *imbalances* in the attention scores of s and p_r can suppress pretrained knowledge.

Theorem 4.2 (Attention imbalance can lead to hidden knowledge)**.** *For pretraining data D_{pre}, where all a appear at least once, suppose there exists a value matrix W_V satisfying mild assumptions A.2 to A.4. Then the one-layer transformer $f(s, p_r; W_V, 0)$ achieves 100% accuracy under $\arg\max$ decoding, but there exists W_{KQ} s.t. $f(s, p_r; W_V, W_{KQ})$ does not achieve 100%.*

We provide the specific construction in Appendix A.3 and discuss the relevant assumptions. To summarize, for each relation r, we can ensure that a subset of facts with that relation is predicted incorrectly by sending the attention weight on the subject token towards 0 (equivalently, increasing the attention on the prompting token towards 1). However, *not all facts are equally susceptible to being suppressed in this way* as we highlight below:

Fact Salience Controls Robustness to Attention Imbalance Our proof of Theorem 4.2 relies on ensuring that the attention to the subject token when prompting with (s, p_r) is sufficiently low. In Appendix A.3, we demonstrate that an incorrect prediction occurs when the attention to the subject token $\text{Att}_s \leq \frac{d}{S(s,r,a)}$, for a constant d. This formalizes our intuition that fact salience determines how robustly a fact is stored: a small attention imbalance can only force an incorrect response on facts that are less salient.

Next, we make a connection to the phenomena of hidden knowledge, where a LLM outputs an incorrect response despite the correct response being extractable through other probing methods.

Remark: Hidden Knowledge As Theorem 4.2 does not allow any modification of the value matrix, all factual associations are still stored in W_{V} and could potentially be extracted by examination of the model's internal parameters. As such, our theory agrees with empirical findings where factually correct knowledge can be extracted from model representations, despite an incorrect generation.

Ultimately, we observe that even when factual knowledge is stored in model parameters, it can be suppressed from the output by attention imbalance. In Section 4.3, we study the fine-tuning process and demonstrate how attention imbalance can arise.

4.3. Fine-tuning Attention Dynamics

In Section 4.2, we showed that imbalances in attention can harm factuality by suppressing stored knowledge. Here, we prove that the *facts seen in finetuning* play an important role in controlling this imbalance. Concretely, finetuning on low-salience facts can exacerbate attention imbalance, while the inclusion of high-salience facts can counteract it. We begin by defining two quantities that appear in the W_{KQ} gradient during finetuning (i.e. updating on (s, p_r, a) triples).

Definition 4.3 (Subject Token Relevance)**.** $s_{\text{rel}} = (\phi(a) - f(s, p_r; W_{\text{V}}, W_{\text{KQ}}))^\top (W_{\text{V}}\phi(s))$

and correspondingly the relation token relevance:

Definition 4.4 (Relation Token Relevance)**.** $p_{\text{rel}} = (\phi(a) - f(r, p_r; W_{\text{V}}, W_{\text{KQ}}))^\top W_{\text{V}}\phi(p_r)$.

As derived in the appendix, the update to the attention matrix takes the form

$$-\frac{\partial L}{\partial W_{\text{KQ}}} \propto (s_{\text{rel}} - p_{\text{rel}})(\phi(s)\phi(p_r)^\top - \phi(p_r)\phi(p_r)^\top).$$

The term $\phi(s)\phi(p_r)^\top$ increases the attention on s, while the term $\phi(p_r)\phi(p_r)^\top$ increases attention on p_r. Thus, the W_{KQ} gradient up-weights attention on the most relevant token (as measured by s_{rel} and p_{rel}).

Theorem 4.5 (Factuality vs. Nonfactuality Inducing Gradients)**.** *When finetuning on a fact (s, p_r), if $s_{rel} - p_{rel} < 0$ then the attention update $-\frac{\partial L}{\partial W_{KQ}}$ decreases the attention on all s' when prompting with (s', p_r), whereas when $s_{rel} - p_{rel} > 0$, $-\frac{\partial L}{\partial W_{KQ}}$ increases the attention on all s' when prompting with (s', p_r).*

We postpone the formal proof to Appendix A.5 but provide the following key observations.

Role of Factual Salience: Observe that the definition of subject token relevance (Def. 4.3) includes the previously defined fact salience. Intuitively, gradient steps taken on less salient facts (relative to the token's correlation with the final output $f(s, r; W_{\text{V}}, W_{\text{KQ}})$) downweight attention on the s token (where pretraining knowledge is stored) and push the transformer globally towards attention imbalance.

Global Effect of p_r Attention Updates: The term $\phi(p_r)^\top W_{\text{KQ}}\phi(p_r)$ appears in the forward pass of *all* facts with relation r. Therefore, updates on a fact where $s_{\text{rel}} - p_{\text{rel}} < 0$ implicitly decrease attention on all $s \in \mathcal{S}$ (by increasing the attention score on p_r). When training on many such (s, p_r), these updates can accumulate and contribute to significant attention imbalance. Conversely, when $s_{\text{rel}} - p_{\text{rel}} > 0$ the attention on p_r will be decreased, implicitly up-weighting the subject attention for all $s \in \mathcal{S}$. Importantly, this is not a specific consequence of the (s, p_r) ordering examined in this work: it holds whenever the final prompt token is not subject-specific.

4.4. Fact Popularity and Salience

While our analysis so far has relied on how strongly facts are internally stored by the model ($S(s, r, a)$), it is unclear how to compute this quantity beyond the simplified one-layer transformer setting. Here, we verify that the number of times a fact is seen during pretraining correlates with its salience, as suggested by our results in Section 3. This suggests pretraining popularity as a proxy for salience.

Theorem 4.6 (Lower bound on fact salience)**.** *Consider pretraining $f(s, r; W_V, W_{KQ})$ on a dataset D_{pre} of size N for one epoch with learning rate ϵ. Suppose that the $\|W_{KQ}\|_\infty < C_{KQ}$ and $\|W_V\|_\infty < C_V$ throughout training. Suppose that the combination (s, r) appears n times and s appears no more than $n\frac{\exp(-C_{KQ})(|\mathcal{T}|-1)}{2\exp(C_V)}$ times. Then $S(s, r, a) \geq nc_1\epsilon$ where $c_1 > 0$.*

We postpone the proof to Appendix A.4.

Ultimately, our examination of the one-layer transformer provides a tight-fitting conceptual explanation of our simulated observations in Section 3. We quantify how strongly a fact is stored in the pretrained weights (fact salience) and demonstrate it grows with pretraining frequency (Theorem 4.6). Our analysis illustrates that fact salience plays a central role in determining (a) the suppression of pretrained knowledge (Theorem 4.5) and (b) how robustly a fact can be extracted at test time (Theorem 4.2). We verify this intuition in large language models in Section 5.

5. Large Language Model Experiments

In this section, we verify our findings on the role of the QA dataset when finetuning pretrained large language models (Llama 7B and Mistral-7B). Unlike Section 3, where we prescribed an idealized model of the pretraining distribution, the settings examined here test our theory with models trained on large-scale pretraining corpora.

Table 1. Construction of PopQA-Controlled

Question	Ans	Pop.
FT-Top		
In what country is Chrysler?	USA	55586
What sport does Matty Blair play?	Soccer	50643
What is Vanessa Angel's occupation?	Actor	157667
FT-Bottom		
In what country is Robinson?	USA	142
What sport does Patrick Edema play?	Soccer	46
What is Edwin Wallock's occupation?	Actor	68

Table 2. Results on PopQA-Controlled

Method	Test-Acc
Zero-Shot Prompting	20.1%
FT-Top	44.5 %
FT-Bottom	37.4%

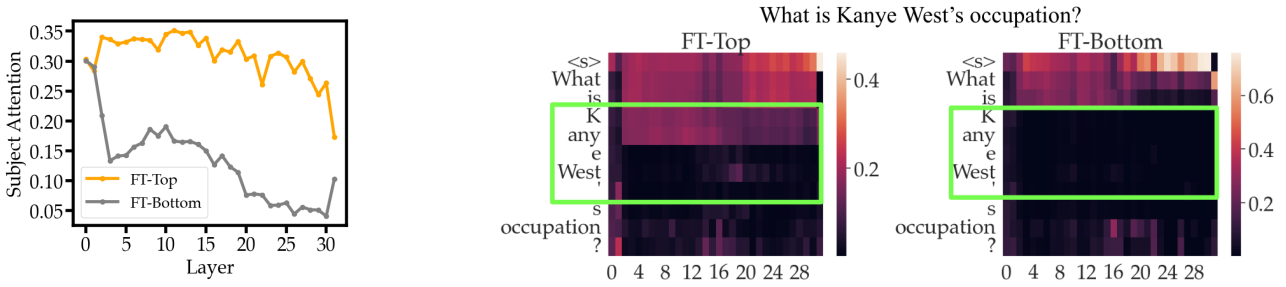
5.1. Controlled Experiment

We first perform a controlled experiment to test the impact of fact salience without confounders.

Controlled Setup To isolate the effect of fact salience on downstream performance, we construct two fine-tuning datasets that differ in fact salience but have the same make-up of relation types and corresponding answers. We use a subset of the PopQA dataset (Mallen et al., 2023) consisting of the **country**, **sport** and **occupation** relations, which we refer to as PopQA-Controlled. We take all questions from each relation with the *most frequent answer* (respectively USA, Soccer, and Actor) and divide them into more and less popular halves (respectively `FT-Top` and `FT-Bottom`). Examples from the two fine-tuning datasets are shown in Table 1. We disambiguate whether the fine-tuned models learn to use their pretrained knowledge or shortcuts in fine-tuning by testing on questions whose answers are not one of the three seen in fine-tuning. Our theory predicts that fine-tuning on more salient facts would encourage the model to use pretrained knowledge, resulting in better test performance.

Results In Table 2, we observe a significant decrease (7%) in the factual accuracy of models finetuned on `FT-Bottom` versus `FT-Top`. Our results establish that both (a) the varying impact of finetuning on popular versus unpopular knowledge occurs in language models and (b) this effect cannot be explained by correlations between popularity and answer entities or relation types.

Stratified Analysis In Figure 4, we additionally observe a surprising trend: the gap between `FT-Top` and



(a) Average Subject Attention Score

(b) Attention Pattern on FT-Top versus FT-Bottom

Figure 3. **Analysis of Llama-7B Attention Patterns** (a) We plot the maximum attention score over subject tokens for Llama-7B models finetuned on `FT-Top` and `FT-Bottom` across layers, where the maximum attention score is averaged over the heads in each layer. All results are averaged over examples in the PopQA-Controlled test set. (b) We compare the attention patterns for a specific question between the `FT-Top` and `FT-Bottom` fine-tuned models. The tokens corresponding to the subject are enclosed within the green rectangle.

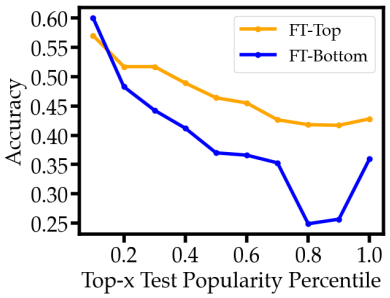


Figure 4. **PopQA-Controlled Test Accuracy on Popularity Percentiles** We plot the accuracy on the top x popularity percentiles of the PopQA-Controlled test set as a function of x. We compare the performance of finetuning on `FT-Top` versus `FT-Bottom`. We observe that while both finetuning datasets perform comparably on the most popular facts in the test set, training on the less popular data significantly underperforms on relatively less popular test questions.

`FT-Bottom` *increases* as we consider increasingly less popular test set examples. While both finetuning datasets yield similar results in the most-popular 20% of the test set, the gap between the methods widens as we include increasingly unpopular questions, dropping sharply around the 70th popularity percentile. This finding evinces that our observations *are not* merely a result of matching the finetuning distribution to the test set in popularity (where we would expect large gains for `FT-Top` on more popular knowledge). Counter-intuitively, skewing the finetuning dataset to more popular examples appears to be especially beneficial in improving performance on less popular knowledge.

Analysis of Attention Patterns We study the attention patterns of models fine-tuned on `FT-Top` versus `FT-Bottom` and find that they match the theoretical predictions made in Section 4. In the left panel of Figure 3 we plot the average attention to the subject tokens (over the test set) as a function of Llama-7B layer index and find that `FT-Top` trained models attend significantly more to the subject than do models fine-tuned on `FT-Bottom`. On the right panel, we visualize the attention patterns of the `FT-Top` and `FT-Bottom` trained models and see that the attention to the subject-relevant tokens (highlighted in green) is suppressed after training on `FT-Bottom`. In this setting, these results provide evidence that the mechanistic prediction made in our one-layer transformer model in Section 4 is reflective of what occurs in a real large language model. Further experimental results are presented in Appendix A.5.

5.2. Real QA Datasets

Previously, we demonstrated the impact of the fine-tuning QA dataset on question-answering ability in a controlled setting. In this section, we test the implications of our findings for improving factual QA performance.

5.2.1. Setup

Datasets We specialize our evaluation to short answer and multiple choice QA involving facts of varying popularity (frequency in the pretraining data). Mallen et al. (2023) introduce the PopQA dataset which is sampled to include questions about a range of popular and less-popular topics. We also examine a subset of the Entity Questions (Sciavolino et al., 2022) dataset, which includes a diverse range of popular and less popular facts. In both datasets, we utilize the Wikipedia page count of the question *subject-entity* as a proxy for pretraining fact frequency (Mallen et al., 2023; Razeghi et al., 2022). This is necessary as it is challenging to directly measure fact popularity on large-scale pretraining corpora. Finally, we examine a subset of the MMLU dataset (Hendrycks et al., 2021) consisting of history questions. Here, we use the pretrained model's loss as a proxy for fact popularity as introduced by Kang et al. (2024).

Table 3. MMLU-History

Finetuning Dataset	Llama-7B	Mistral-7B
FT-Top	**61.4**% (0.3)	**68.7**% (0.5)
FT-Bottom	55.6 % (0.4)	59.4% (0.5)
FT-Whole	58.8% (0.2)	67.4 %(0.4)

Models Our experiments are performed on the Llama 7B (Touvron et al., 2023) and Mistral-7B (Jiang et al., 2023) pretrained base language models. Restricting to base (non-chat-tuned) models allows us to directly study the effect of pretraining knowledge frequency without confounding introduced by prior finetuning stages. In all experiments, we use the best of LoRA (Hu et al., 2021) and full fine-tuning. In addition, we tune over standard regularization techniques including weight decay, early stopping, and learning rate individually for each model and fine-tuning dataset (further details in Appendix B.2).

Evaluation We evaluate the performance of models on short-answer QA by appropriately normalizing all model predictions and ground-truth answers and checking for exact string matches. We describe the specific normalization techniques in Appendix B.2. We note that both short answer datasets used in this work provide multiple synonymous ground-truth answers, mitigating the potential harshness of exact-match-based evaluations. For multiple-choice evaluation (MMLU), we evaluate the exact match of the model with the ground-truth answer choice.

5.2.2. Results

Unpopular Facts Harm Downstream Factuality In Figure 5, we observe that finetuning on the least popular knowledge consistently under-performs across both QA datasets (PopQA and Entity-Questions) and models (Llama-7B and Mistral). Similar results are also seen in Table 3 on the MMLU dataset, where finetuning on less confident examples performs significantly (7-10%) worse than both the top and whole datasets. The consistency of this observation across models and tasks supports that our observations are a general property of the finetuning process, rather than an artifact of a particular LLM or dataset.

Impact Relative to Test Popularity Figure 5 displays similar trends relative to test-set popularity as those seen in the more restricted settings. In particular, we observe that the gap between fine-tuning on more versus less popular examples widens away from the most popular 10% of the test points as we include more unpopular points in our test set. The advantage of training on more popular examples persists even when we include the least popular test points. This finding provides further evidence of our hypothesis that although some highly popular facts are relatively invariant to the choice of fine-tuning dataset, performance on relatively less popular facts varies more significantly.

Popular Facts Mitigate Unpopular Surprisingly, we find that even a *randomly selected* 50% subset (plotted sky-blue in Figure 5) significantly outperforms `FT-Bottom`, performing only slightly worse than `FT-Top` across all settings. This suggests that some popular points (which would be present in `FT-Random` but not `FT-Bottom`) can significantly mitigate the damage incurred by finetuning on less popular knowledge. Moreover, this conclusion is supported by our theoretical analysis: the gradients on more popular examples *globally counteract* attention imbalance, as shown in Theorem 4.5.

Finetuning Data Quantity in Question-Answering In Figure 5, we compare the performance of the *best top popularity subset* with fine-tuning on the entire training dataset. Across all settings, we observe that training on a smaller subset of the most popular facts performs comparably or better than using the entire dataset. Moreover, these variations are amplified on the same percentiles as the difference between `FT-Top` and `FT-Bottom` (i.e. between the 30th-60th popularity percentiles). In Table 3, we similarly observe that training only on the most familiar MMLU examples performs better than using the whole dataset across both models. This suggests that (a) *only a subset* of the most popular training points are actually helpful in fine-tuning for factual question-answering and that (b) including the additional QA examples could be harmful to facts that are especially sensitive to finetuning distribution.

6. Related Works

Impact of Unfamiliar Knowledge Recent works have examined the impact of unfamiliar examples during finetuning. Kang et al. (2024) argues that unfamiliar examples in the fine-tuning dataset determine how a model responds to unfamiliar test examples. However, they do not consider the impact of unfamiliar fine-tuning examples on the general factuality of the model as is the focus of this work. Concurrently to this work, Gekhman et al. (2024) demonstrate empirically that finetuning on unfamiliar examples can worsen factuality, characterizing it as a result of overfitting. In this paper, however, we present a conceptual model of this phenomena, demonstrating that it arises from suppression of pretrained knowledge in favor of generic "shortcuts". Our theory additionally explains the varying impact that different fine-tuning strategies have on *test points* of varying popularity or familiarity.

Reliable Factuality of Language Models Prior works have extensively studied challenges and approaches for improving the factuality of LLMs. Mallen et al. (2023) and Kandpal et al. (2023) demonstrate that LLMs often underperform

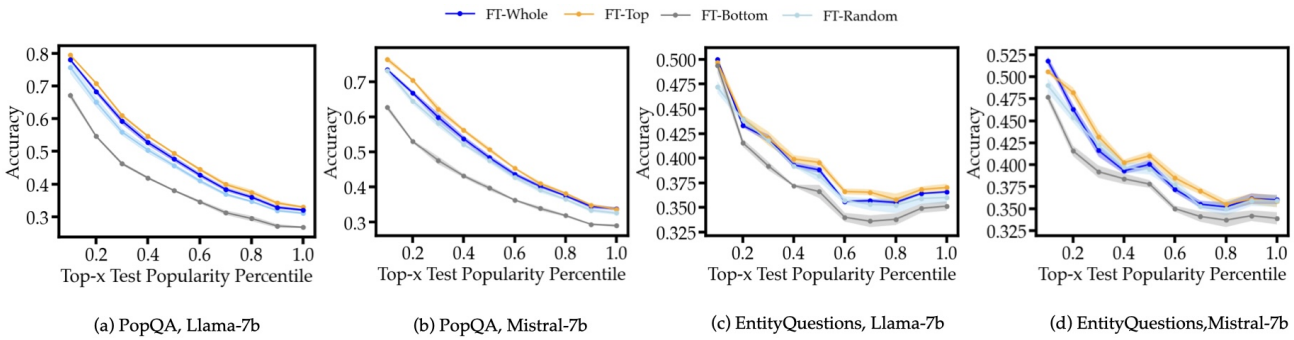


Figure 5. **Finetuning Performance on Real Datasets** We plot the factual QA accuracy across two models and question-answering datasets under different fine-tuning strategies. `FT-Top` denotes finetuning on the most popular half of data, `FT-Whole` denotes finetuning on the whole training dataset, `FT-Random` denotes finetuning on a randomly selected half of the data, and `FT-Bottom` denotes finetuning on the lower 50% of the data, sorted by popularity. We plot performance restricting to the top-x popularity percentiles of the test set.

on obscure or long-tailed knowledge. Li et al. (2023a) find that the factuality of language models can be improved by upweighting certain attention heads. Similarly, Burns et al. (2022) demonstrate that unsupervised internal probes can reveal factual associations in language models, even when they ultimately output an incorrect response. Chuang et al. (2023) demonstrate that contrasting the final prediction from earlier layers of language models can improve factual accuracy. Prior works have also examined methods to improve factuality via abstention.Varshney et al. (2023) demonstrate that low confidence outputs can be hallucinations. Similarly, Yuksekgonul et al. (2023) use token attention scores to detect when language models hallucinate. On the other hand, Yang et al. (2023); Zhang et al. (2023); Schulman (2023) introduce fine-tuning techniques to induce large-language models to refuse questions that are outside their knowledge boundary. Collectively, these prior works demonstrate failure modes of factual reliability in language models, at times even when they can output the correct answer. In this work, on the other hand, we study the impact of the fine-tuning distribution on the model's downstream factuality.

Understanding LLM Mechanisms and Training Dynamics Many prior works have sought to explain the behaviors of language models and understand their failure modes. Allen-Zhu & Li (2023) examine the conditions on pretraining data necessary for facts to be stored in an extractable form on a synthetic dataset. Geva et al. (2023) identify the mechanisms by which facts are stored and extracted in language models. Li et al. (2023b) study one-layer transformer pretraining dynamics on a topic modeling task. Chen et al. (2024) empirically studies the pretraining dynamics of syntax acquisition in masked language models. Tian et al. (2023b) analyze the attention dynamics of one-layer transformers, demonstrating that uniquely co-occurring tokens are upweighted in attention. Liu et al. (2023a) examine long-range reasoning failures of large language models, attributing them to erroneous attention scores. In this work, we focus on understanding the mechanics of fine-tuning relating to promoting or suppressing pretrained knowledge, thereby impacting the extractability of facts downstream.

7. Discussion

In this work, we investigate the impact of QA dataset composition on model factuality, making a notable finding: fine-tuning on questions about well-known facts uniformly improves factuality over fine-tuning on less known facts. We observe this trend across a range of simulation and real-world settings and develop a conceptual model of QA fine-tuning in a simplified one-layer transformer. Our results challenge intuitive heuristics for designing QA fine-tuning datasets. In particular, over-representing well-known facts in QA fine-tuning can actually be beneficial. Our results can inform principled methods to improve the downstream factuality of language models. Guided by our theory, a valuable area for future work can be developing regularization techniques to mitigate attention imbalance during finetuning. Another promising avenue is curriculum learning, which could enable more obscure facts to be trained on *after* finetuning on more popular knowledge to mitigate attention imbalance. Finally, we hypothesize that our conceptual model can guide the development of synthetic data to efficiently improve knowledge extractability.

Acknowledgements

This research was supported by the Center for AI Safety Compute Cluster. Any opinions, findings, conclusions, or recommendations expressed in this material are those of the author(s) and do not necessarily reflect the views of the sponsors. This work was supported in part by the AI2050 program at Schmidt Sciences (Grant #G2264481). We gratefully acknowledge the support of Apple. TH acknowledges the support of Open Philanthropy.

Impact Statement

This paper presents work whose goal is to advance the field of Machine Learning. There are many potential societal consequences of our work, none of which we feel must be specifically highlighted here.

References

Allen-Zhu, Z. and Li, Y. Physics of language models: Part 3.1, knowledge storage and extraction, 2023.

Burns, C., Ye, H., Klein, D., and Steinhardt, J. Discovering latent knowledge in language models without supervision, 2022.

Chen, A., Shwartz-Ziv, R., Cho, K., Leavitt, M. L., and Saphra, N. Sudden drops in the loss: Syntax acquisition, phase transitions, and simplicity bias in MLMs. In *The Twelfth International Conference on Learning Representations*, 2024. URL https://openreview.net/forum?id=MO5PiKHELW.

Chuang, Y.-S., Xie, Y., Luo, H., Kim, Y., Glass, J., and He, P. Dola: Decoding by contrasting layers improves factuality in large language models, 2023.

Elsahar, H., Vougiouklis, P., Remaci, A., Gravier, C., Hare, J., Laforest, F., and Simperl, E. T-rex: A large scale alignment of natural language with knowledge base triples. In *Proceedings of the Eleventh International Conference on Language Resources and Evaluation (LREC 2018)*, 2018.

Gekhman, Z., Yona, G., Aharoni, R., Eyal, M., Feder, A., Reichart, R., and Herzig, J. Does fine-tuning llms on new knowledge encourage hallucinations?, 2024.

Geva, M., Bastings, J., Filippova, K., and Globerson, A. Dissecting recall of factual associations in auto-regressive language models, 2023.

Hendrycks, D., Burns, C., Basart, S., Zou, A., Mazeika, M., Song, D., and Steinhardt, J. Measuring massive multitask language understanding, 2021.

Hu, E. J., Shen, Y., Wallis, P., Allen-Zhu, Z., Li, Y., Wang, S., Wang, L., and Chen, W. Lora: Low-rank adaptation of large language models, 2021.

Huang, L., Yu, W., Ma, W., Zhong, W., Feng, Z., Wang, H., Chen, Q., Peng, W., Feng, X., Qin, B., and Liu, T. A survey on hallucination in large language models: Principles, taxonomy, challenges, and open questions, 2023.

Jiang, A. Q., Sablayrolles, A., Mensch, A., Bamford, C., Chaplot, D. S., de las Casas, D., Bressand, F., Lengyel, G., Lample, G., Saulnier, L., Lavaud, L. R., Lachaux, M.-A., Stock, P., Scao, T. L., Lavril, T., Wang, T., Lacroix, T., and Sayed, W. E. Mistral 7b, 2023.

Jiang, Z., Xu, F. F., Araki, J., and Neubig, G. How can we know what language models know?, 2020.

Joshi, N., Rando, J., Saparov, A., Kim, N., and He, H. Personas as a way to model truthfulness in language models, 2023.

Kandpal, N., Deng, H., Roberts, A., Wallace, E., and Raffel, C. Large language models struggle to learn long-tail knowledge, 2023.

Kang, K., Wallace, E., Tomlin, C., Kumar, A., and Levine, S. Unfamiliar finetuning examples control how language models hallucinate, 2024.

Kazemi, M., Mittal, S., and Ramachandran, D. Understanding finetuning for factual knowledge extraction from language models, 2023.

Li, K., Patel, O., Viégas, F., Pfister, H., and Wattenberg, M. Inference-time intervention: Eliciting truthful answers from a language model, 2023a.

Li, Y., Li, Y., and Risteski, A. How do transformers learn topic structure: Towards a mechanistic understanding, 2023b.

Liu, B., Ash, J. T., Goel, S., Krishnamurthy, A., and Zhang, C. Exposing attention glitches with flip-flop language modeling, 2023a.

Liu, K., Casper, S., Hadfield-Menell, D., and Andreas, J. Cognitive dissonance: Why do language model outputs disagree with internal representations of truthfulness?, 2023b.

Mallen, A., Asai, A., Zhong, V., Das, R., Khashabi, D., and Hajishirzi, H. When not to trust language models: Investigating effectiveness of parametric and non-parametric memories, 2023.

Ouyang, L., Wu, J., Jiang, X., Almeida, D., Wainwright, C. L., Mishkin, P., Zhang, C., Agarwal, S., Slama, K., Ray, A., Schulman, J., Hilton, J., Kelton, F., Miller, L., Simens, M., Askell, A., Welinder, P., Christiano, P., Leike, J., and Lowe, R. Training language models to follow instructions with human feedback, 2022.

Petroni, F., Rocktäschel, T., Lewis, P., Bakhtin, A., Wu, Y., Miller, A. H., and Riedel, S. Language models as knowledge bases? *arXiv preprint arXiv:1909.01066*, 2019.

Radford, A., Wu, J., Child, R., Luan, D., Amodei, D., Sutskever, I., et al. Language models are unsupervised multitask learners. *OpenAI blog*, 1(8):9, 2019.

Razeghi, Y., au2, R. L. L. I., Gardner, M., and Singh, S. Impact of pretraining term frequencies on few-shot reasoning, 2022.

Roberts, A., Raffel, C., and Shazeer, N. How much knowledge can you pack into the parameters of a language model? In Webber, B., Cohn, T., He, Y., and Liu, Y. (eds.), *Proceedings of the 2020 Conference on Empirical Methods in Natural Language Processing (EMNLP)*, pp. 5418–5426, Online, November 2020. Association for Computational Linguistics. doi: 10.18653/v1/2020.emnlp-main.437. URL https://aclanthology.org/2020.emnlp-main.437.

Schulman, J. Reinforcement learning from human feedback: Progress and challenges. Talk given at the University of California, Berkeley on April 19, 2023., 2023. URL https://www.youtube.com/watch?v=hhiLw5Q_UFg.

Sciavolino, C., Zhong, Z., Lee, J., and Chen, D. Simple entity-centric questions challenge dense retrievers, 2022.

Tian, K., Mitchell, E., Yao, H., Manning, C. D., and Finn, C. Fine-tuning language models for factuality, 2023a.

Tian, Y., Wang, Y., Chen, B., and Du, S. Scan and snap: Understanding training dynamics and token composition in 1-layer transformer, 2023b.

Touvron, H., Lavril, T., Izacard, G., Martinet, X., Lachaux, M.-A., Lacroix, T., Rozière, B., Goyal, N., Hambro, E., Azhar, F., Rodriguez, A., Joulin, A., Grave, E., and Lample, G. Llama: Open and efficient foundation language models, 2023.

Varshney, N., Yao, W., Zhang, H., Chen, J., and Yu, D. A stitch in time saves nine: Detecting and mitigating hallucinations of llms by validating low-confidence generation, 2023.

Yang, Y., Chern, E., Qiu, X., Neubig, G., and Liu, P. Alignment for honesty, 2023.

Yuksekgonul, M., Chandrasekaran, V., Jones, E., Gunasekar, S., Naik, R., Palangi, H., Kamar, E., and Nushi, B. Attention satisfies: A constraint-satisfaction lens on factual errors of language models, 2023.

Zhang, H., Diao, S., Lin, Y., Fung, Y. R., Lian, Q., Wang, X., Chen, Y., Ji, H., and Zhang, T. R-tuning: Teaching large language models to refuse unknown questions, 2023.

A. Theory Appendix

A.1. Notations and Setup

Representation of Tokens We consider a synthetic language with a token set $\mathcal{T}$ where $|\mathcal{T}|$. When representing tokens, we consider them to be integers in the interval $[0, |\mathcal{T}|]$. To represent factual associations, we further partition $\mathcal{T} = \mathcal{S} \cup \mathcal{R} \cup \mathcal{A} \cup \{p_r | r \in \mathcal{R}\}$. As such $\mathcal{S}$, $\mathcal{R}$, $\mathcal{A}$, and $\{p_r | r \in \mathcal{R}\}$ are sets of integers, representing the underlying tokens.

Embedding Layer As introduced in Section 4, we consider fully fixed, fully orthogonal token embeddings (i.e. the embedding matrix is the identity matrix) and the embedding of a token i is $e_i \in \mathbb{R}^{|\mathcal{T}|}$ (i.e. a vector with all entries 0 except for the i-th component). Moreover, the embedding and unembedding modules are considered to be weight-tied as examined in Li et al. (2023b). In this setting, we have that the embedding of a token i is the i-th basis vector (i.e. e_i), and as a result the embeddings of the different tokens are orthogonal to one another. In addition, the i-th component of the model output $f(s, r, W_{\text{V}}, W_{\text{KQ}})_i$ is the probability of token i being the next token (as we discuss further when introducing $\arg\max$ decoding).

One-Layer Transformer Architecture We consider a one-headed, one-layer transformer in this work with fully orthogonal and weight-tied embedding and unembedding layers. We assume that the key, query, and value matrices are square, thereby preserving the dimensions of the embedding. We additionally assume that the language modeling head corresponds to an identity transformation (this is possible due to the projections preserving the dimensions of the embedding).

Denote a matrix of embedded inputs $X \in \mathbb{R}^{|\mathcal{T}| \times l}$, where l is the sequence length. We can then write the output of this single head of attention (given parameters W_K, W_Q, W_V) as :

$$\text{Self-Att}(X; W_Q, W_K, W_{\text{V}}) = (W_{\text{V}} X)\sigma((W^K X)^T (W^Q X))$$

where σ denotes the column-wise softmax operation.

In our simplified model, we consider only one self-attention layer and consider that the language modeling head is the identity, which is possible because the embeddings, query, key, and value projections all lie in $\mathbb{R}^{|\mathcal{T}|}$. We can then write the *next-token prediction* function, given a sequence of tokens $t_1, ..., t_l$, as $f : \mathcal{T}^l \to \Delta(\mathcal{T})$ where l is the sequence length and $\Delta(\mathcal{T})$ is the space of probability distributions over the token space $\mathcal{T}$. Applying our simplifying assumptions, we can write that

$$f([t_1, , , .t_l], W^Q, W^K, W^V) = \sigma(\text{Self-Att}(X; W^Q, W^K, W_{\text{V}})_{:-1})$$

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. Note that this can be rewritten:

$$f([t_1, , , .t_l], W^Q, W^K, W^V) = \sigma((W^V X)\sigma((W^K X)^T W^Q X_{:-1})).$$

.

Note that the actual computation depends only on the product $(W^K)^\top W^Q$ and thus, we will reparameterize as $W_{\text{KQ}} = (W^K)^\top W^Q$. For convenience, in the main text, we redefine the Self-Att function to map from an input sequence embedding matrix to the last token's embedding (i.e. Self-Att : $\mathbb{R}^{|\mathcal{T}| \times l} \to \mathbb{R}^{|\mathcal{T}|}$) and we parametrize with only W^V and W^{KQ}. Thus, we use the definition

$$\text{Self-Att}(X; W_{\text{V}}, W_{\text{KQ}}) = (W^V X)\sigma(X^T W^{KQ} X_{:-1}).$$

For most of our analysis, we will focus on the specialized setting of next-token prediction given the context (s, r) or (s, p_r). In this specialized setting (considering (s, r) for instance), we have that $X = \begin{bmatrix}\phi(s) & \phi(r)\end{bmatrix}$. We can write the following

$$f(s, r; W_{\text{V}}, W_{\text{KQ}}) = \sigma(W^V \begin{bmatrix}\phi(s) & \phi(r)\end{bmatrix} \sigma(\begin{bmatrix}\phi(s)^\top \\ \phi(r)^\top\end{bmatrix} W_{\text{KQ}} \phi(r)).$$

Finally, we will rewrite this in order to more clearly demonstrate the contribution of different tokens to the final prediction.

$$\sigma(\begin{bmatrix}\phi(s)^\top\\ \phi(r)^\top\end{bmatrix} W_{\text{KQ}}\phi(r))_0 W_{\text{V}}\phi(s) + \sigma(\begin{bmatrix}\phi(s)^\top\\ \phi(r)^\top\end{bmatrix} W_{\text{KQ}}\phi(r))_1 W^V\phi(r).$$

In the following proofs, we will often abbreviate $\text{Att}_s = \sigma(\begin{bmatrix}\phi(s)^\top\\ \phi(r)^\top\end{bmatrix} W_{\text{KQ}}\phi(r))_0$ and $\text{Att}_r = \sigma(\begin{bmatrix}\phi(s)^\top\\ \phi(r)^\top\end{bmatrix} W_{\text{KQ}}\phi(r))_1$. Thus, the pre-softmax output of the one-layer transformer can be written as

$$\text{Att}_s W_{\text{V}}\phi(s) + \text{Att}_r W^V\phi(r).$$

$\arg\max$ **Decoding** Recall that the language model embedding layer is is fixed to be an identity matrix. In addition, the output projection is defined to be an identity transformation. Thus, the final token output of the self-attention layer can be interpreted as an un-normalized probability distribution over the token space $\mathcal{T}$. We then define $\arg\max$-decoding as:

$$\arg\max_{t\in\mathcal{T}}(f(s,r;W_{\text{V}},W_{\text{KQ}}))_t.$$

That is, the index of the maximum value in the one-layer transformer output corresponds to the index of the next token. Additionally, due to the softmax layer preserving the ordering of the vector's components, this is equivalent to:

$$\arg\max_{t\in\mathcal{T}}(\text{Self-Att}(X;W_{\text{V}},W_{\text{KQ}}))_t.$$

A.2. One-Layer Transformer Can Memorize All Facts

In this section, we prove that despite its simplified nature, a one-layer transformer is capable of memorizing all facts in the pretraining dataset. This helps establish that our simplified model does not restrict the ability to learn facts in our setting.

Theorem A.1 (One-layer transformer can fully memorize the pretraining dataset)**.** *Consider any pretraining dataset $D_{pre} = \{(s^{(1)},r^{(1)},a^{(1)}),...,(s^{(N)},r^{(N)},a^{(N)}\}$ such that any (s,r) combination appears exactly once. There exists a one-layer transformer with parameters $f(s,r;W_V,W_{KQ})$ s.t. $\arg\max f(s,r;W_V,W_{KQ}) = a\ \forall(s,r,a)\in D_{pre}$.*

Proof. We will prove that the choice of parameters: $W_{\text{V}} = \sum_{(s,r,a)\in D_{\text{pre}}} \phi(a)\phi(s)^\top + \phi(a)\phi(r)^\top$ and $W_{\text{KQ}} = 0$ satisfies the result of the theorem.

Observe that $f(s,r;W_{\text{V}},0) = \sigma(\frac{1}{2}W_{\text{V}}\phi(s^*) + \frac{1}{2}W_{\text{V}}\phi(r^*))$. Next, we will define two sets $\mathcal{A}^{s^*}$ and $\mathcal{A}^{r^*}$

$$\mathcal{A}^{s^*} = \{a|(s^*,r,a)\in D_{\text{pre}}\,\forall r\in\mathcal{R}\}$$
$$\mathcal{A}^{r^*} = \{a|(s,r^*,a)\in D_{\text{pre}}\,\forall s\in\mathcal{S}\}$$

Intuitively, $\mathcal{A}^{s^*}$ denotes the set of all answers that are observed associated with s^* (as r is allowed to vary), and likewise $\mathcal{A}^{s^*}$ is the set of answers seen with r^* as s is allowed to vary. Using these sets and applying the orthogonality of the embeddings, we can simplify the expression for f to

$$f(s,r;W_{\text{V}},W_{\text{KQ}}) = \sigma(\frac{1}{2}\sum_{a\in A^{s*}}\mathbb{1}\{a\} + \frac{1}{2}\sum_{a\in\mathcal{A}^{r*}}\mathbb{1}\{a\}).$$

We can further simplify by pulling out the term corresponding to $a\in\mathcal{A}^{s^*}\cap\mathcal{A}^{r*}$. We will abbreviate $\mathcal{A}^{\cap} = \mathcal{A}^{s^*}\cap\mathcal{A}^{r*}$, $\mathcal{A}^{r^*}_{\setminus} = \mathcal{A}^{r^*}\setminus\mathcal{A}^{\cap}$, and $\mathcal{A}^{s^*}_{\setminus} = \mathcal{A}^{s^*}\setminus\mathcal{A}^{\cap}$

$$f(s,r;W_{\text{V}},W_{\text{KQ}}) = \sigma(\sum_{a\in\mathcal{A}^{\cap}}\phi(a) + \frac{1}{2}(\sum_{a\in\mathcal{A}^{s*}_{\setminus}}\phi(a) + \sum_{a\in\mathcal{A}^{r*}_{\setminus}}\phi(a))).$$

For the remainder of the proof, we will examine the *pre-softmax* output of the one-layer transformer, which we will write as

$$Z = \sum_{a\in\mathcal{A}^{\cap}}\phi(a) + \frac{1}{2}(\sum_{a\in\mathcal{A}^{s*}_{\setminus}}\phi(a) + \sum_{a\in\mathcal{A}^{r*}_{\setminus}}\phi(a)).$$

Observe that each (s, r) tuple is associated with only one a in the pretraining dataset D_{pre}, implying that $\mathcal{A}^r \cap \mathcal{A}^s = \{a^*\}$ where we define a^* such that $(s^*, r^*, a^*) \in D_{\text{pre}}$. This implies that (using orthogonality) that:

$$\phi(a^*)^\top Z = 1.$$

On the other hand, we have

$$\phi(a)^\top Z = \frac{1}{2} \quad \forall a \in \mathcal{A}_{\setminus}^{r^*} \cup \mathcal{A}_{\setminus}^{r^*}.$$

Finally, by orthogonality, we have for all $a \in \mathcal{A} \setminus (\mathcal{A}^{s^*} \cup \mathcal{A}^{r^*} \cup \mathcal{A}^{\cap})$

$$\phi(a)^\top (W_{\text{V}}\phi(s) + W_{\text{V}}\phi(r)) = 0.$$

Therefore $\text{argmax}_{\tilde{a}} \phi(\tilde{a})^\top (\frac{1}{2}\sum_{a \in A^s} \phi(a) + \frac{1}{2}\sum_{a \in A^r} \phi(a)) = a^*$ and we have completed the proof since this holds for arbitrary (s, r). □

Intuitively, the full-rank nature of the embedding and value matrices enable us to use the value matrix as a key-value store which encodes the mapping $(s, r) \to a$.

A.3. Proof of Theorem A.5

In this theorem, we demonstrate that despite potentially having all factual associations encoded in the value matrix W^V, the attention weights can be modified such that information is suppressed from the output layer. Our construction here depends primarily on the attention scores becoming imbalanced against the subject token, which we will demonstrate occurs during fine-tuning in the next section. We introduce three additional assumptions regarding the structure of the value matrix and the fact distribution. These assumptions are largely minor in nature: Assumption A.2 simply requires that the activations $\phi(a)^\top W^V \phi(s)$ are not all identical. In a similar vein, we assume that each answer is seen at least once in the dataset. Finally, as indicated in the hypothesis of the theorem, we assume that all the facts are memorized. Note that Assumption A.4 is a significantly weaker consequence of the one-layer transformer achieving 100% accuracy – however it is all that is needed for the proof.

Assumption A.2 (Non-Uniform Relation Marginal)**.** $\forall r \max_{a \in \mathcal{A}^r} \phi(a)^\top W_{\text{V}} \phi(p_r) - \min_{a \in \mathcal{A}^r} \phi(a)^\top W_{\text{V}} \phi(p_r) > 0$

Assumption A.3 (Answer Diversity)**.** $\forall r \ \forall a \in A^r \ \exists s \in \mathcal{S}$ such that$(s, r, a) \in D_{\text{pre}}$

Assumption A.4 (All Facts Memorized)**.** $\forall s \in \mathcal{S} \ \forall r \in \mathcal{R} \ \phi(a)^\top W^V \phi(s) \geq \max_{a' \in \mathcal{A}^r} \phi(a')^\top W_{\text{V}} \phi(r) - \phi(a)^\top W^V \phi(r)$

For simplicity, we will additionally assume that all entries of the value matrix W^V are greater than or equal to 0. This can be achieved by simply shifting all entries in the matrix, without changing the relative orderings of the activations.

Theorem A.5 (Attention imbalance can lead to hidden information)**.** *Consider any value matrix W_V satisfying assumptions A.2 throug A.4. Then a one-layer transformer with parameters $[W^V, 0]$ achieves 100% accuracy but there exists W^{QK} s.t. $f_{[W_V, W_{KQ}]}([s, r])$ does not achieve 100%.*

Proof. We will construct a W_{KQ} such that f does not achieve 100% accuracy. By Assumption A.3, $\forall r$ we have that there is at least one s such that $(s, r, a') \in D_{\text{pre}}$ where $a' \neq \arg\max_{a \in \mathcal{A}} \phi(a)^\top W_{\text{V}} \phi(r)$.

We will refer to $D_{min} = \{(s, r, a) \in D_{\text{pre}} \mid a \neq \arg\max_{a \in \mathcal{A}} \phi(a)^\top W_{\text{V}} \phi(r)\}$ (i.e. the set of (s, r, a) triples whose answers are not the most strongly encoded with respect to relation token r). We will show that we can construct W_{KQ}(without modifying W^V) such that all points in D_{min} are incorrectly responded to.

Consider $(s, r, a) \in D_{\text{min}}$. At balanced attention, we have that $\arg\max_{a' \in \mathcal{A}} \phi(a')^\top (W_{\text{V}}\phi(s) + W_{\text{V}}\phi(r)) = a$ because the one-layer transformer achieves 100% accuracy. However we also have that $\arg\max_{a' \in \mathcal{A}} \phi(a')^\top W_{\text{V}}\phi(r) \neq a$ by the construction of D_{min}. Denote the last-token attention scores given an attention matrix W^{QK} as $\begin{bmatrix}\text{Att}_s \\ \text{Att}_r\end{bmatrix} = \sigma(\begin{bmatrix}\phi(s)^\top W_{\text{KQ}}\phi(r) \\ \phi(r)^\top W_{\text{KQ}}\phi(r)\end{bmatrix})$. We have that if $\text{Att}_s(\phi(a)^\top W_{\text{V}}\phi(s)) \leq \text{Att}_r(\max_{a' \in \mathcal{A}} \phi(a')^\top W_{\text{V}}\phi(r) - \phi(a)^\top W_{\text{V}}\phi(r))$ then $f(s, r; W_{\text{V}}, W_{\text{KQ}}) \neq a$. We can see this by rearranging this inequality, yielding

$$\phi(a)^\top (\text{Att}_s W_{\text{V}}\phi(s) + \text{Att}_r W_{\text{V}}\phi(r)) \leq \text{Att}_r \phi(\tilde{a})^\top W_{\text{V}}\phi(r)) \leq \phi(\tilde{a})^\top (\text{Att}_s W_{\text{V}}\phi(s) + \text{Att}_r W_{\text{V}}\phi(r)).$$

where $\tilde{a} = \arg\max_{a \in \mathcal{A}} \phi(a)^\top W_{\text{V}} \phi(r)$. This implies that $f(s, r; W_{\text{V}}, W_{KQ}) \neq a$. We will term $\max_{a' \in \mathcal{A}^r} \phi(a')^\top W_{\text{V}} \phi(r) - \phi(a)^\top W_{\text{V}} \phi(r)$ as a relation specific constant d. Then we can achieve erasure of the fact (s, r) by ensuring

$$\sigma(\begin{bmatrix} \phi(s)^\top W_{\text{KQ}} \phi(r) \\ \phi(r)^\top W_{\text{KQ}} \phi(r) \end{bmatrix})_0 \leq \sigma(\begin{bmatrix} \phi(s)^\top W_{\text{KQ}} \phi(r) \\ \phi(r)^\top W_{\text{KQ}} \phi(r) \end{bmatrix})_1 \frac{d}{\phi(a)^\top W_{\text{V}} \phi(s)},$$

Since both the terms $\phi(s)^\top W_{\text{KQ}} \phi(r)$ and $\phi(r)^\top W_{\text{KQ}} \phi(r)$ are free variables, we will fix $\phi(r)^\top W_{\text{KQ}} \phi(r) = 0$ without loss of generality and compute the required constraint on the term $\phi(s)^\top W_{\text{KQ}} \phi(r)$. For convenience, we will use the abbreviation $c = \phi(s)^\top W_{\text{KQ}} \phi(r)$.

Substituting these simplifications, we have the following inequality

$$\frac{\exp\{c\}}{\exp\{c\} + 1} \leq \frac{1}{\exp\{c\} + 1} \frac{d}{\phi(a)^\top W_{\text{V}} \phi(s)}.$$

We have that setting $c = \phi(s)^\top W_{\text{KQ}} \phi(r) \leq \log \frac{d_r}{\phi(a)^\top W_{\text{V}} \phi(s)}$ achieves this. This confirms our intuition that as fact salience $\phi(a)^\top W_{\text{V}} \phi(s)$ becomes large, we must set the entry $\phi(s)^\top W_{\text{KQ}} \phi(r)$ increasingly negative to ensure that an incorrect answer is output. □

A.4. Results on Token Learning

In this section, we establish some theory relating to the representations of tokens after pretraining. First, we prove the following theorems regarding bounded softmax functions.

Theorem A.6 (Softmax on ℓ_∞ bounded vectors)**.** *Consider $x \in \mathbb{R}^d$ and suppose $\|x\|_\infty \leq C$. Then $\max_i(\sigma(x))_i \leq \frac{e^{2k}}{d-1}$ and $\min_i(\sigma(x))_i \geq \frac{e^{-2k}}{d}$*

Proof. $\sigma(x)_i = \frac{\exp(x_i)}{\sum_{j \in d} \exp(x_j)} \leq \frac{\exp(C)}{\exp(C) + (d-1)\exp(-C)} = \frac{\exp(2C)}{\exp(2C) + (d-1)} \leq \frac{\exp(2C)}{d-1}$. Likewise $\sigma(x)_i \geq \frac{\exp(-C)}{\exp(-C) + (d-1)\exp(C)} = \frac{\exp(-2C)}{\exp(-2C) + (d-1)} \geq \frac{\exp(-2C)}{d}$. □

Now we will prove a result on the activation of a token t, $W_{\text{V}} \phi(t)$ when trained by gradient descent with Cross Entropy loss with learning rate ϵ and updated N times. We will make the following assumptions:

Assumption A.7 (Attention Matrix Bounded)**.** $\forall t \in \mathcal{T}\ \|W_{\text{KQ}} \phi(t)\|_\infty \leq \frac{C_{KQ}}{2}$

Assumption A.8 (Value Matrix Bounded)**.** $\forall t \in \mathcal{T}\ \|W_{\text{V}} \phi(t)\|_\infty \leq \frac{C_V}{2}$

We require that Assumptions A.8 and A.7 hold throughout the training trajectory we consider. Now, consider a fixed token $t^* \in \mathcal{T}$ and a pretraining dataset D_{pre}. Let $D_t = \{(t^*, t_1, t_{a_1}), ..., (t^*, t_n, t_{a_n})\}$ be all examples in D_{pre} where t^* occurs. Furthermore consider that $t_{a_1}, ..., t_{a_n} \in \mathcal{T}^a$ and that $|\mathcal{T}^a| = k$, i.e. that $T^* = \{t_1^a, ..., t_k^a\}$. Finally let n_i denote the number of time t_i^a appears in D_{pre}. Now we are ready to state the theorem.

Theorem A.9 (Token Training Dynamics)**.** *Consider training a one-layer transformer with parameters $[W_V, W_{KQ}]$ starting at 0 initialization with batch size 1 and learning rate ϵ. Assume throughout training, we satisfy Assumptions A.7 and A.8. Then after one pass through D_{pre} we have that $\forall t_i^a \in \mathcal{T}^a\ \phi(t_{a_n})^\top (W^V \phi(t^*)) \geq (\frac{n_i \exp(-C_{KQ})}{2} - n \frac{\exp(C_V)}{|\mathcal{T}|-1})\epsilon$.*

Proof. We have that the single gradient step update for W_V on example (t^*, t_i, t_j^a) can be written:

$$W_V^{T+1} = W_V^T + (\text{Att}_{t^*}(\phi(t_j^a) - f(t^*, t_i; W_{\text{V}}, W_{\text{KQ}}))\phi(t^*)^\top + \text{Att}_{t_i}(\phi(t_j^a) - f(t^*, t_i; W_{\text{V}}, W_{\text{KQ}}))\phi(t_i)^\top)$$

where we denote the attention scores respectively on t^* and t_i as Att_{t^*} and Att_{t_i}. Since we are primarily focused on the value matrix projection of t^*, we will discard the t_i term in what follows (due to orthogonality).

We will first establish an upper bound on the pre-softmax output of the transformer, Z, in terms of the ℓ_∞ norm. By expanding and applying the triangle inequality we have

$$\|Z\|_\infty = \|\text{Att}_{t^*} W_{\text{V}} \phi(t^*) + \text{Att}_{t_i} W_{\text{V}} \phi(t_i)\|_\infty \leq \text{Att}_{t^*} \|W_{\text{V}} \phi(t^*)\|_\infty + \text{Att}_{t_i} \|W_{\text{V}} \phi(t_i)\|_\infty.$$

By Assumption A.8 we have that $\|W_{\text{V}}\phi(t^*)\|_\infty \leq \frac{C_V}{2}$ and $\left\|W^V\phi(t_i)\right\|_\infty \leq \frac{C_V}{2}$. This implies that

$$\|Z\|_\infty \leq \text{Att}_{t^*}\|W_{\text{V}}\phi(t^*)\|_\infty + \text{Att}_{t_i}\|W_{\text{V}}\phi(t_i)\|_\infty \leq \frac{C_V}{2}(\text{Att}_{t^*} + \text{Att}_{t_i}) = \frac{C_V}{2}$$

where the final equality comes from softmax property that $\text{Att}_{t^*} + \text{Att}_{t_i} = 1$

Next observe that $f(t^*, t_i; W_{\text{V}}, W_{\text{KQ}}) = \sigma(Z)$. We can then apply Theorem A.6 to upper and lower bound the components of $f(t^*, t_i; W_{\text{V}}, W_{\text{KQ}})$. In particular, we have that $\forall i \in [0, |\mathcal{T}|]$

$$\frac{\exp(-C_V)}{|\mathcal{T}|} \leq (f(t^*, t_i; W_{\text{V}}, W_{\text{KQ}}))_i \leq \frac{\exp(C_V)}{|\mathcal{T}| - 1}.$$

Now, we will examine the attention term, we have that

$$\begin{bmatrix}\text{Att}_{t^*}\\ \text{Att}_{t_i}\end{bmatrix} = \sigma(\begin{bmatrix}\phi(t^*)^\top W_{\text{KQ}}\phi(t_i)\\ \phi(t_i)^\top W_{\text{KQ}}\phi(t_i)\end{bmatrix}).$$

By Assumption A.7 we have that

$$\left\|\begin{bmatrix}\phi(t^*)^\top W_{\text{KQ}}\phi(t_i)\\ \phi(t_i)^\top W_{\text{KQ}}\phi(t_i)\end{bmatrix}\right\|_\infty \leq \frac{C_{KQ}}{2}$$

and thus, by Theorem A.6 we have that for $t \in \{t^*, t_i\}$ we have that

$$\frac{\exp(-C_{KQ})}{2} \leq \text{Att}_t \leq \exp(C_{KQ}).$$

Next for any $t_i^a \in \mathcal{T}^a$, the co-ordinate $\phi(t_i^a)^\top(W_{\text{V}}\phi(t^*))$ receives n_i updates of the form $+\epsilon\text{Att}_{t^*}$ and n updates of the form $-f(t^*, t'; W_{\text{V}}, W_{\text{KQ}})\text{Att}_s$. We can then lower bound the quantity $\phi(t_i^a)^\top(W_{\text{V}}\phi(t^*)) \geq n_i\epsilon\frac{\exp(-C_{QK})}{2} - n\frac{\exp(C_V)}{|\mathcal{T}|-1}\epsilon$ (where we have upper bounded the attention score in the second term by 1), thereby yielding the desired claim. □

Now, we are ready to prove the Theorem 4.6, as a straightforward application of Theorem A.9. We restate Theorem 4.6 below for convenience.

Theorem A.10 (Lower bound on fact salience)**.** *Consider pretraining $f(s, r; W_V, W_{KQ})$ on a dataset D_{pre} of size N for one epoch with learning rate ϵ. Suppose that the $\|W_{KQ}\|_\infty < C_{QK}$ and $\|W_V\|_\infty < C_V$ throughout training. Suppose that the combination (s, r) appears n times and s appears no more than s appears no more than $n_{tot} < n\frac{(|\mathcal{T}|-1)\exp(-C_{KQ})}{2\exp(C_V)}$ times. Then $(\phi(a)^\top)(W_V\phi(s)) \geq nc_1\epsilon$ where $c_1 > 0$.*

Proof. We can simply treat the set $\{a|(s, r, a) \in D_{pre}\}$ as $\mathcal{T}^a$ in the theorem above. Then we have that $n_i = n$ (for the purposes of Theorem A.9) and finally that $n < n_{tot}$. This implies that $\phi(a)^\top(W^V\phi(s)) \geq (n\frac{\exp(-C_{KQ})}{2} - n_{tot}\frac{\exp(C_V)}{|\mathcal{T}-1|})\epsilon$. We have that by the hypothesis of the theorem, $(n\frac{\exp(-C_{KQ})}{2} - n_{tot}\frac{\exp(C_V)}{|\mathcal{T}-1|}) > 0$ which gives the desired result

□

A.5. Results on Attention Dynamics

Now, we examine the process by which attention is learned in the one-layer transformer. Preliminarily, we have that the update corresponding to the attention matrix W_{KQ} on an abstract gradient update (s, r, a) is

$$-\frac{\partial L}{W_{KQ}} = c\underbrace{((\phi(a) - f(s, r; W_{\text{V}}, W_{\text{KQ}}))^\top(W_{\text{V}}\phi(r))}_{\text{correlation of r with update}}\underbrace{(\phi(r)\phi(r)^\top - \phi(s)\phi(r)^\top)}_{\text{increase attention on r}} +$$
$$\underbrace{(\phi(a) - f(s, r; W_{\text{V}}, W_{\text{KQ}}))^\top(W_{\text{V}}\phi(s))}_{\text{correlation of s with update}}\underbrace{(\phi(s)\phi(r)^\top - \phi(r)\phi(r)^\top)}_{\text{increase attention on s}}$$

Table 4. Large Language Model Hyperparameters

Hyperparameter	Range
Learning Rate	1e-5, 1e-4, 1e-3
Weight Decay	1e-6, 1e-5, 1e-4, 1e-3, 1e-2
(LoRA rank, LoRA α)	(8,16), (16,32), (32, 64), (64,128)
LoRA	True, False

where $c > 0$, arising from the gradient of the softmax term. For subsequent convenience, we will condense the update as

$$-\frac{\partial L}{W_{KQ}} = (\phi(a) - f(s, r; W_{\text{V}}, W_{\text{KQ}}))^\top (W_{\text{V}}\phi(r) - W_{\text{V}}\phi(s))(\phi(r)\phi(r)^\top - \phi(s)\phi(r)^\top).$$

On this basis, we now prove Theorem A.11, first restating it for convenience.

Theorem A.11 (Factuality vs. Nonfactuality Inducing Gradients)**.** *When finetuning on a fact (s, p_r), if $s_{rel} - p_{rel} < 0$ then the attention update $-\frac{\partial L}{\partial W_{KQ}}$ globally decreases the attention on all s' when prompting with (s', p_r), whereas when $s_{rel} - p_{rel} > 0$, $-\frac{\partial L}{\partial W_{KQ}}$ globally increases the attention on all s'.*

Proof. Beginning with the update of the query-key matrix, we observe that when $s_{rel} - p_{rel} < 0$, then we have (denoting the *post-update* value matrix as $W_{\text{KQ}}^{'}$).

$$\phi(r)^\top W_{\text{KQ}}^{'}\phi(r) = \phi(r)^\top W_{\text{KQ}}\phi(r) + t$$

where $t > 0$. This follows from the update rule:

$$W_{\text{KQ}}^{'} = W_{\text{KQ}} + t(\phi(r)\phi(r)^\top - \phi(s)\phi(r)^\top)$$

since the embeddings $\phi(r)$ are unit norm and we can omit the impact of the $\phi(s)\phi(r)^\top$ term by orthogonality.

Next, to show the global nature of the update, we consider an arbitrary subject token s' and compare the attention before and after the update on the prompt $[s', r]$. We will denote the pre-update subject token attention as $\text{Att}_{s'}^t$ and the post-update subject token attention as $\text{Att}_{s'}^{t+1}$. Then, we have

$$\text{Att}_{s'}^{t-1} = \frac{\exp\big(\phi(s')^\top W_{\text{KQ}}\phi(r)\big)}{\exp(\phi(s')^\top W_{\text{KQ}}\phi(r)) + \exp(\phi(p_r)^\top W_{\text{KQ}}\phi(p_r))}$$

and the attention on the subject after the update is

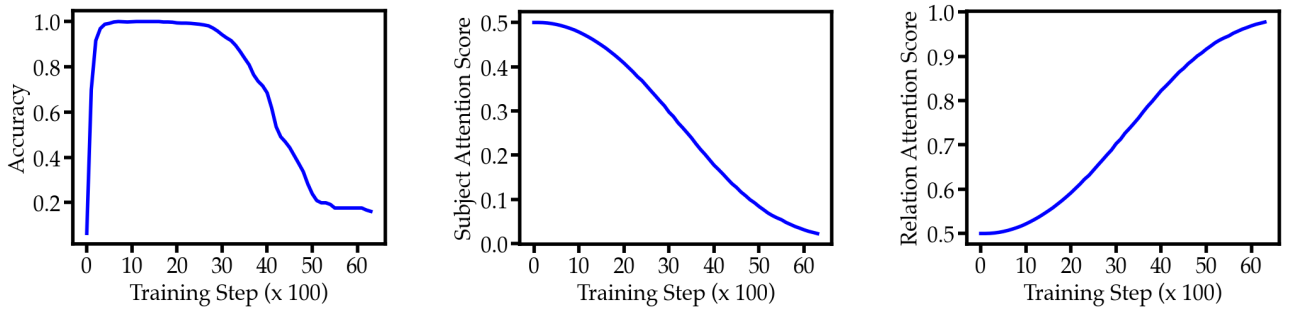
$$\text{Att}_{s'}^{t} = \frac{\exp\big(\phi(s')^\top W_{\text{KQ}}\phi(r)\big)}{\exp(\phi(s')^\top W_{\text{KQ}}\phi(r)) + \exp(\phi(p_r)^\top W_{\text{KQ}}\phi(p_r) + t)}$$

Thus, by the monotonicity of $\exp$ and the fact that $f(x) = x^{-1}$ is decreasing, we have that $\text{Att}_{s'}^{t-1} > \text{Att}_{s'}^{t}$. In the second case $s_{rel} - p_{rel} > 0$, the update to W_{KQ} is $-t\phi(r)\phi(r)^\top$ and we have the result by the same reasoning. □

B. Experimental Details

B.1. Hyperparameters and Tuning

Across all experiments, we tune the following hyper-parameters on the ranges shown in Table 4 on a held-out validation set. The LoRA entry refers to selecting whether LoRA or full-finetuning is used. In all experiments, we found that LoRA achieved better validation performance than full-finetuning. We report the performance after tuning on a held-out validation set in all experiments. Tuning is performed individually for each fine-tuning dataset.



(a) Accuracy (b) Subject Attention (c) Relation Attention

Figure 6. **Numerical Results on One-Layer Transformer** We consider the same pretraining data distribution as previously but with training a one-layer transformer rather than a multi-layer transformer. This represents an exact simulation of the setting of our theoretical model from Section 4. In (a) we plot the factual accuracy as a function of the finetuning step. In (b), we plot the attention score on the subject token, and in (c) the attention score on the relation token.

B.2. Evaluation

For short answer questions, we normalize the LLM output (generating up to 10 tokens), `llm_out` by the following:

```
llm_out_norm = llm_out.lower().rstrip().lstrip()
```

Given a list of possible ground-truth answers in the dataset `gt_list`, we compute whether the LLM response is correct as:

```
any([x in llm_out_norm for x in gt_list])
```

We consider the LLM's output on a multiple choice dataset as:

```
torch.argmax(out_scores[token_set])
```

where `out_scores` are the next token prediction scores on and `token_set` are the indices of the answer choice tokens (i.e. A, B, C, or D).

B.3. Attention Mechanism Analysis

We denote the subject attention as the maximum attention score over the tokens corresponding to the subject entity (relative to the final prompt token), as proposed in (Yuksekgonul et al., 2023). The attention score for each layer is computed by averaging over all attention heads in that layer. On the left panel, we show the average attention score to the subject entity averaged over the PopQA test set. We observe, past the initial few layers, that the attention to the subject significantly drops in the model fine-tuned on FT-Bottom, providing mechanistic evidence of our hypothesis in a large language model.

C. Additional Experimental Results

C.1. Numerical Experiments with One-Layer Transformer

In this section, we provide numerical evidence of our 1-layer transformer theory introduced in Section 4. We replicate our experiments in the synthetic language seen in Section 3 in a one-layer transformer. As introduced in (Li et al., 2023b), we consider the initialization setting in which both the value and query-key matrices are initialized from a normal distribution with very low variance (0.001).

As a result of this configuration, the attention matrix initially is not updated (i.e. steps 1-10), because the magnitude of the update depends on the entries of the value matrix which are initially small. Later on in training, the attention matrices begin to update (in this case once all facts are learned). This results in the attention shifting. This can be seen as an example of *two-stage* training dynamics for a one-layer transformer, as is observed in (Li et al., 2023b).

In Figure C.1, we plot the attention to the relation token as a function of finetuning training steps. We observe that the attention to the relation token grows throughout training, eventually approaching 1 (i.e. the one-layer transformer entirely

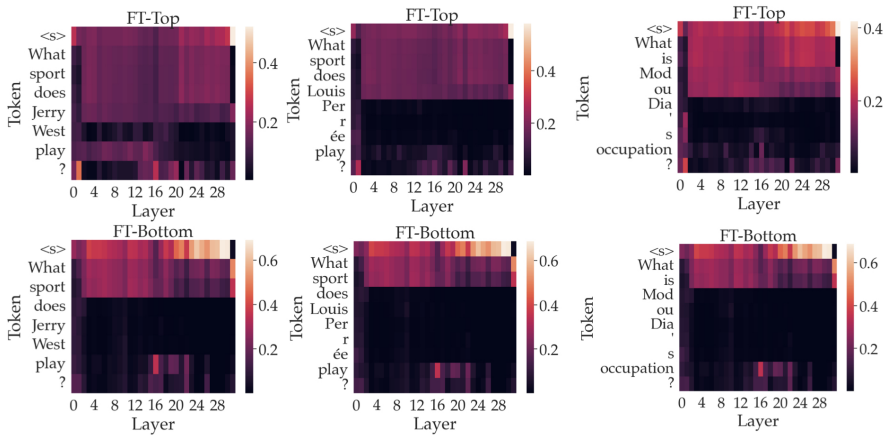


Figure 7. **Additional Attention Maps** We include additional example-specific attention maps for models fine-tuned with `FT-Top` and `FT-Bottom`, respectively. Overall, we find the trends predicted by our analysis of the one-layer transformer continue to hold. In particular, subject attention is markedly reduced for models that are fine-tuned on `FT-Bottom`.

ignores the subject token) and similarly, the subject attention declines to 0. As a result of this, we see that the performance of the model *steeply declines* as training continues, demonstrating "hidden information" caused by the imbalanced attention that we present in our theory.

C.2. Additional Attention Maps

Due to space constraints, we include additional attention maps of models trained on `FT-Top` and `FT-Bottom` in Figure 7. Overall, we observe a reduction of attention scores when evaluating on the `FT-Bottom` model, as we describe in Section 5.