Racing Thoughts: Explaining Contextualization Errors in Large Language Models

Anonymous ACL submission

Abstract

The profound success of transformer-based language models can largely be attributed to their ability to integrate relevant contextual information from an input sequence in order to generate a response or complete a task. However, we know very little about the algorithms that a model employs to implement this capability, nor do we understand their failure modes. For example, given the prompt *John is going fishing, so he walks over to the bank. Can he make an ATM transaction?*, a model may incorrectly respond *Yes* if it has not properly contextualized *bank* as a geographical feature, rather than a financial institution. We propose the **LLM Race Conditions Hypothesis** as an explanation of contextualization errors of this form. This hypothesis identifies dependencies between tokens (e.g., *bank* must be properly contextualized before the final token, *?*, integrates information from *bank*), and claims that contextualization errors are a result of violating these dependencies. Using a variety of techniques from mechanistic intepretability, we provide correlational and causal evidence in support of the hypothesis, and suggest inference-time interventions to address it.

1 Introduction

Large Language Models (LLMs) have demonstrated a remarkable capacity for accomplishing a wide variety of language generation and classification tasks (Team et al., 2023; Touvron et al., 2023; Brown et al., 2020). Many of these capabilities crucially rely on *contextualization*—integrating relevant contextual information from the input sequence at inference time. Indeed, LLMs' profound success at contextualization has given rise to a novel method of adapting models to new tasks: in-context learning (Achiam et al., 2023; Wei et al., 2022). Although contextualization is fundamental to the success of large language models, we understand fairly little about the mechanisms underlying successful contextualization. At an implementation level, we understand that attention heads allow information to be read from and written to token representations across different positions (Elhage et al., 2021). However, we are currently in the dark about the algorithm-level factors that result in success or failure of contextualization.

Consider the contextualization error presented in Fig. 1 (left). This error was generated by a state of the art LLM, Gemini[1] (Team et al., 2023). Crucially, it betrays that the model has not properly contextualized the representation of *bank*. The model first interprets the word to refer to a geographic feature, and then interprets that same instance of bank in the same context as a financial institution.

In this work, we propose an explanation for contextualization errors in LLMs, denoted the **LLM Race Conditions Hypothesis**. A race condition is a circumstance where a system deploys two or more subroutines in parallel, but relies on these operations executing in a particular order (Huffman, 1954). We identify race conditions occurring throughout the layers of LLMs (right side of Fig. 1). These race conditions occur whenever one token (e.g., the final question-mark token of *Can he make an ATM transaction at this bank?*) must read from a contextualized representation of another (e.g., the earlier instance of the token *bank*). This establishes a dependency where the contextualization of *bank* must complete at an earlier layer than the contextualization of the end-of-sentence delimiter. Out-of-order contextualization can result in incorrect or inconsistent LLM generations.

We verify the LLM Race Conditions Hypothesis using state-of-the-art techniques in mechanistic interpretability, offering qualitative and quantitative evidence as well as several inference-time interventions to ameliorate the problem. We conclude by

[1] Example generated on 9/11/2024.

arguing that race conditions are endemic to feedforward language models, and point toward potential solutions.

In summary, our key contributions are:

1. We identify a simple and robust failure mode of LLM contextualization, which impacts state-of-the-art models across three distinct datasets.

2. We establish the LLM Race Conditions Hypothesis as an explanation for contextualization errors in language models.

3. We apply state-of-the-art techniques from mechanistic interpretability to obtain empirical support for the LLM Race Conditions Hypothesis and suggest inference-time interventions to address the problem.

2 Methods

In this section, we describe the setting in which we study contextualization errors, datasets we construct for this purpose, the tools used for our analyses, and the models under investigation.

Task. To investigate contextualization errors, we define an LLM question-answering task that requires integrating a particular cue that is found in context. Specifically, instances of our task consist of the following components:

1. **Subject Entity:** A noun that will be the subject of the question.

2. **Cue**: A sentence that contextualizes the subject entity, biasing the answer to the question.

3. **Question:** A yes/no question about the subject entity, whose correct answer is disambiguated by the cue.

4. **Distractors:** A random sample of sentences from the WikiText dataset (Merity et al., 2016) injected before and/or after the cue. Distractor sentences are used to systematically increase the difficulty of the question-answering task.

For example: **The game's success led Sega to develop an extensive media franchise.** **I am holding a fishing rod.** **I see a bank.** **Is it a geographical feature?**

For each subject entity and cue, we specify two questions, the correct answer to one of which is "yes" and the other is "no." In the above example, the questions would be ***Is it a geographical feature?*** and ***Is it a financial institution?***. Together, these questions comprise a *question pair*. We measure model performance in terms of question pairs: a question pair is marked correct only if the model answers both questions accurately. This controls for a model's potential bias towards responding "yes" (Dentella et al., 2023).

For each subject entity and cue, we systematically vary the number of distractors included in the prompt, as well as the position of the cue amongst these distractors. To ensure that our analyses are robust to the particular distractors that we randomly select, we include three independent random samples of distractors for each subject entity and cue.

Datasets.

1. **Polysemous Words**: This dataset assesses a model's ability to incorporate contextual cues when deciding on which word sense to adopt for a polysemous word. Subject entities are all 20 words from the CoarseWSD dataset (Loureiro et al., 2021) (e.g., ***bank***). For each subject entity, two cues were manually written in order to bias an LLM toward one sense or another in the absence of distractors (e.g., ***{I am holding a fishing rod / I am trying to make a deposit}***). Questions directly query the sense of the polysemous word (e.g., ***Is it a {geographical feature / financial institution}?***). This dataset contains 120 question pairs (240 questions) for each number of distractors and cue position.

2. **Facts**: This dataset assesses a model's ability to incorporate contextual cues to overwrite known facts, specifically, associations between countries and their capital cities (Merullo et al., 2024). The subject entities of this dataset are countries (e.g., ***Egypt***), and the cues inform the model that the capital city of that country has been renamed to another existing capital city (e.g., ***Forget everything you know about geography. The capital city of Egypt was just renamed from Cairo to Beirut.***). Questions directly query whether this context has been integrated (e.g., ***Is the capital city of Egypt named {Beirut / Cairo}?***). This dataset contains 741 question pairs (1482 questions) for each number of distractors and cue position.

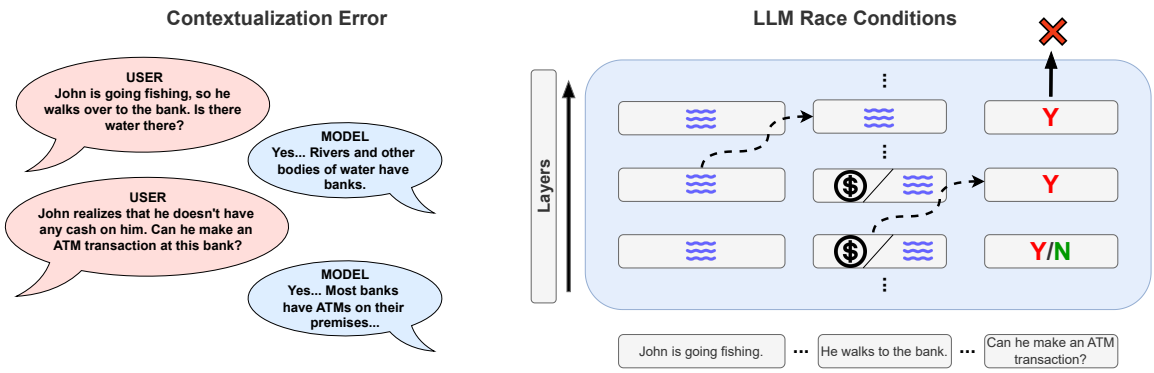

Figure 1: (Left) A contextualization error found in Gemini, a frontier LLM (Team et al., 2023). In this dialogue, the user's first message implies that the correct word sense of *bank* is as a geographical feature (i.e., a river bank). Though the model recognizes this in its first reply, it fails to maintain this word sense of *bank* when probed in the very next user message, instead defaulting to the interpretation of *bank* as a financial institution. (Right) Illustrating the LLM Race Conditions Hypothesis. The Race Conditions Hypothesis suggests that contextualization errors result from out-of-order contextualization over the layers of an LLM. In this case, the question tokens are contextualizing with the polysemous word *bank* in an early layer, before its word sense is resolved via contextualizing with cue tokens.

3. **Gender Bias**: This dataset assesses a model's ability to incorporate contextual cues to overwrite harmful biases. The subject entities of this dataset are the 40 professions included in the WinoBias dataset (e.g., ***soldier***) (Zhao et al., 2018). We generate two cues per profession, implying that the professional is either a man or a woman (e.g., ***The soldier is somebody's {grandfather / grandmother}***). Questions directly query the gender of the professional (e.g., ***Is the soldier a {man / woman}?***). This dataset contains 240 question pairs (480 questions) for each number of distractors and cue position.

Patchscopes. We leverage the Patchscopes framework, which encompasses a set of techniques for inspecting the information that is encoded within a particular intermediate representation of a transformer (Ghandeharioun et al., 2024a). This intermediate or *source* representation is copied to a *target* position in a separate text prompt of the LLM, constructed such that the output will be informative regarding the content of the source. Using the terminology of Ghandeharioun et al., the source representation is defined by a tuple, $h \equiv (S, i, \mathcal{M}, l)$, where S refers to a sequence of n tokens, i refers to a position in that sequence ($i \in [1, ..., n]$), $\mathcal{M}$ refers to a model with L layers, and l refers to a particular layer in $\mathcal{M}$. In other words, a source representation is the result of running $\mathcal{M}$ on S, then extracting the residual stream state of position i from layer l.

The Patchscopes target is defined as another tuple, $\bar{h} \equiv (T, i^*, \mathcal{M}, l^*)$, where T is a target prompt consisting of n^* tokens, i^* refers to a position in T, $\mathcal{M}$ refers to the same model that generates the source representation, h, and l^* refers to a layer in $\mathcal{M}$. A patchscope intervention refers to replacing the representation $\bar{h}$ with h at inference time. T is constructed such that the output of $\mathcal{M}$ will be informative regarding the content of h.

Models. We study `gemma-2-9b-it` (Team et al., 2024), and replicate using another commonly used open-weight model in § G. These models have been specifically tuned for instruction-following and chat capabilities and represent a sample of the state of the art in open-weight models. This work required approximately 50 GPU-hours, using Nvidia A100 GPUs.

3 Behavioral Failure Modes

In this section we document a simple and robust failure mode of contextualization in LLMs. Inspired by recent work investigating context integration in multi-document question answering (Liu et al., 2024) and long-context retrieval (Kamradt, 2024), we inject randomly-selected distractor sentences into prompts that contains a context-sensitive question. We find that the presence of distractors significantly deteriorates a model's ability to integrate relevant contextual information when queried with context-dependent questions. For each of the three domains described in § 2, we generate six datasets, containing between zero and five distractors. We then evaluate the LLM's ability

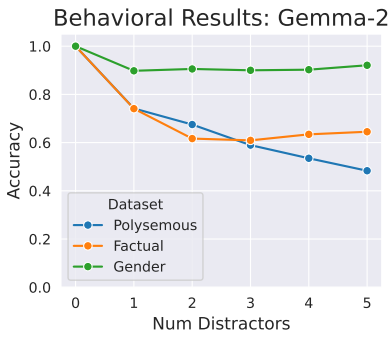

Figure 2: Model accuracy vs. number of distractors. We find that injecting distractor text into the prompt causes sharp performance degredations across all datasets.

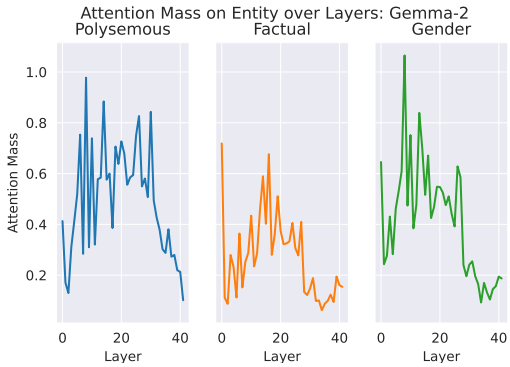

Figure 3: Attention mass over layers for all datasets for Gemma-2. We observe an inverse U-shape over layers, suggesting that the question tokens might only incorporate information present in the subject entity in the middle layers of processing.

to answer question pairs correctly in the face of distractors. The results are shown in Fig. 2. We observe that, across the board, distractor text greatly harms a model's ability to properly contextualize. Furthermore, we observe different performance depending on where the cue is interleaved within the distractor sentences (See § A).

4 The LLM Race Conditions Hypothesis

To explain the behavioral results presented in § 3, we propose the *LLM Race Conditions Hypothesis*, illustrated pictorially in Fig. 1.

> **LLM Race Conditions Hypothesis**. Proper contextualization is a race between parallel processes operating across the layers of an LLM. One of these processes involves contextualizing the subject entity token with the cue; the other process involves contextualizing the question tokens with the subject entity in order to generate an answer. If the question tokens read from the subject entity before contextualization is complete, the LLM is more likely to produce an incorrect answer.

The LLM Race Conditions Hypothesis can be broken into two separate, but related, claims: (1) There exists a critical window in which the question tokens are contextualized with the subject entity (and other tokens). Any further refinement of the subject entity after the critical window will not be reflected in model responses. (2) Model failures are driven by unfinished contextualization of the subject entity at *early* layers (i.e., before the end of the critical window). In the following sections, we provide correlational and causal evidence in support of both of these claims.

For all of the analyses to follow, we select a dataset slice (i.e., number of distractors and cue position) that results in as close to 50% accuracy as possible for each model-dataset pair. This is helpful because some interventions have the goal of increasing accuracy, and others have the goal of decreasing accuracy through ablation. Note that 50% is **not** chance accuracy, as we are measuring in terms of question *pairs*, which require two correct answers (corresponding to one "yes" and one "no"). Thus, 25% is chance accuracy. See § B for details.

4.1 A Critical Window

We first verify the existence of a critical window, i.e., a set of layers in which the question token representations integrate information from the broader context, and after which they commit to an answer. We support this by (1) analyzing attention patterns, and finding that attention to the subject entity peaks in the middle layers, (2) performing a logit lens analysis of the final token of the prompt, and showing that the model's ultimate decision is decipherable halfway through the network, and (3) performing a causal analysis by modulating attention patterns, and demonstrating that contextualization fails to impact downstream performance after the middle layers of the model.

Attention Mass Analysis. In each dataset, correct answers crucially rely on the contextualization of a subject entity (i.e., a polysemous word, a profession, or a country). We begin our investigation by analyzing attention to the subject entities at each layer by all subsequent tokens (which comprise the

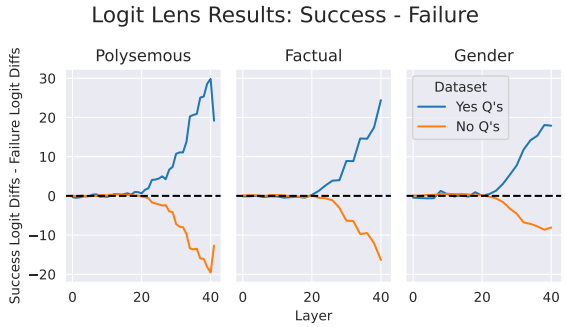

Figure 4: Logit lens results for Gemma-2 over all three datasets. For each layer, we plot the difference in mean logit differences between the 'yes' token and 'no' token between questions that the model answered correctly vs. incorrectly. Each dataset is disaggregated to separate questions where the correct answer is 'yes' from those where the correct answer is 'no.' This metric demonstrates the impact of successful contextualization on the final question token over layers. For all datasets and partitions, we find that the model's ultimate answer becomes identifiable around layer 20.

question). Specifically, for each layer l, we compute the mean attention paid to the (final token of the) subject entity over all heads by all subsequent tokens, which we call the *attention mass*:

$$\text{AttentionMass}_{l,s} = \sum_{n \in [s+1,\ldots,N]} \sum_{h \in H} A_{n,s,l,h} \quad (1)$$

where s is the index of the subject entity token, N is the total number of tokens in the sequence, H is the set of attention heads, and $A_{n,s,l,h}$ is an attention map, providing the normalized attention score from token n to token s in layer l for head h. The attention mass for each data set is plotted as a function of layer in Fig. 3.

Though this is a coarse, correlational metric, we nevertheless observe an obvious pattern across all three datasets: the ensuing tokens appear to allocate more attention mass to the subject entity token in the middle layers of processing, as compared to the earlier or later layers. This suggests that the final tokens in the query may only be integrating information present in the subject entity in intermediate (rather than late) layers. If relevant information becomes available within the representation of the subject entity after this critical window, it may not be reflected in the model's ultimate response.

Logit Lens. If the middle layers identified during the attention mass analysis truly comprise a critical window, then we would expect that the representation of the final token becomes properly contextualized enough to display a preference towards its ultimate decision during and after this window. We implement logit lens (nostalgebraist, 2020) as a patchscope (see Ghandeharioun et al., 2024a, for details), and record the logit difference between the "yes" token and "no" token over layers for all three datasets using Gemma-2. This analysis is performed at the individual question level. Formally, given a last token representation h_l generated by layer l and the unembedding matrix W_U, with column y corresponding to the "yes" token and column n corresponding to the "no" token, we define $\text{log_odds}(h_l) = W_{U_y} h_l - W_{U_n} h_l$.

In Fig. 4, the bifurcation indicates the point at which the logit lens indicates the model's ultimate decision. This graph is formed by splitting each dataset (with distractors) according to whether the correct answer is ultimately "yes" or "no". Let these splits be termed D_y and D_n, respectively. Furthermore, we split each partition according to whether the model ultimately gets the answer correct. Let D_y^+ and D_y^- denote these further partitions of D_y, and similarly for D_n. For each layer, l, we report the following for D_y, and similarly for D_n:

$$\text{diff}_{y,l} = \frac{\sum_{h_l \in D_y^+} \text{log_odds}(h_l))}{|D_y^+|} - \frac{\sum_{h_l \in D_y^-} \text{log_odds}(h_l))}{|D_y^-|}$$

This value quantifies when the logit differences of a successful model generation differs from the logit differences of a failure, while controlling for intrinsic biases toward producing affirmative or negative answers that occur throughout the layers of a model (see § C). From Fig. 4, we see that there is no difference between success and failures until the middle layers of the network. After this point, there is clear signal indicating the answer that the model will ultimately generate. Thus, we might pin down the end of the critical window approximately to layer 20, or shortly thereafter. See § C for disaggregated details, which further support the conclusions presented here, and additional analysis.

Attention Ablation. Given the qualitative results obtained by analyzing attention and logit lens, we should expect that contextualization should only impact model performance in early and middle layers, before the question tokens finish reading from the subject token(s). We perform causal interventions on the attention maps produced throughout the transformer layers to test this. Specifically, we perform attention ablations: zeroing out the attention map entry that defines how much attention

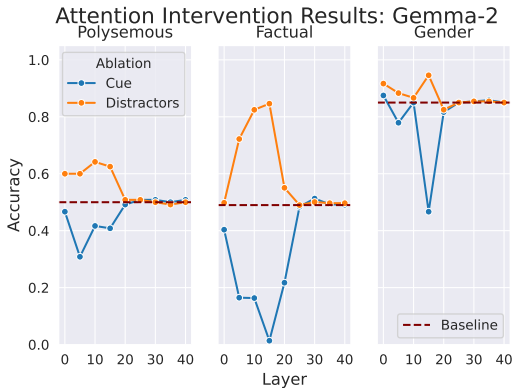

Figure 5: Attention Ablation Results. Ablating cue tokens and distractor tokens both have the intended impact on performance – ablating cue tokens drops performance and ablating distractors increases performance. Notably, the interventions only impact model performance in the first half of layers.

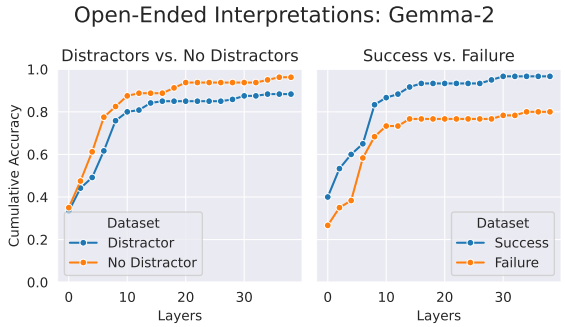

Figure 6: Open-Ended Interpretation Results. We employ an open-ended patchscope to generate natural language interpretations of the semantic content within the subject entities of the polysemous dataset over layers. (Left) We find that distractors delay the proper contextualization of subject entities. (Right) We find that questions that the model answers incorrectly are overall more poorly contextualized than questions that the model answers correctly.

one token pays to another and renormalizing the resulting values. This effectively removes the influence of one token on another within the layers in which we apply this intervention. Formally, let T_E be the set of tokens to edit and T_A be the set of tokens to ablate. Attention ablations are defined as follows: $\forall_{e \in T_E, a \in T_A} A_{e,a} \leftarrow 0$. We use the `transformerlens` package to implement this intervention (Nanda and Bloom, 2022). We perform two types of attention ablations: distractor ablation, where T_A corresponds to the tokens comprising distractor sentences, and cue ablation, where T_A corresponds to tokens comprising the cue. Distractor ablation mimics the no distractors setting—up to a difference in positional embeddings—over the layers in which it is applied, and should boost performance. Cue ablation should leave the model unable to properly contextualize the subject entity, harming accuracy.

We perform distractor ablation and cue ablation in blocks of 5 layers, and analyze the impact of attention ablation over the course of the model. Our main Gemma-2 results are presented in Fig. 5, and a replication with another LLM is presented in § G. In all cases, we find that distractor ablation reliably increases performance in early layers, and cue ablation reliably decreases performance in early layers. Furthermore, contextualization interventions stop impacting model performance at approximately halfway through the model layers—almost precisely when logit lens reveals contextualization of the final question token.

4.2 Subject Entity Contextualization

Next, we verify that model failures are caused by unresolved contextualization of the subject entity in early layers of the model (i.e., before the end of the critical window). We support this by (1) analyzing the content of subject entity representations over layers, and finding that distractors deteriorate contextualization, (2) performing causal patching analyses, which show that contextualization of the subject entity mediates model performance and that well-contextualized representations emerge in late layers of the model.

Open-Ended Interpretations. First, we qualitatively investigate how the content of subject entity representations evolve over layers of processing. To do so, we employ an open-ended Patchscope (Ghandeharioun et al., 2024a) inspired by the SelfIE technique (Chen et al., 2024). This technique formulates a patchscope where the target prompt, T, is an *interpretation prompt*—a prompt designed to encourage the model to describe the hidden representation. Source representations are extracted from i, where i is position of the last token of the subject entity, and l ranges over every even layer (i.e., $l \in [0, 2, 4..., N]$) in an LLM, $\mathcal{M}$. The source representations are patched into a target defined by $T =$ "`Tell me about X X X`", $i^* =$ the index of the `X` tokens, and $l^* = 3$ (following Chen et al., 2024). We then condition the model to respond with "`Sure! In this context, the word refers to`" and allow the model to freely generate 15 tokens. This process ideally results in

a natural language description of the content of the hidden representation.

We employ this technique to investigate how the word sense of the subject entities in our polysemous word dataset evolve over the layers of the model. Following prior work (Ghandeharioun et al., 2024b), we use another LLM[2] to autoscore these generations as being indicative of one sense or another. See § D for details and § E for example interpretations. We present the results of this analysis in Fig. 6, showing cumulative accuracies over layers. A question pair is marked correct at layer m if the model has properly contextualized the intermediate representation in at least one layer [0, m]. In Fig. 6 (left), we compare question pairs with and without distractors. We find that the presence of distractors delays proper contextualization of intermediate representations, shifting the cumulative accuracy curve for question pairs with distractors to the right. This delay may limit the number of layers within the critical window where the subject tokens are well-contextualized. Furthermore, in Fig. 6 (right), we compare distractor question pairs that the model ultimately answers correctly with those that the model ultimately answers incorrectly. We find that cumulative accuracy is lower overall in failure cases, indicating that subject entities are sometimes never properly contextualized during model failures.

Cross-Patching. At minimum, the LLM Race Conditions Hypothesis implies that improperly contextualized subject-entity representations somewhere in the model are causing incorrect responses. Behaviorally, we know that the model achieves ceiling performance without distractors, implying that the model contains a properly contextualized subject-entity representation. Can we boost performance in the presence of distractors by injecting subject-entity representations generated by models when they are faced with data without distractors? To test this, we perform a Patchscopes intervention. Source representations are specified where S corresponds to a no-distractors prompt, i corresponds to the position of the last token of the subject entity in S, $\mathcal{M}$ corresponds to an LLM with N layers, and $l \in [0, 2, 4, 6, ..., M]$, where $M = N/2$. Target representations are specified where T corresponds to a prompt with distractors, i^* refers to the position of the last token of the subject entity in T, and $l^* = l$. We refer to this intervention as *cross-patching*. We search over layers for each question pair separately. If any source layer-target layer combination results in correct generations for both questions, we consider the question pair correct. From Fig. 7, one can see that cross-patching reliably boosts performance across all datasets. See § G for a replication using Llama-2.

[2] PaLM2 `text-unicorn@001` (Anil et al., 2023)

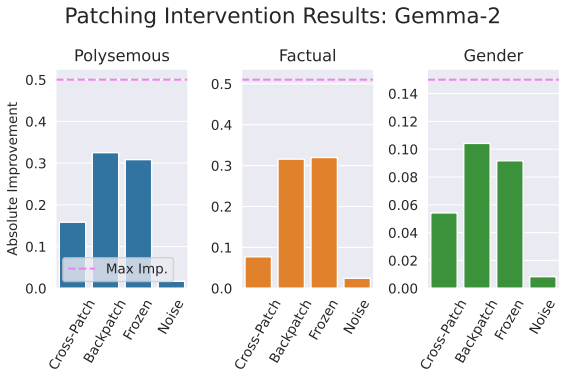

Figure 7: Gemma-2 Patching Results: We perform three patching interventions and a baseline for each dataset. First, we find that patching a subject token from a no-distractors question into a distractors question boosts performance (Cross-Patching) over the baseline (Noise). Second, we observe that extracting later layers' representations and patching them into early layers also boosts performance (Backpatching, Frozen Backpatching). For reference, the maximum possible improvement in accuracy is represented by a dashed line.

Backpatching. The LLM Race Conditions Hypothesis requires that subject entity contextualization must happen either before or during the critical window of question contextualization. Otherwise, the properly contextualized subject entity would not be integrated in the final prediction. To test this, we employ a *backpatching* intervention described in (Biran et al., 2024): source representations are extracted from a prompt, S, at some position i and layer l. Target representations are defined by $T = S$, $i^* = i$ and $l^* < l$. In our case, we let $l^* \in [0, 2, 4, ..., M]$ and $l \in [M, M+2, M+4, ..., N]$, where the LLM $\mathcal{M}$ has L layers and $M = L/2$. For each layer in the target set, we attempt to patch in a representation from each layer in the source set. In Fig. 7, we find that this reliably increases performance across all datasets. See § G for a replication using Llama-2.

Frozen Backpatching. In the previous experiment, the backpatched representation undergo further processing through the MLP and attention mechanisms after the target layer, l^*. Perhaps the performance advantages conferred by backpatch-

ing are a result of this extra computation, rather than the relative ordering of question contextualization and subject entity contextualization? To address this, we rerun the same experiment and source-target layer search, except we patch the source representation into *every* layer in between the source and target. Formally, if l' indicates the set of target layers, $l' = [l^*, l^*+1, ..., l]$. This effectively freezes the representation, rendering it insensitive to MLP and attention mechanisms throughout l'. From Fig. 7, we see that this setting produces similar result as the standard backpatching setting. Thus, the ordering of contextualization, rather than extra computation, engenders higher performance. See § F for further analysis, including results from every source-target layer combination. See § G for a replication using Llama-2.

Baseline. All patching experiments require searching over a space of perturbations over word embeddings. Is it the case that a random perturbation would produce similar results? To test this possibility, we run a baseline intervention where we perturb intermediate representations with Gaussian noise in each dimension. Formally, we extract a source representation, h, where S corresponds to a prompt (with distractors), i corresponds to the position of the last token of the subject entity in S, $\mathcal{M}$ corresponds to an LLM with N layers, and $l \in [0, 2, 4, 6, ..., M]$, where $M = N/2$. We then sample Gaussian noise, ϵ with one of two levels of strength, $\epsilon \sim \{\mathcal{N}(\mu = 0, \sigma = h * .01), \mathcal{N}(\mu = 0, \sigma = h * .05\}$. We form our final source representation as $\bar{h} = h + \epsilon$. We patch this representation into a target defined by $T = S$, $i^* = i$, $l^* = l$. For each question pair, we iterate over all $l \in [0, 2, 4, 6, ..., M]$. For each l, we resample ϵ 10 times per noise level (i.e. $(M/2) \times 20$ different perturbation attempts per question pair) and intervene. We record whether any of these perturbations result in correct answers to a question pair. We find that this intervention boosts performance very modestly, far less than any of the experimental interventions.

5 Related Work

This work takes inspiration from the burgeoning field of mechanistic interpretability, which attempts to understand the circuits and algorithms that a neural network implements in order to accomplish a particular task (Olah, 2022). Our work contributes to a trend of using *patching* to study algorithms implemented in production-level LLMs (Ghandeharioun et al., 2024a; Biran et al., 2024; Wu et al., 2024; Lieberum et al., 2023; Wang et al., 2022; Geiger et al., 2021; Heimersheim and Nanda, 2024).

Several recent works have attempted to understand in-context learning (ICL), which might be viewed as an extreme case of context-dependent generation (e.g., Akyürek et al., 2022; Von Oswald et al., 2023; Wies et al., 2024; Falck et al.; Hahn and Goyal, 2023; Xie et al., 2022). Our work complements this line of research by studying contextualization in more natural settings. Finally, this work contributes to a body of work that identifies distinct processing stages throughout the layers of language models (Tenney et al., 2019). Recent mechanistic work has localized refusal mechanisms to the early layers of LLMs (Ghandeharioun et al., 2024b) and suggested the existence of universal stages of processing (Lad et al., 2024). Earlier work in interpretability has studied the impact of training objective on processing stages (Voita et al., 2019). The present study demonstrates distinct stages of contextualization over layers and suggests interventions that are informed by these stages.

6 Conclusion

In this work, we introduced a hypothesis to explain contextualization errors in LLMs: the LLM Race Conditions Hypothesis. After establishing a setting for studying contextualization errors systematically, we tested two predictions of the LLM Race Conditions Hypothesis: 1) that there exists a critical window when the question tokens are contextualized, and 2) that model failures are caused by improper contextualization of the subject entity during this window. This is a natural consequence of purely feedforward architectures, as early layers simply do not have access to later layer representations.

This work provides an example of using techniques from mechanistic interpretability to understand a problem found in state-of-the-art LLMs. By exposing and understanding algorithmic shortcomings of these models, one can suggest plausible solutions for addressing them. Our work-in-progress aims to explore two potential solutions. First, new models might include recurrent connections, allowing later layer representations of earlier tokens to influence earlier layer representations of later tokens. This would enable deep, well-contextualized representations to influence the entire residual stream of all ensuing tokens. Second, this problem may be ameliorated in existing pretrained models by adopting more advanced versions of the inference-time interventions presented in § 4.2.

7 Limitations

This work conducts a mechanistic analysis of one form of contextualization error that occurs in LLMs. Though the LLM Race Conditions Hypothesis appears to apply across a range of settings (as indicated by the diverse datasets employed in this work), it does not comprehensively explain all LLM contextualization errors. In particular, errors due to missing factual knowledge (Hase et al., 2023) lie outside the scope of this work. Additionally, this work suggests several remediations, including those based on the inference-time interventions presented in § 4.2. However, we note that these interventions are not practical in their current state, as they require one to know which entities to patch and which layers to use as source and target a priori. Future work must develop these ideas further before employing them in a production system to address LLM Race Conditions.

8 Ethical Considerations

This work demonstrates that contextualization errors occur in settings where a model is likely to encode a gender bias associated with a particular profession. This adds a small contribution to a longstanding literature describing gender biases in language models (May et al., 2019; Rudinger et al., 2018; Weidinger et al., 2021). As noted in § 7, none of the interventions presented here should be used to address issues of gender bias in production language models.

References

Josh Achiam, Steven Adler, Sandhini Agarwal, Lama Ahmad, Ilge Akkaya, Florencia Leoni Aleman, Diogo Almeida, Janko Altenschmidt, Sam Altman, Shyamal Anadkat, et al. 2023. Gpt-4 technical report. *arXiv preprint arXiv:2303.08774*.

Ekin Akyürek, Dale Schuurmans, Jacob Andreas, Tengyu Ma, and Denny Zhou. 2022. What learning algorithm is in-context learning? Investigations with linear models. In *The Eleventh International Conference on Learning Representations*.

Rohan Anil, Andrew M Dai, Orhan Firat, Melvin Johnson, Dmitry Lepikhin, Alexandre Passos, Siamak Shakeri, Emanuel Taropa, Paige Bailey, Zhifeng Chen, et al. 2023. Palm 2 technical report. *arXiv preprint arXiv:2305.10403*.

Eden Biran, Daniela Gottesman, Sohee Yang, Mor Geva, and Amir Globerson. 2024. Hopping too late: Exploring the limitations of large language models on multi-hop queries. *arXiv preprint arXiv:2406.12775*.

Tom Brown, Benjamin Mann, Nick Ryder, Melanie Subbiah, Jared D Kaplan, Prafulla Dhariwal, Arvind Neelakantan, Pranav Shyam, Girish Sastry, Amanda Askell, Sandhini Agarwal, Ariel Herbert-Voss, Gretchen Krueger, Tom Henighan, Rewon Child, Aditya Ramesh, Daniel Ziegler, Jeffrey Wu, Clemens Winter, Chris Hesse, Mark Chen, Eric Sigler, Mateusz Litwin, Scott Gray, Benjamin Chess, Jack Clark, Christopher Berner, Sam McCandlish, Alec Radford, Ilya Sutskever, and Dario Amodei. 2020. Language models are few-shot learners. In *Advances in Neural Information Processing Systems*, volume 33, pages 1877–1901. Curran Associates, Inc.

Haozhe Chen, Carl Vondrick, and Chengzhi Mao. 2024. Selfie: Self-interpretation of large language model embeddings. In *Forty-first International Conference on Machine Learning*.

Vittoria Dentella, Fritz Günther, and Evelina Leivada. 2023. Systematic testing of three language models reveals low language accuracy, absence of response stability, and a yes-response bias. *Proceedings of the National Academy of Sciences*, 120(51):e2309583120.

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. 2021. A mathematical framework for transformer circuits. *Transformer Circuits Thread*. Https://transformer-circuits.pub/2021/framework/index.html.

Fabian Falck, Ziyu Wang, and Christopher C Holmes. Is in-context learning in large language models bayesian? a martingale perspective. In *Forty-first International Conference on Machine Learning*.

Atticus Geiger, Hanson Lu, Thomas Icard, and Christopher Potts. 2021. Causal abstractions of neural networks. *Advances in Neural Information Processing Systems*, 34:9574–9586.

Asma Ghandeharioun, Avi Caciularu, Adam Pearce, Lucas Dixon, and Mor Geva. 2024a. Patchscopes: A unifying framework for inspecting hidden representations of language models. In *Forty-first International Conference on Machine Learning*.

Asma Ghandeharioun, Ann Yuan, Marius Guerard, Emily Reif, Michael A Lepori, and Lucas Dixon. 2024b. Who's asking? user personas and the mechanics of latent misalignment. *arXiv preprint arXiv:2406.12094*.

Michael Hahn and Navin Goyal. 2023. A theory of emergent in-context learning as implicit structure induction. *arXiv preprint arXiv:2303.07971*.

Peter Hase, Mohit Bansal, Been Kim, and Asma Ghandeharioun. 2023. Does localization inform editing? surprising differences in causality-based localization vs. knowledge editing in language models. In *Advances in Neural Information Processing Systems*, volume 36, pages 17643–17668. Curran Associates, Inc.

Stefan Heimersheim and Neel Nanda. 2024. How to use and interpret activation patching. *arXiv preprint arXiv:2404.15255*.

David A Huffman. 1954. The synthesis of sequential switching circuits. *Journal of the franklin Institute*, 257(3):161–190.

Greg Kamradt. 2024. Needle in a haystack - pressure testing llms.

Vedang Lad, Wes Gurnee, and Max Tegmark. 2024. The remarkable robustness of llms: Stages of inference? In *ICML 2024 Workshop on Mechanistic Interpretability*.

Tom Lieberum, Matthew Rahtz, János Kramár, Neel Nanda, Geoffrey Irving, Rohin Shah, and Vladimir Mikulik. 2023. Does circuit analysis interpretability scale? evidence from multiple choice capabilities in chinchilla. *arXiv preprint arXiv:2307.09458*.

Nelson F Liu, Kevin Lin, John Hewitt, Ashwin Paranjape, Michele Bevilacqua, Fabio Petroni, and Percy Liang. 2024. Lost in the middle: How language models use long contexts. *Transactions of the Association for Computational Linguistics*, 12:157–173.

Daniel Loureiro, Kiamehr Rezaee, Mohammad Taher Pilehvar, and Jose Camacho-Collados. 2021. Analysis and evaluation of language models for word sense disambiguation. *Computational Linguistics*, 47(2):387–443.

Chandler May, Alex Wang, Shikha Bordia, Samuel Bowman, and Rachel Rudinger. 2019. On measuring social biases in sentence encoders. In *Proceedings of the 2019 Conference of the North American Chapter of the Association for Computational Linguistics: Human Language Technologies, Volume 1 (Long and Short Papers)*, pages 622–628.

Stephen Merity, Caiming Xiong, James Bradbury, and Richard Socher. 2016. Pointer sentinel mixture models. *Preprint*, arXiv:1609.07843.

Jack Merullo, Carsten Eickhoff, and Ellie Pavlick. 2024. Language models implement simple word2vec-style vector arithmetic. In *Proceedings of the 2024 Conference of the North American Chapter of the Association for Computational Linguistics: Human Language Technologies (Volume 1: Long Papers)*, pages 5030–5047.

Neel Nanda and Joseph Bloom. 2022. Transformerlens. https://github.com/TransformerLensOrg/TransformerLens.

nostalgebraist. 2020. interpreting gpt: the logit lens. *LessWrong*.

Christopher Olah. 2022. Mechanistic interpretability, variables, and the importance of interpretable bases. *Transformer Circuits Thread*.

Rachel Rudinger, Jason Naradowsky, Brian Leonard, and Benjamin Van Durme. 2018. Gender bias in coreference resolution. In *Proceedings of the 2018 Conference of the North American Chapter of the Association for Computational Linguistics: Human Language Technologies, Volume 2 (Short Papers)*, pages 8–14.

Gemini Team, Rohan Anil, Sebastian Borgeaud, Yonghui Wu, Jean-Baptiste Alayrac, Jiahui Yu, Radu Soricut, Johan Schalkwyk, Andrew M Dai, Anja Hauth, et al. 2023. Gemini: a family of highly capable multimodal models. *arXiv preprint arXiv:2312.11805*.

Gemma Team, Thomas Mesnard, Cassidy Hardin, Robert Dadashi, Surya Bhupatiraju, Shreya Pathak, Laurent Sifre, Morgane Rivière, Mihir Sanjay Kale, Juliette Love, et al. 2024. Gemma: Open models based on gemini research and technology. *arXiv preprint arXiv:2403.08295*.

Ian Tenney, Dipanjan Das, and Ellie Pavlick. 2019. Bert rediscovers the classical nlp pipeline. In *Proceedings of the 57th Annual Meeting of the Association for Computational Linguistics*, pages 4593–4601.

Hugo Touvron, Louis Martin, Kevin Stone, Peter Albert, Amjad Almahairi, Yasmine Babaei, Nikolay Bashlykov, Soumya Batra, Prajjwal Bhargava, Shruti Bhosale, et al. 2023. Llama 2: Open foundation and fine-tuned chat models. *arXiv preprint arXiv:2307.09288*.

Elena Voita, Rico Sennrich, and Ivan Titov. 2019. The bottom-up evolution of representations in the transformer: A study with machine translation and language modeling objectives. In *2019 Conference on Empirical Methods in Natural Language Processing and 9th International Joint Conference on Natural Language Processing*, pages 4387–4397. Association for Computational Linguistics (ACL).

Johannes Von Oswald, Eyvind Niklasson, Ettore Randazzo, João Sacramento, Alexander Mordvintsev, Andrey Zhmoginov, and Max Vladymyrov. 2023. Transformers learn in-context by gradient descent. In *International Conference on Machine Learning*, pages 35151–35174. PMLR.

Kevin Ro Wang, Alexandre Variengien, Arthur Conmy, Buck Shlegeris, and Jacob Steinhardt. 2022. Interpretability in the wild: a circuit for indirect object identification in gpt-2 small. In *The Eleventh International Conference on Learning Representations*.

Jason Wei, Xuezhi Wang, Dale Schuurmans, Maarten Bosma, Fei Xia, Ed Chi, Quoc V Le, Denny Zhou,

et al. 2022. Chain-of-thought prompting elicits reasoning in large language models. *Advances in neural information processing systems*, 35:24824–24837.

Laura Weidinger, John Mellor, Maribeth Rauh, Conor Griffin, Jonathan Uesato, Po-Sen Huang, Myra Cheng, Mia Glaese, Borja Balle, Atoosa Kasirzadeh, et al. 2021. Ethical and social risks of harm from language models. *arXiv preprint arXiv:2112.04359*.

Noam Wies, Yoav Levine, and Amnon Shashua. 2024. The learnability of in-context learning. *Advances in Neural Information Processing Systems*, 36.

Zhengxuan Wu, Atticus Geiger, Thomas Icard, Christopher Potts, and Noah Goodman. 2024. Interpretability at scale: Identifying causal mechanisms in alpaca. *Advances in Neural Information Processing Systems*, 36.

Sang Michael Xie, Aditi Raghunathan, Percy Liang, and Tengyu Ma. 2022. An explanation of in-context learning as implicit bayesian inference. In *International Conference on Learning Representations*.

Jieyu Zhao, Tianlu Wang, Mark Yatskar, Vicente Ordonez, and Kai-Wei Chang. 2018. Gender bias in coreference resolution: Evaluation and debiasing methods. In *Proceedings of the 2018 Conference of the North American Chapter of the Association for Computational Linguistics: Human Language Technologies*, volume 2.

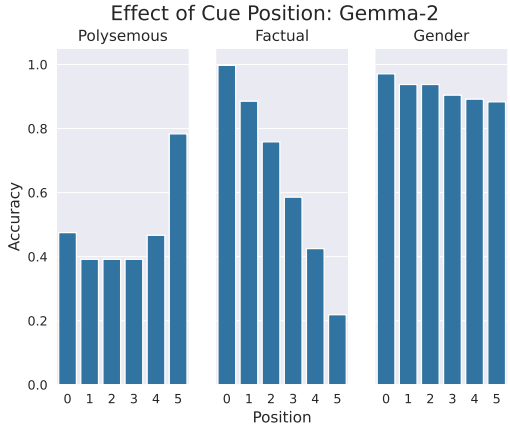

Figure 8: Performance on the 5-distractors dataset, disagreggated by the position of the cue amongst distractor sentences.

A Serial Position Effects

(Liu et al., 2024) show serial position effects—multi-document question-answering performance is highest when the relevant document is presented at the beginning or at the end of a set of documents, and lowest when it occurs in the middle. However, their work studies contexts that contain thousands of tokens, whereas our longest questions are on the order of 100-200 tokens. Interestingly, we also observe sensitivity to the position of relevant information: Fig. 8 presents results from our five-distractor datasets, indexed by the position of the cue amongst the distractors. The model often performs better when the cue occurs before the distractors, and sometimes (i.e., in the Polysemous dataset) demonstrates the U-shaped curve found in (Liu et al., 2024). See § G for a replication using Llama-2.

B Dataset Partitions

After establishing a contextualization failure mode in § 3, we study a partition (i.e., a fixed number of distractors, and position of the cue amongst distractors) of each dataset. We choose a partition such that model accuracy is as close to 50% as possible. This enables us to detect both the positive and negative impact of our different causal interventions. We enumerate the details of our partitions for both Gemma-2 and Llama-2 in Table 1.

C Disaggregated Logit Lens Results

In this section, we disaggregate the results presented in Fig. 4. In Fig. 9 we visualize the logit difference between the "yes" and "no" token generated by applying the logit lens (i.e., the final lay-

Model	Dataset	Distractor Count	Cue Index	Accuracy
Gemma	Polysemous	3	2	50.00%
Gemma	Factual	1	1	48.99%
Gemma	Gender	1	0	85.00%
Llama-2	Polysemous	5	0	50.83%
Llama-2	Factual	1	0	34.82%
Llama-2	Gender	5	3	57.92%

Table 1: Details for each dataset partition used for mechanistic analysis.

ernorm and unembedding matrix) to intermediate representations of the final token in the question prompt. We visualize these results separately for questions that the model ultimately answers correctly and incorrectly. We also disaggregate by the ground truth label associated with the question; in Fig. 9 (left) we visualize "yes" questions and in Fig. 9 (right) we visualize "no" questions.

We observe several trends in these results. First, we see a clear bias towards "no" in the first half of layers, followed by a clear bias towards "yes" in the second half of layers. Thus, one could not necessarily early decode the model's ultimate answer until the last several layers of the LLM.

Crucially, however, we see a clear effect of contextualization between the success and failure cases starting around the midpoint of the model. Around these layers, the model's ultimate answer becomes recoverable from the magnitude of the logit differences between "yes" and "no' — though the logit difference is uniformly positive from layers 20-30 (or beyond), the magnitude of that difference is larger when the model will ultimately respond "yes", and smaller when it will respond "no".

D Autoscoring Details

In § 4.2, we generated natural language descriptions of the content in intermediate representations of polysemous words using open-ended patchscopes. We follow prior work (Ghandeharioun et al., 2024b) in using another LLM (`PaLM2 text-unicorn@001`) to autoscore these generations. We prompt Palm2 with the following template:

Consider the following description: `{GENERATION}` *Is this description referring to* `{CORRECT_SENSE, INCORRECT_SENSE}`*? Please answer with yes or no:*

where `GENERATION` refers to the output of the open-ended patchscope at a give layer, `CORRECT_SENSE` refers to the sense implied by the cue, and `INCORRECT_SENSE` refers to the alternative.

Similarly to how we assess model performance in terms of question pairs, we only consider `GENERATION` to have exhibited the correct word sense if the autoscorer replies "yes" when prompted with `CORRECT_SENSE` and "no" when prompted with `INCORRECT_SENSE`.

E Qualitative Examples of Open-Ended Patchscopes

In this section, we provide example generations of the open-ended patchscope described in § 4.2. In these cases, there is clear contextualization of the subject entity over the course of layers. Examples are presented in Table 2.

F Pairwise Patching Results

In this section, we disaggregate the backpatching results from § 4.2 for Gemma-2 on the polysemous word dataset – displaying the accuracy of backpatching particular pairs of layers in Fig. 10. First, we note that many layer pairs achieve modest performance gains over the baseline. However, backpatching late representations into the earliest layers regularly harms performance, especially when freezing these representations.

We note that no individual pair of layers achieves the 30+% performance improvement achieved when searching over layer pairs for every question pair (as displayed in Fig. 7). This strengthens our intuition that different questions might require backpatching using different pairs of layers.

Finally, we note that backpatching and frozen backpatching provide very similar results for the majority of layer pairs, especially after layer 10. However, frozen backpatching underperforms when patching in layers 2 and 4. This indicates that the extra processing of representations extracted from layers 21-29 can increase performance over the baseline more substantially.

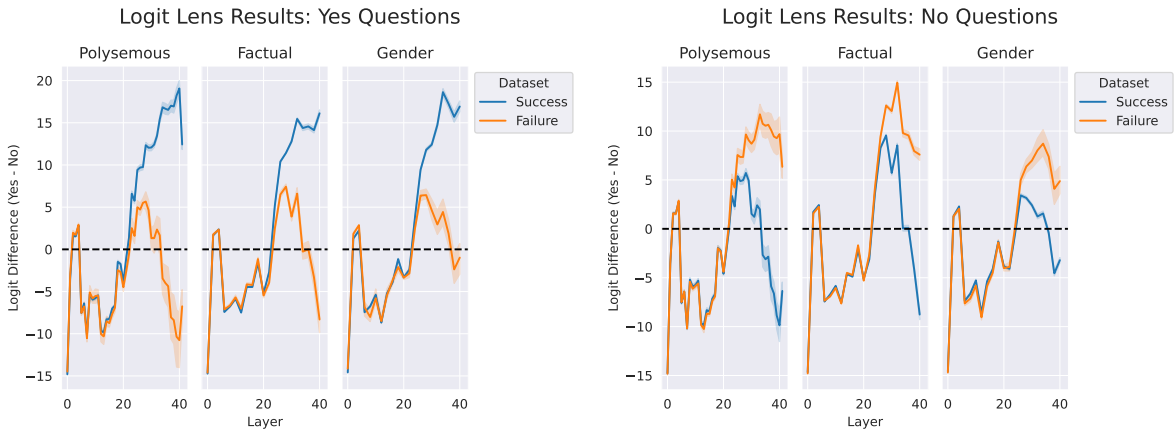

Figure 9: Disaggregated logit lens results.

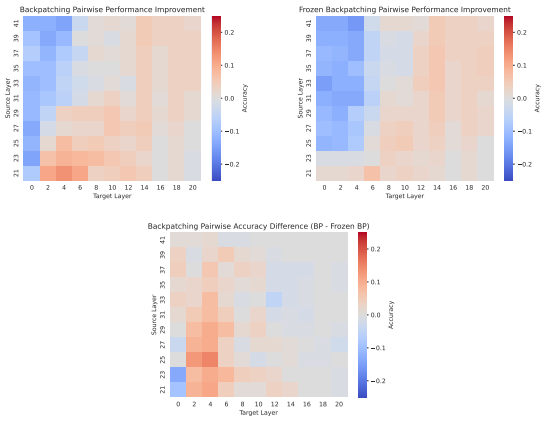

Figure 10: Pairwise Patching Results for Gemma-2 on the polysemous word dataset, measured as a performance difference from the baseline. (Top Left) Standard Backpatching Pairwise Results. (Top Right) Frozen Backpatching Results. (Bottom) Difference between the two heatmaps. Overall, we see largely similar results between backpatching and frozen backpatching.

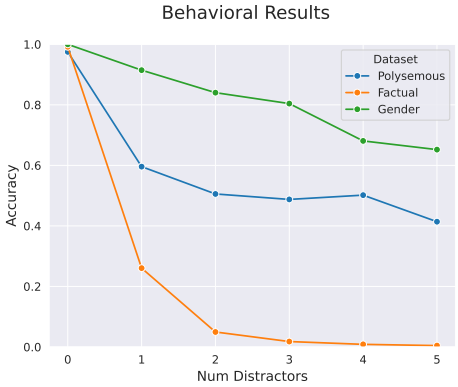

Figure 11: Accuracy vs. number of distractors in Llama-2. We find that injecting distractor text into the prompt causes sharp performance degredations across all datasets.

G Replicating results on Llama-2

In this section, we replicate the behavioral analysis found in § 3 and causal interventions found in § 4.1 and 4.2 on `Llama-2-13b-chat-hf` (Touvron et al., 2023). In Fig. 11 and Fig. 12, we find the same trends as observed in § 3: performance degrades as the number of distractors increase, and performance depends on cue position. However, we observe an even more reliable U-shaped curve with cue position using Llama-2.

For both interventions, we closely replicate the results found in Gemma-2. Fig. 13 replicates the attention ablation intervention discussed in § 4.1, and Fig. 14 replicates the patching results from § 4.2. This replication provides confidence that LLM Race Conditions can be found in a variety of LLMs, and that it can be addressed using similar techniques across LLMs.

Word	Sense	Layer	Interpretation
Bow	Ribbon	0	Sure! In this context, the word refers to a **bow and arrow**, a weapon that has been used for hunting and
Bow	Ribbon	6	Sure! In this context, the word refers to a **bow and arrow**, a weapon that has been used for centuries.
Bow	Ribbon	8	Sure! In this context, the word refers to a **decorative knot** tied in a ribbon or similar material.
Club	Building	2	Sure! In this context, the word refers to **Club Penguin**, a massively multiplayer online game (MMOG) that was
Club	Building	6	Sure! In this context, the word refers to a **social club**.
Club	Building	8	Sure! In this context, the word refers to a **nightclub**, a place where people go to dance, socialize,
Spring	Source of Water	0	Sure! In this context, the word refers to the **Spring Framework**, a popular open-source Java framework for building enterprise
Spring	Source of Water	6	Sure! In this context, the word refers to the season of spring, not a natural water source. **Spring
Spring	Source of Water	8	Sure! In this context, the word refers to a natural source of water that flows out of the ground. Here

Table 2: A sample of open-ended patchscope generations.

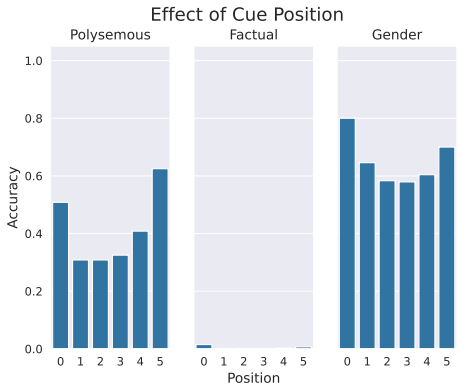

Figure 12: Performance on the 5-distractors dataset in Llama-2, disagreggated by the position of the cue amongst distractor sentences. We reliably find evidence of a U-shaped performance curve.

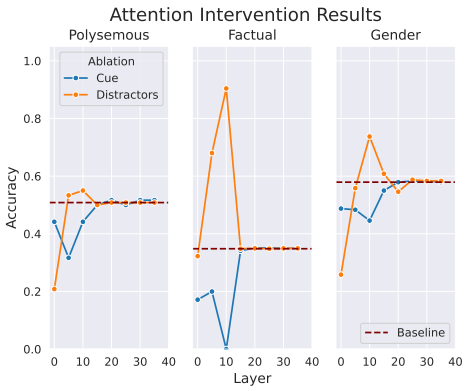

Figure 13: Replicating the attention ablating intervention in Llama-2. We find very similar trends to those found in Gemma-2.

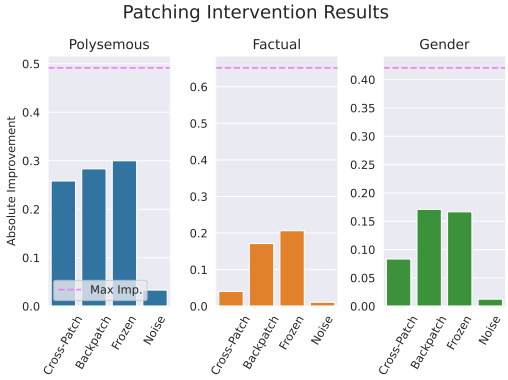

Figure 14: Replicating the patching interventions in Llama-2. We find very similar trends to those found in Gemma-2.