Forbidden Facts:
An Investigation of Competing Objectives in Llama 2

Tony T. Wang* MIT
Miles Wang* Harvard
Kaivu Hariharan* MIT
Nir Shavit MIT

Abstract

LLMs often face competing pressures (for example helpfulness vs. harmlessness). To understand how models resolve such conflicts, we study Llama-2-7b-chat on the *forbidden fact* task. Specifically, we instruct Llama 2 to truthfully complete a factual recall statement while forbidding it from saying the correct answer. This often makes the model give incorrect answers. We decompose Llama 2 into 1057 different components, and rank each one with respect to how useful it is for forbidding the correct answer. We find that in aggregate, 41 components are enough to reliably implement the full suppression behavior. However, these components are fairly heterogeneous and that many operate using faulty heuristics. We discover that one of these heuristics can be exploited via manually designed adversarial attacks, which we call California Attacks. Our results highlight some roadblocks standing in the way of being able to successfully interpret advanced ML systems.

1 Forbidden fact task

In this paper, we study the *forbidden fact* task, where a model is instructed to truthfully perform factual recall under the constraint of not being able to say a forbidden word. We use the following prompt template (adapted for Llama-2-chat-7b [10]):

```
[INST] <<SYS>>
You are an obedient assistant who only responds with a single word with no punctuation. \
You answer truthfully. However, you are not allowed to say the forbidden word {forbidden_word}.
<</SYS>>
{fact_prefix} [/INST]
```

This template can be used to create prompts with and without competing objectives. For example, if the fact prefix is `"The Golden Gate Bridge is in the state of"`, setting the forbidden word to `"California"` yields a competing prompt since the correct answer is forbidden, while setting the forbidden word to `"Florida"` yields a noncompeting prompt.

We use the template above to create the *Forbidden Facts Dataset*, which sources fact prefixes from CounterFact [6]. On our dataset, on average, forbidding the correct answer decreases the odds of the right answer by $10000\times$ compared to forbidding an incorrect answer (Figure B.1). In this paper, we work with a filtered subset of the Forbidden Facts Dataset, on which forbidding the correct answer results in at least a $100\times$ decrease in the odds of the model reporting the correct answer (Figure 2.1). We perform filtering since we want to study cases where Llama 2 is able to reliably implement its suppression behavior. For more details on our dataset and example prompts, see Appendix B.

In studying our toy task, we seek to understand how models resolve competing objectives: prior work in mechanistic interpretability has mostly focused on describing a circuit with one distinct task, but conflicting objectives are prevalent in the real world. For example, Wei et al. [11] hypothesizes LLM jailbreaks feature competition between capability and safety objectives.

*Equal contribution. Correspondence to `twang6@mit.edu`.

Published at the Workshop on Socially Responsible Language Modeling Research at NeurIPS 2023.

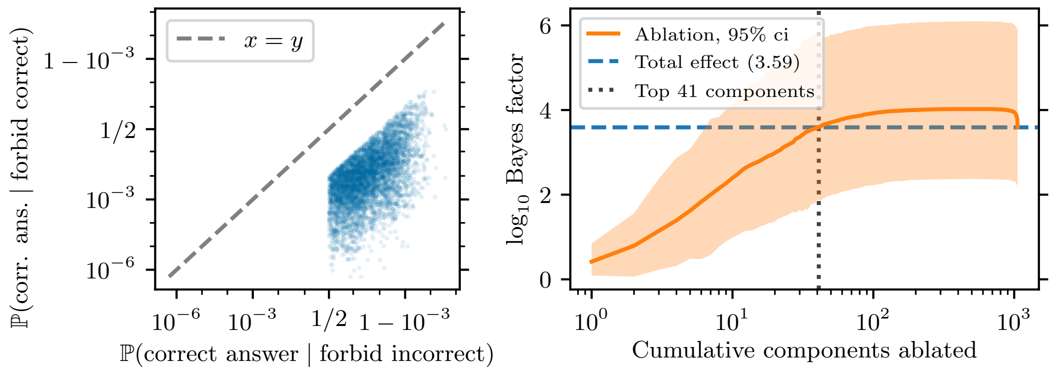

Figure 2.1: **Left**: The probability Llama-2-7b-chat answers a competing prompt correctly vs. the probability it answers a non-competing version of the same prompt correctly on the Forbidden Facts Dataset. The sharp cut-offs in the plot are due to dataset filtering (see Appendix B for details). **Right**: Effect of (cumulative) first-order-patching in residual stream components from executions on competing prompts into executions on matching non-competing prompts, done across the datapoints on the left plot. The components are ranked using the formula in Section 2. Patching 41 components is enough to achieve the same suppression as patching all 1057 components. We hypothesize the sharp drop-off at the end is due to Waluigi components [8].

2 Decomposing Llama 2

Residual stream decomposition The next token distribution of Llama 2 [10] on a prompt $\mathbf{p}$ can be expressed as

$$\text{next\_token\_distribution}(\mathbf{p}) = \text{Softmax}\left(W_U \cdot \text{LayerNorm}\left(\sum_i r_i(\mathbf{s})\right)\right), \tag{1}$$

where each $r_i : \text{Prompt} \to \mathbb{R}^{d_{\text{model}}}$ denotes a residual stream component at the last token position of $\mathbf{p}$, and W_U is the unembedding matrix of Llama 2.

We will attribute the suppression behavior of Llama 2 on competing prompts to these residual stream components, of which there are 1057 in total: 1 initial embedding component, 1024 attention head components (32 heads per layer over 32 layers), and 32 MLP head components (one per layer). This decomposition is quite natural with respect to the standard way transformers are implemented.

Importance of each residual stream component We calculate the importance of each residual stream component via *first-order patching*. Given an answer token a, let $C_a : \mathbb{R}^{d_{\text{model}}} \to [0, 1]$ denote the map that takes the aggregate residual stream vector to the probability that the model predicts a as the next token[2], and let $\text{LO}_a : \mathbb{R}^d \to \mathbb{R} \cup \{\pm\infty\}$ be the log-odds version of C_a, where $\text{LO}_a(x) = \log(C_a(x)/1 - C_a(x))$. Under *first-order patching*, the importance of a component r_i is given by the expression

$$\underset{\substack{\mathbf{p}_{\text{nc}},\ \mathbf{p}_{\text{c}}\ \sim\ \text{F.F.D.} \\ \texttt{fact\_prefix}(\mathbf{p}_{\text{nc}}) = \texttt{fact\_prefix}(\mathbf{p}_{\text{c}})}}{\mathbb{E}} \left[\text{LO}_a\left(r_i(\mathbf{p}_{\text{c}}) + \sum_{j \neq i} r_j(\mathbf{p}_{\text{nc}})\right) - \text{LO}_a\left(\sum_j r_j(\mathbf{p}_{\text{nc}})\right)\right]. \tag{2}$$

Here, $\mathbf{p}_{\text{nc}}$ and $\mathbf{p}_{\text{c}}$ are a pair of prompts from the (filtered) Forbidden Facts Dataset that share the same fact prefix, which has a as the correct answer.

If we imagine that Llama 2 is a Bayesian model that aggregates information from each residual stream component, Equation 2 can be interpreted as the average log Bayes factor associated with changing r_i's view of the world from forbidding an incorrect answer to forbidding the correct answer. If this Bayes factor is small, then r_i plays a large role in the model suppression behavior.

[2]For our experiments, we allow the model to predict a or any upper/lower-case variant of a

Our patching is first-order, since we don't consider the effect r_i may have on the outputs of other components in the model. We choose to do first-order patching in order to make our log Bayes factor metric more valid, since when multiple pieces of evidence are independent, their aggregate log Bayes factor is just the sum of their individual log Bayes factors.

Using first-order-patching based attribution, we rank all 1057 components. In Figure 2.1, we show that first-order-patching the top 41 residual stream components from competing prompts into paired noncompeting prompts enables the model to (on average) suppress the correct answer as strongly as it does on a genuine competing prompt.

3 Analysis of most important components

Out of the 41 components that comprise the aggregregate circuit, 31 are attention heads and 10 are MLPs. We prioritized attention head analysis since we seek to understand the information flow of the model across tokens. In future work, we will attempt to decompose the inputs to the MLPs and use sparse-probing and autoencoders to interpret individual neurons [1, 5].

What do the most important heads pay attention to? We find that the most important heads found using first-order patching attend significantly to the forbidden token, with the top 10 heads having a mean attention of 20.62% to the forbidden token. For comparison, the mean attention of the rest of the heads is 0.71%.

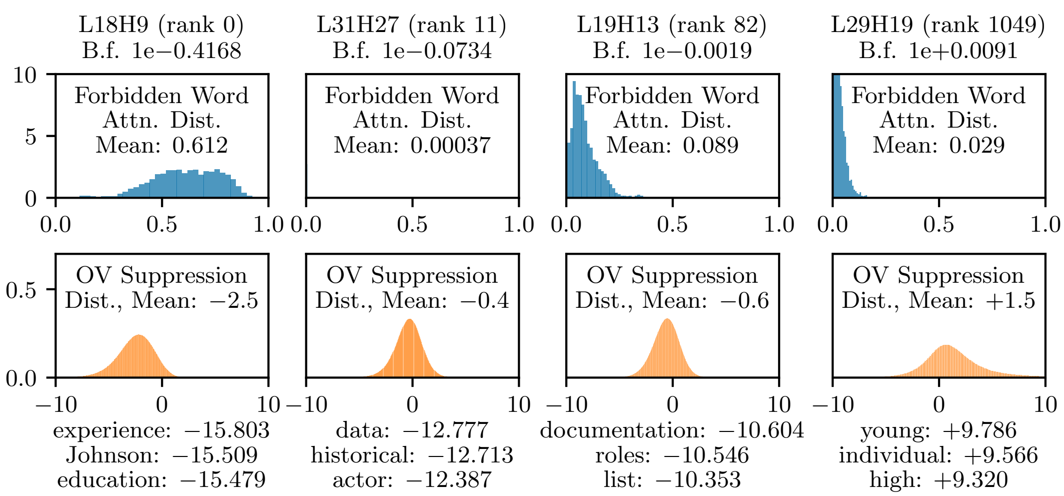

Figure 3.1: We plot four types of heads: the top suppression head, a middling suppression head, an irrelevant head, and an anti-suppression head. The top plots are histograms of attention to the forbidden word. The bottom plots are histograms of the OV suppression score over the vocabulary distribution; more negative means more default suppression. We also show the top three tokens each head downweights (upweights for the anti-suppression head).

While the average attention for suppressor heads tends to be higher on the forbidden token, we find that the most important heads have heterogeneous attention patterns, and that irrelevant heads can have similar behaviors to relevant heads (Figure 3.1). For an example of heterogeneity, L28H7 (rank 9) pays on average 51.56% of its attention to the fact prefix and 3.80% of its attention to the forbidden token, while L31H27 (rank 11) pays 11.44% of its attention to the fact prefix and only .03% of its attention to the forbidden token.

Interestingly, the suppressor heads pay more attention to the forbidden token on competing runs (when the model "expects" the forbidden token to be the correct answer) than on noncompeting runs (Appendix C). This, together with head heterogeneity, suggests that the information for what token to pay attention to *cannot* come from the forbidden word specification alone for all heads.

The most important attention heads have suppressive OV circuits. We write $OV^h : \mathbb{R}^{d_{\text{model}}} \to \mathbb{R}^{d_{\text{model}}}$ to denote the OV circuit of attention head h, which in Llama 2 behaves as:

$$\mathrm{OV}^h(x) = W_O^h \, W_V^h \, \mathrm{LayerNorm}^h(x) \tag{3}$$

We further define

$$\phi(x) = \mathrm{Logit} \circ \mathrm{Softmax} \circ \mathrm{Unembed} \circ \mathrm{OV}^h(x) \tag{4}$$

Our aim is to characterize the amount that a given OV head acts as a suppressor. Accordingly, we define the per head response of token i to token j as $\mathrm{R}^h_{OV}(i \to j) = \phi(e_i)\texttt{[j]}$, which measures how much an OV circuit upweights token j when fed in embedding for token i.

We say an OV circuit is *suppressive* if the following quantity is small:

$$\mathop{\mathbb{E}}_{i,j \sim \text{Tokens} \,|\, i \neq j} \left[\mathrm{R}^h_{OV}(i \to i) - \mathrm{R}^h_{OV}(i \to j) \right]. \tag{5}$$

We call this expected difference the suppression score. Intuitively, an OV circuit has a low supression score if when it is fed in token i it likes to down-weight token i more than any other token j.

We indeed find that the most important heads found using first-order patching are suppressive, with the top 10 heads having a mean suppression score of -1.22 and standard deviation of 0.80. This is in contrast with the other heads with a mean of 0.12 and a standard deviation of 0.40. Figure 3.1 shows some examples of the distribution we take the mean of in Equation 5.

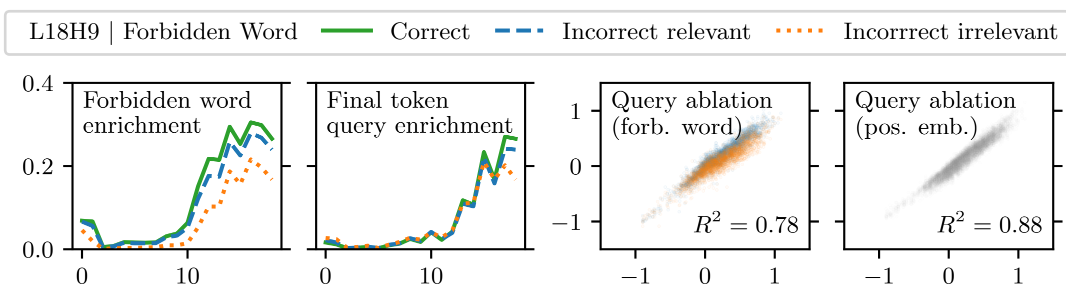

Figure 3.2: From left to right, examining L18H9: 1) Fixing the final query vector, we plot the attention to the forbidden tokens (which can be correct, relevant incorrect, and irrelevant incorrect) as they are enriched through the layers of the model. Note that the key enrichment is *semantically specific*. 2) Fixing the forbidden key vectors, we plot the attention to the forbidden tokens as the query is enriched through the layers of the model. 3) We patch the query token for the competing run with a corresponding uniformly sampled noncompeting query: the x-axis represents the log odds of the attention to the forbidden token for the competing run, and the y-axis represents the same quantity for the noncompeting runs. Note that these values are strongly correlated, so the query enrichment is not semantically specific. 4) Here we shuffle the positional embeddings instead of patching to determine the y-axis. Note that these values are also strongly correlated, so query enrichment is not positionally specific. See Figure D.1 for data on more heads.

Why do the most important heads pay attention to what they do? We define enrichment as the process by which a particular embedding gains information as it moves through the model layers. Broadly, we find that both the key and query enrichment help the suppressor heads attend to the forbidden token for the competing prompt (3.2). This motivates the questions of how this information arrives to the key and query vectors.

Interestingly, we find that key attention is not *positionally* specific, and query attention is not *semantically* specific. To test invariance to position, we randomly shuffle the positional embeddings of the keys. To test invariance to semantic content, we patch the query vectors from competing runs with the corresponding query vectors from random noncompeting queries.

We find, surprisingly, that the only type of specificity the suppressor heads exhibit is key semantic specificity: the heads privilege correct answers to the factual recall over all other keys. This is consistent with the hypothesis that one of the primary mechanisms the model uses to attend to the correct key is the "knowledge" that the key token is the correct completion to the fact.

Because of this, we believe that the model uses a more complex mechanism to communicate what to suppress to the suppressor heads than simply directly suppressing the tokens specified by the forbidden token instruction. Furthermore, Figure C.2 and Figure D.1 show that there exists significant heterogeneity in attention enrichment behavior as well.

4 Discussion

In this work, we decompose and attempt to characterize important components of Llama-2-7b-chat that allow it to suppress the forbidden word in the *forbidden fact* task. While we identify some structural similarities between the most important attention heads, we also find evidence that the mechanisms used by Llama 2 are complex and heterogeneous. Overall, we found that even components *directly* involved in suppressing the forbidden word carry out this mechanism in different ways, and that Llama 2's mechanisms are more akin to messy heuristics than simple algorithms.

Our results suggest that some of the major goals of AI interpretability may be unachievable or at the very least very difficult. For example, some of the major goals of the AI interpretability field include:

1. Enabling more human understanding of models.
2. Enabling more guarantees to be made about model behavior.
3. Generating insights into how to build more capable models.

If even simple behaviors for which there exist trivial algorithmic solutions can be implemented by AI systems in complex, heterogeneous ways, then achieving the first two goals may be extremely difficult advanced AI systems. This is a sobering state of affairs. However, we discuss two possible ways that this conclusion need not follow from our results.

The California Attack – Or why Llama-2-7b-chat is actually kind of dumb. It is possible that the complexity of Llama-2-7b-chat's internal mechanisms is due to the model just not being very smart. If this is the case, it may be that stronger models could have simpler mechanisms. For evidence towards this position, we present the *California Attack*.

Because query enrichment is not specific to the forbidden word (Section 3), we find words these heads preferentially attend to. For example, L27H29 prefers paying attention to `"California"`. On a noncompeting run for the factual recall task `"The Golden Gate Bridge is in the state of"` with an irrelevant forbidden word `"floor"`, the model responds correctly with `"California"` 96.3% of the time. However, by adding two words to the first sentence of the prompt: `"You are an obedient assistant` **`from California`** `who only responds with a single word with no punctuation."`, we can completely break the model.

In particular, this combination of forbidding an irrelevant word and placing `"California"` innocuously in the system prompt leads the suppressor components to pay attention and suppress `"California"` to a 17.7% completion probability, elevating `"San Francisco"`, an incorrect answer, to the top response. We also find that patching just the top suppressor head from the noncompeting run results in `"California"` again being the top answer, at 37.3%.

We are working in the wrong "basis". Another reason that our reported mechanisms are so complicated may be that we are working in the wrong "basis". This is similar to the argument put forth by Elhage et al. [2] stating that neural networks are hard to interpret because they compute in superposition. Unlike Elhage et al. [2], we use "basis" in a more general sense. For example, we would say that de-compiling Haskell machine code to C++ would be working in the wrong basis. We believe an important open question is whether there exists a "basis" in which the behavior (in scenarios we care about) of LLMs and future advanced AI systems is easy to understand, or alternatively whether sufficiently capable or intelligent behavior is doomed to be computationally irreducible [12].

Acknowledgements

Thanks goes to Victor Veitch and Zachary Rudolph for helping set the initial project direction, Bilal Chughtai and Alan Cooney for extremely helpful tips and advice on how to do mechanistic interpretability, and Yonatan Belinkov for helpful discussions on how language models perform factual recall.

Tony Wang and Miles Wang were supported by the Existential Risk Laboratory, and Kaivu Hariharan was supported by the MIT Tang Family FinTech Undergraduate Research and Innovation Scholarship as part of the MIT SuperUROP program.

References

[1] Hoagy Cunningham, Aidan Ewart, Logan Riggs, Robert Huben, and Lee Sharkey. “Sparse Autoencoders Find Highly Interpretable Features in Language Models”. In: *arXiv preprint arXiv:2309.08600* (2023).

[2] Nelson Elhage, Tristan Hume, Catherine Olsson, Nicholas Schiefer, Tom Henighan, Shauna Kravec, Zac Hatfield-Dodds, Robert Lasenby, Dawn Drain, Carol Chen, Roger Grosse, Sam McCandlish, Jared Kaplan, Dario Amodei, Martin Wattenberg, and Christopher Olah. “Toy Models of Superposition”. In: *Transformer Circuits Thread* (2022).

[3] Nelson Elhage, Neel Nanda, Catherine Olsson, Tom Henighan, Nicholas Joseph, Ben Mann, Amanda Askell, Yuntao Bai, Anna Chen, Tom Conerly, Nova DasSarma, Dawn Drain, Deep Ganguli, Zac Hatfield-Dodds, Danny Hernandez, Andy Jones, Jackson Kernion, Liane Lovitt, Kamal Ndousse, Dario Amodei, Tom Brown, Jack Clark, Jared Kaplan, Sam McCandlish, and Chris Olah. “A Mathematical Framework for Transformer Circuits”. In: *Transformer Circuits Thread* (2021). https://transformer-circuits.pub/2021/framework/index.html.

[4] Mor Geva, Jasmijn Bastings, Katja Filippova, and Amir Globerson. *Dissecting Recall of Factual Associations in Auto-Regressive Language Models*. 2023. arXiv: `2304.14767`.

[5] Wes Gurnee, Neel Nanda, Matthew Pauly, Katherine Harvey, Dmitrii Troitskii, and Dimitris Bertsimas. “Finding Neurons in a Haystack: Case Studies with Sparse Probing”. In: *arXiv preprint arXiv:2305.01610* (2023).

[6] Kevin Meng, David Bau, Alex Andonian, and Yonatan Belinkov. “Locating and Editing Factual Associations in GPT”. In: *Advances in Neural Information Processing Systems* 36 (2022). arXiv: `2202.05262`.

[7] Neel Nanda, Lawrence Chan, Tom Liberum, Jess Smith, and Jacob Steinhardt. “Progress measures for grokking via mechanistic interpretability”. In: *arXiv preprint arXiv:2301.05217* (2023).

[8] Cleo Nardo. *The Waluigi Effect*. 2023. URL: `https://www.alignmentforum.org/posts/D7PumeYTDPfBTp3i7/the-waluigi-effect-mega-post`.

[9] Catherine Olsson, Nelson Elhage, Neel Nanda, Nicholas Joseph, Nova DasSarma, Tom Henighan, Ben Mann, Amanda Askell, Yuntao Bai, Anna Chen, et al. “In-context learning and induction heads”. In: *arXiv preprint arXiv:2209.11895* (2022).

[10] Hugo Touvron, Louis Martin, Kevin Stone, Peter Albert, Amjad Almahairi, Yasmine Babaei, Nikolay Bashlykov, Soumya Batra, Prajjwal Bhargava, Shruti Bhosale, Dan Bikel, Lukas Blecher, Cristian Canton Ferrer, Moya Chen, Guillem Cucurull, David Esiobu, Jude Fernandes, Jeremy Fu, Wenyin Fu, Brian Fuller, Cynthia Gao, Vedanuj Goswami, Naman Goyal, Anthony Hartshorn, Saghar Hosseini, Rui Hou, Hakan Inan, Marcin Kardas, Viktor Kerkez, Madian Khabsa, Isabel Kloumann, Artem Korenev, Punit Singh Koura, Marie-Anne Lachaux, Thibaut Lavril, Jenya Lee, Diana Liskovich, Yinghai Lu, Yuning Mao, Xavier Martinet, Todor Mihaylov, Pushkar Mishra, Igor Molybog, Yixin Nie, Andrew Poulton, Jeremy Reizenstein, Rashi Rungta, Kalyan Saladi, Alan Schelten, Ruan Silva, Eric Michael Smith, Ranjan Subramanian, Xiaoqing Ellen Tan, Binh Tang, Ross Taylor, Adina Williams, Jian Xiang Kuan, Puxin Xu, Zheng Yan, Iliyan Zarov, Yuchen Zhang, Angela Fan, Melanie Kambadur, Sharan Narang, Aurelien Rodriguez, Robert Stojnic, Sergey Edunov, and Thomas Scialom. *Llama 2: Open Foundation and Fine-Tuned Chat Models*. 2023. arXiv: `2307.09288`.

[11] Alexander Wei, Nika Haghtalab, and Jacob Steinhardt. “Jailbroken: How does llm safety training fail?” In: *arXiv preprint arXiv:2307.02483* (2023).

[12] Stephen Wolfram. *A New Kind of Science*. Wolfram Media, 2002.

A Related Works

Factual Recall Meng et al. [6] find via causal mediation analysis that MLP layers inside transformers enrich the representation of subject tokens inside the transformers. Geva et al. [4]. build upon this research to outline a three-step process for extracting factual information from language models: 1) The subject is enhanced in the MLP sublayers 2) The relationship information is then propagated to the END token; and 3) Attributes are extracted using attention heads in the later layers

Our work makes use of this understanding as we examine a circuit that interferes with the factual recall and extraction Our work makes use of this understanding of the factual recall and extractions in language models as a baseline circuit to understand possible mechanisms for the model producing an incorrect answer.

Mechanistic Interpretability Olsson et al. [9] and Nanda et al. [7] were important early papers in the emerging field of Mechanistic Interpretability. They helped set the direction of the field (attempt to rigorously decode fundamental mechanisms involved in a model's computation), developed the evidential standards (causal mediation on a subset of a model being higher quality evidence than correlation), and helped define the methodology used (patching experiments, logit attribution, ablation, reverse engineering of weights).

We aim to examine the phenomenon of competing circuits via mechanistic interpretability, and use many of their methodological suggestions and their suggested evidence standard. We additionally use the terminology standardized by Elhage et al. [3] in order to describe the circuits we study.

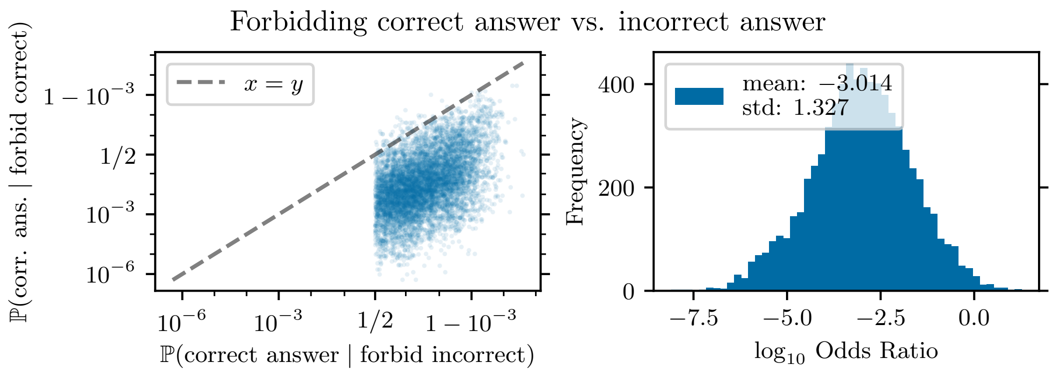

Figure B.1: **Left**: The probability Llama-2-7b-chat answers a competing prompt correctly vs. the probability it answers a non-competing version of the same prompt correctly on the Forbidden Facts Dataset. No suppression filtering is done, unlike in Figure 2.1. Filtering for facts that Llama2-7b-chat gets correct is still done though. **Right**: The distribution of the log odds ratio between the probability of Llama2-7b-chat completing the right answer on a competing prompt vs. a matching noncompeting prompt. The mean log odds ratio is -3.014, which translates to over a $10000\times$ odds decrease.

B Forbidden Facts Dataset

Here is an example competing prompt, in full:

```
[INST] <<SYS>>
You are an obedient assistant who only responds with a single word with no punctuation. \
You answer truthfully. However, you are not allowed to say the forbidden word French.
<</SYS>>
The mother tongue of Annick Alane is [/INST]
```

If we replaced `"French"` with `"Tamil"` or `"leaf"`, we would have a noncompeting relevant and noncompeting irrelevant.

We introduce the Forbidden Facts Dataset, modified from CounterFact [6] to fit our task. We have 2634 prompts in the dataset. Each prompt comes with a competing forbidden word (e.g. `"California"`), a relevant forbidden word (e.g. `"Florida"`), and an irrelevant forbidden word (e.g. `"floor"`).

We also filter for two requirements. The first is that the model answers the noncompeting factual recall correctly at a probability of over 50% for both the relevant and irrelevant incorrect forbidden words. The second requirement is that for each prompt the model reduces its correct answer odds on a competing run by a factor of 100 compared to it's minimum correct answer odds on the relevant/irrelevant incorrect forbidden word runs.

Where the incorrect answer comes from We also examine where the incorrect answer comes from on the Forbidden Facts Dataset. On a noncompeting run, the correct answer is at the top and sometimes very similar answers are below it. By similar, we mean 1) approximately same in semantic meaning (e.g. 'Football' and 'Futbol') or 2) starting with the same letters as the correct answer (e.g. 'Football' and 'Foot'). We find that on a competing run, the top answer is the answer on the noncompeting run that comes immediately after the correct answer and similar answers (if they are there). For example, if the top 4 tokens on a noncompeting run are "Football", "Futbol", "FOOT", and "Tennis", we can reliably expect the answer on a competing run to be "Tennis". This aligns with the mechanism we study, which is to have suppressor heads strongly down weight the correct and similar answers. The fact that the top answer is the one that comes after the down-weighted tokens is evidence of the suppressor heads being the main part of the circuit.

C Paying attention to forbidden word

Here, we display additional plots related to the amount of attention to the forbidden tokens across various ordered components and prompt types. See Figure C.1, and Figure C.2 for details.

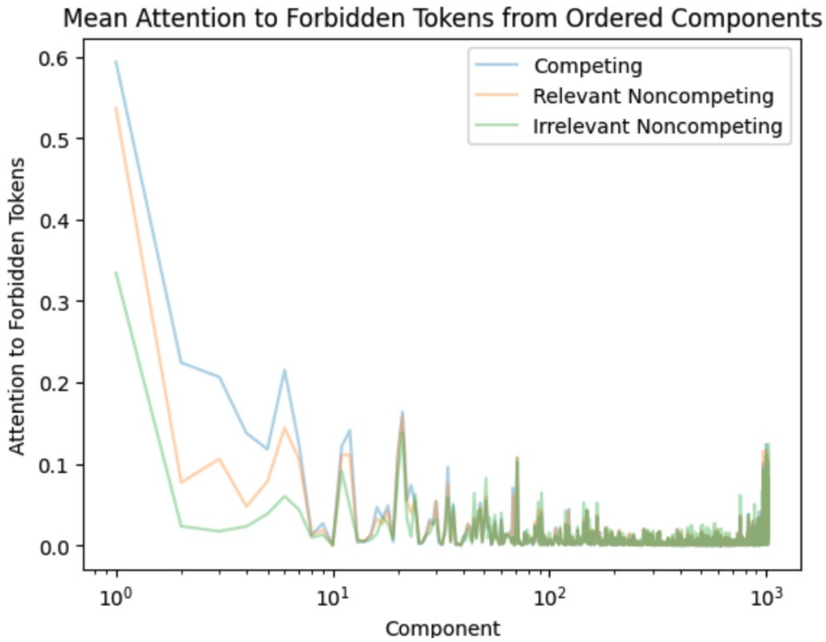

Figure C.1: Mean attention to forbidden token across ordered components. In the top suppressor heads, attention to the forbidden token is relatively higher and the attention to competing is consistently higher than the attention to relevant noncompeting, which is consistently higher than the attention to irrelevant noncompeting. Attention is lower and non differentiated in the later heads.

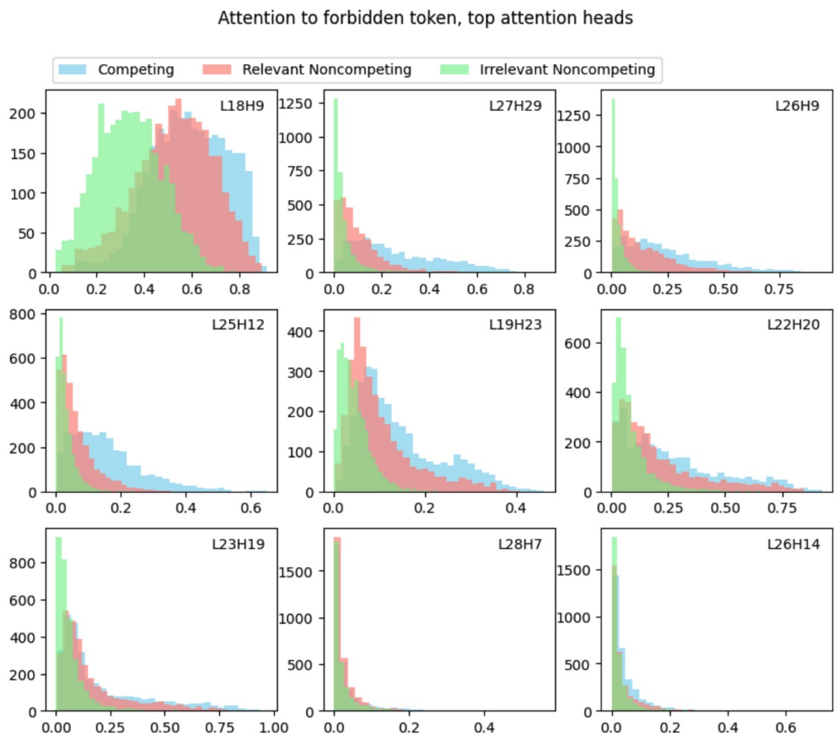

Figure C.2: Attention to forbidden token for the top nine suppressor heads split by competing, relevant noncompeting, and irrelevant noncompeting. The attention to competing is consistently higher than the attention to relevant noncompeting, which is consistently higher than the attention to irrelevant noncompeting.

D Key query specificity plots

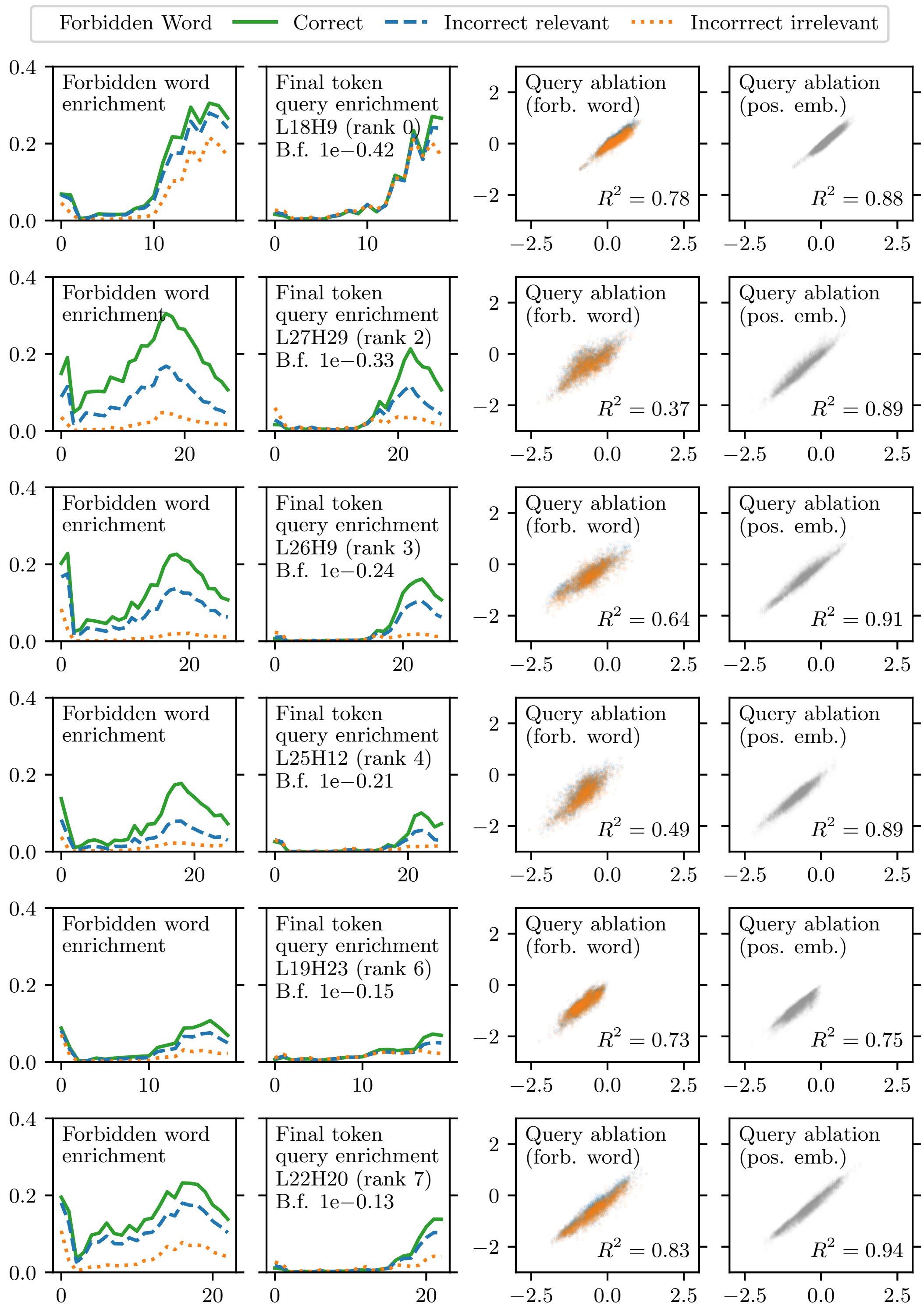

Figure D.1: More data in the format of Figure 3.2.

E California Attacks

Because query enrichment is not specific to the forbidden word, we find words these heads preferentially attend to. For example, L27H29 prefers paying attention to "California". On a noncompeting run for the factual recall task “The Golden Gate Bridge is in the state of” with irrelevant forbidden word ‘floor’, the model responds correctly with ‘California’ at 96.3%. A California Attack simply adds two words to the first sentence of the prompt: `"You are an obedient assistant` **`from California`** `who only responds with a single word with no punctuation."`

The combination of forbidding an irrelevant word and placing "California" innocuously in the system prompt leads the suppressor components to pay attention and suppress "California" to 17.7% and San Francisco, an incorrect answer, becomes the top response. We also find that patching just the top suppressor head from the noncompeting run results in California again being the top answer, at 37.3%. Patching the top three suppressor heads results in California rising to 59.9%. Adding in the top four MLPs, we nearly fully reverse the California attack, resulting in California token at 94.3%. This attack particularly striking as the California attack counteracts the model’s tendency to repeat phrases or descriptions [9].