Language Models use Lookbacks to Track Beliefs

Anonymous authors
Paper under double-blind review

Abstract

How do language models (LMs) represent characters' beliefs, especially when those beliefs may differ from reality? This question lies at the heart of understanding the Theory of Mind (ToM) capabilities of LMs. We analyze Llama-3-70B-Instruct's ability to reason about characters' beliefs using causal mediation and abstraction. We construct a dataset that consists of simple stories where two characters each separately change the state of two objects, potentially unaware of each other's actions. Our investigation uncovered a pervasive algorithmic pattern that we call a *lookback mechanism*, which enables the LM to recall important information when it becomes necessary. The LM binds each character-object-state triple together by co-locating reference information about them, represented as their Ordering IDs (OIDs) in low rank subspaces of the state token's residual stream. When asked about a character's beliefs regarding the state of an object, the *binding lookback* retrieves the corresponding state OID and then an *answer lookback* retrieves the state token. When we introduce text specifying that one character is (not) visible to the other, we find that the LM first generates a *visibility ID* encoding the relation between the observing and the observed character OIDs. In a *visibility lookback*, this ID is used to retrieve information about the observed character and update the observing character's beliefs. Our work provides insights into the LM's beliefs tracking mechanism, taking a step toward reverse-engineering ToM reasoning in LMs.

1 Introduction

The ability to infer mental states of others—known as Theory of Mind (ToM)—is an essential aspect of social and collective intelligence (Premack & Woodruff, 1978; Riedl et al., 2021). Recent studies have established that language models (LMs) can solve some tasks requiring ToM reasoning (Street et al., 2024; Strachan et al., 2024a; Kosinski, 2024), while others have argued against it (Sclar et al., 2025; Shapira et al., 2024; Kim et al., 2023a, *inter alia*). However, existing works primarily involve behavioral assessments, which do not reveal the internal mechanisms by which LMs encode and manipulate representations of mental states to solve (or fail to solve) such tasks (Hu et al., 2025).

In this work, we investigate how LMs represent and update characters' beliefs, which is a fundamental element of ToM (Dennett, 1981; Wimmer & Perner, 1983). For instance, the Sally-Anne test (Baron-Cohen et al., 1985), a canonical measure of ToM in humans, evaluates these abilities by requiring participants to track Sally's belief, which diverges from reality due to missing information, and Anne's belief, which updates based on new observations.

We construct *CausalToM*, a dataset of simple stories involving two characters, each interacting with an object to change its state, with the possibility of observing one another. We then analyze the internal mechanisms that enable Llama-3-70B-Instruct (Grattafiori et al., 2024) to reason about and answer questions regarding the characters' beliefs about the state of each object. For a sample story, see Section 3 and for the full prompt refer to Appendix A.

During our investigation of the underlying mechanism responsible for belief tracking, we discover a pervasive mechanism that performs multiple subtasks, which we refer to as the *Lookback Mechanism*. This mechanism enables the model to recall important

information only when it becomes necessary. In a lookback mechanism, two copies of a single piece of information are transferred to two distinct tokens. In case needed, this allows the attention heads at the latter token to look back at the earlier one and retrieve vital information stored there, rather than transferring that information directly (see Fig. 1).

We identified three key lookback mechanisms that collectively perform belief tracking: 1) *Binding Lookback* (Fig. 3(a)): First the LM assigns *Ordering IDs* (OIDs) (Dai et al., 2024) that encode whether a character, object, or state token appears first or second. Then, the character and object OIDs are copied to low-rank subspaces of the corresponding state token and the final token residual stream. Later, went the LM needs to answer a question about a character's beliefs, it uses this information to retrieve the answer state OID. 2) *Answer Lookback* (Fig. 3(b)): Uses the answer state OID from the binding lookback to retrieve the answer state token value. 3) *Visibility Lookback* (Fig. 7): When an explicit visibility condition between characters is mentioned, the model employs additional reference information called the *Visibility ID* to retrieve information about the observed character, augmenting the observing character's awareness.

Overall, this work not only advances our understanding of the internal computations in LMs that enable ToM abilities but also uncovers a pervasive mechanism that serves as the foundation for executing complex, condition-based logical reasoning.

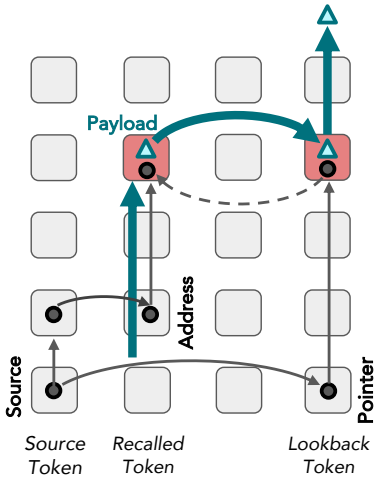

Figure 1: **The lookback mechanism** is used to perform conditional reasoning; The *source token* contains information that is copied into two instances via attention to create a *pointer* and an *address*. Alongside the address in the residual stream is a *payload* information. If necessary, the model can retrieve the payload by dereferencing the pointer. The solid lines are movement via residual connections or attention heads, while the dotted line indicates the attention "looking back" from pointer to address.

2 The Lookback Mechanism

Our investigations of belief tracking uncover a recurring pattern of computation that we call *lookback*.[1] We give here a brief overview of this mechanism; subsequent sections provide detailed experiments and analyses. In lookback, *source information* is copied via attention into an *address* copy in the residual stream of a *recalled token* and a *pointer* copy in the residual stream of a *lookback token* that occurs later in the text. The LM places the address alongside a *payload* of the recalled token's residual stream that can be brought forward to the lookback token via attention if necessary. Fig. 3 schematically describes a general lookback.

That is, the LM can use attention to dereference the pointer and retrieve the payload present in the residual stream of the recalled token (that might contain aggregated information from previous tokens), bringing it to the residual stream of the lookback token. Specifically, the pointer at the lookback token forms an attention query vector, while the address at the recalled token forms a key vector. Because the pointer and the address are copies of the same source information, they would have a high dot-product, hence a *QK-circuit* (Elhage et al., 2021) is established forming a bridge from the lookback token to the recalled token. The LM uses this bridge to move the payload that contains information needed to complete the subtask through the *OV-circuit*.

[1]Although this mechanism may resemble *induction heads* (Elhage et al., 2021; Olsson et al., 2022), they differ fundamentally. In induction heads, information from a previous token occurrence is passed only to the subsequent token through K-composition, without being duplicated to its next occurrence. In contrast, the lookback mechanism copies the same information not only to the location where the vital information resides but also to the target location that needs to retrieve that information.

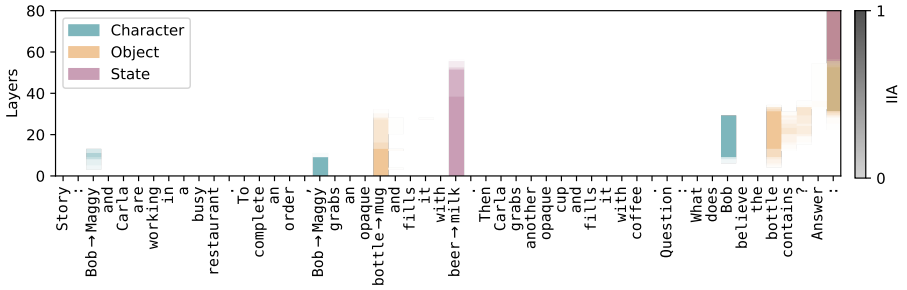

Figure 2: **Tracing Information flow** of crucial input tokens using causal mediation analysis.

To develop an intuition for why an LM would learn to implement lookback mechanisms to solve reasoning tasks such as our belief tracking task, consider that during training, LMs process text with no awareness of what might come next. Then, it would be useful to locate addresses alongside payloads that might be useful for downstream tasks. In our setting, the LM constructs a representation of the story without any knowledge of what questions it may be asked about, so the LM concentrates pieces of information in the residual stream of certain tokens that later become payloads and addresses. When the question text is reached, pointers are constructed that reference this crucial story information and dereference it as the answer to the question.

3 Preliminaries

Dataset

Existing datasets for evaluating ToM capabilities of LMs are designed for behavioral testing and lack the ability to construct counterfactual pairs needed for causal analysis (Kim & Sundar, 2012). To address this, we constructed *CausalToM*, a structured dataset of simple stories, where each story involves two characters, each interacting with a distinct object causing the object to take a unique state. For example: "`Character1 and Character2 are working in a busy restaurant. To complete an order, Character1 grabs an opaque Object1 and fills it with State1. Then Character2 grabs another opaque Object2 and fills it with State2.`" We then ask the LM to reason about one of the characters' beliefs regarding the state of an object: "`What does Character1 believe Object2 contains?`" We analyze the LM's ability to track characters' beliefs in two distinct settings. (1) *No Visibility*, where both characters are unaware of each other's actions, and (2) *Explicit Visibility* where explicit information about whether a character can/cannot observe the other's actions is provided, e.g., "`Bob can observe Carla's actions. Carla cannot observe Bob's actions.`" We also provide general task instructions (e.g., use `unknown` to answer no awareness cases); refer to Appendix A, B for the full prompt and additional dataset details. Our experiments analyze the Llama-3-70B-Instruct model in half-precision, using *NNsight* (Fiotto-Kaufman et al., 2025). The model demonstrates a high behavioral performance on both the no-visibility and explicit-visibility settings, achieving accuracy of 95.7% and 99% respectively. For all subsequent experiments, we filter out samples that the model fails to answer correctly.

Causal Mediation Analysis

Our goal is to develop a mechanistic understanding of how Llama-3-70B-Instruct reasons about characters' beliefs and answers related questions (Saphra & Wiegreffe, 2024). A key method for conducting causal analysis is *Interchange Interventions* (Vig et al., 2020; Geiger et al., 2020; Finlayson et al., 2021), in which the LM is run on paired examples: an *original input* **o** and a *counterfactual input* **c** and certain internal activations in the LM run on the original are replaced with those computed from the counterfactual. The effect of these interventions is quantified using *interchange intervention accuracy* (IIA), which measures the proportion of instances where the intervened output matches an *expected output*.

Drawing inspiration from existing literature (Vig et al., 2020; Meng et al., 2022; Wang et al., 2023), we begin our analysis by performing interchange interventions with counterfactuals

that are identical to the original except for key input tokens. We trace the causal path from these key tokens to the final output. This is a type of *Causal Mediation Analysis*. Specifically, we construct a counterfactual dataset where **o** contains a question about the belief of a character not mentioned in the story, while **c** is identical except that the story includes the queried character. The expected outcome of this intervention is a change in the final output of **o** from *unknown* to a state token, such as **beer**. We conduct similar interchange interventions for object and state tokens. Refer to Appendix E for more details.

Figure 2 presents the aggregated results of this experiment for the key input tokens **Character1**, **Object1**, and **State1**. The cells are color-coded to indicate IIA. Even at this coarse level of analysis, several significant insights emerge: 1) Information from the correct state token (**beer**) flows directly from its residual stream to that of the final token in later layers, consistent with prior findings (Lieberum et al., 2023; Prakash et al., 2024); 2) Information associated with the query character and the query object is retrieved from their earlier occurrences and passed to the final token before being replaced by the correct state token.

Desiderata Based Patching via Causal Abstraction

The causal mediation experiments provide a coarse-grained analysis of how information flows from an input token to the output, but does not identify what that information is. A fact about transformers is that the input to the first layer contains input tokens and the output from the final layer contains the output token, but what is the information content of the causal path in between the input and output?

To answer this question, we turn to *Causal Abstraction* (Geiger et al., 2021; 2024). We align the variables of a high-level causal model with the LM's internal activations and verify the alignment by conducting targeted interchange interventions for each variable. Specifically, we perform aligned interchange interventions at both levels: high-level interventions that target causal variables and low-level interventions that modify features of the LM's hidden vectors. If the LM produces the same output as the high-level causal model under these aligned interventions, it provides evidence supporting the hypothesized causal model. Refer to Appendix C for more details about the high-level causal model.

In addition to performing interchange interventions on entire residual stream vectors in LMs, we also intervene on specific subspaces to further localize causal variables. To identify the subspace encoding a particular variable, we employ the *Desiderata-based Component Masking* (De Cao et al., 2020; Davies et al., 2023; Prakash et al., 2024) technique, which learns a sparse binary mask over the internal activation space by maximizing the logit of the expected output token. Specifically, we train a mask to select the singular vectors in the activation space that encode a high-level variable. For further details, refer to Appendix F.

4 Belief Tracking via Ordering IDs and Lookback Mechanisms

When presented with belief tracking tasks where characters have no visibility of each other, the LM solves the task using three key mechanisms: *ordering ID assignment*, *binding lookback*, and *answer lookback*, which are described in detail in the following subsections and summarized as psuedocode in the Appendix D.

4.1 Ordering ID Assignment

LM processes input tokens by assigning an *Ordering ID* (OID) to each crucial token, including character, object, and state tokens (Dai et al., 2024). These OIDs, encoded in a low-rank subspace of the internal activation, serve as a reference that indicates whether an entity is the first or second of its type, regardless of its token value. For example, in Fig. 3, **Bob** is assigned the first character OID, while **Carla** receives the second character OID. We validate the presence of OIDs through multiple experiments, where intervening on tokens with identical token values but different OIDs alters the model's internal computation, leading to changes in the final output, in the subsequent subsections and Appendix G & H. The same process applies to object and state tokens. The model then uses these OIDs as fundamental units of analysis, feeding them into lookback mechanisms that perform logical operations.

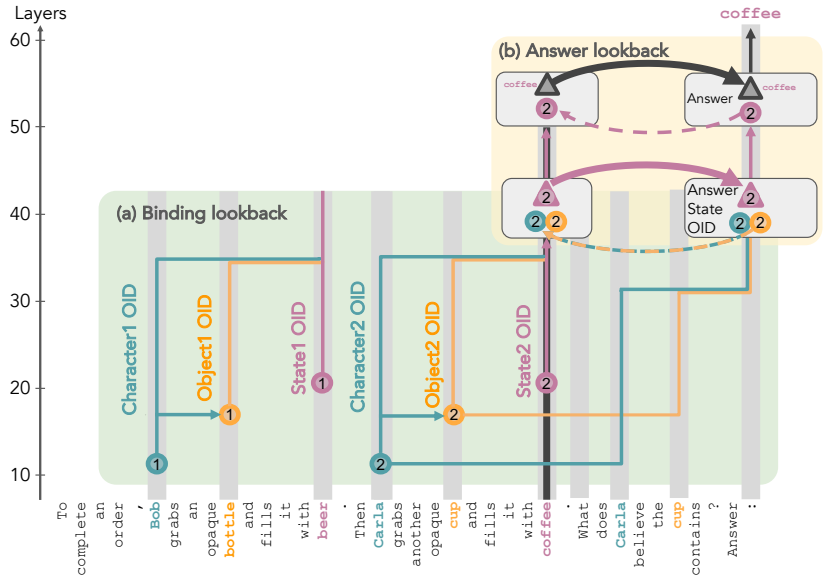

Figure 3: **Belief Tracking with *no visibility* between characters.** The LM assigns **ordering IDs** (OIDs) to each character, object, and state that encode their order of appearance. **(a) Binding lookback.** *Address* copies of character and object OIDs are placed alongside the state OID *payload* in the residual stream of state tokens while *pointer* copies are moved to the final token residual stream. The pointers are dereferenced, bringing the state OID into the final token residual stream. **(b) Answer lookback.** An *address* copy of the state OID is alongside the state token *payload* in the residual stream of state tokens while a *pointer* copy is moved to the final token residual stream via binding lookback. The pointer is dereferenced, bringing the answer state token payload into the final token residual stream.

4.2 Uncovering the Binding Lookback Mechanism

The *Binding lookback* is the first operation applied to these OIDs. The character and object OIDs, serving as the source information, are duplicated into two instances each. One copy, referred to as the *address*, is placed in the residual stream of the state token (*recalled token*), alongside the state OID as the payload to transfer. The other copy, known as the *pointer*, is moved in the residual stream of the final token (*lookback token*). These pointer and address copies are then used to form the QK-circuit at the lookback token, which dereferences the state OID payload, transferring it from the state token to the final token. See Fig.3 for a schematic of this lookback and see Fig.1 for the general mechanism.

Localizing the Address and Payload In our first experiment, we localize the address copies of the character and object OIDs and the state OID payload to the residual stream of the state token (recalled token), as illustrated in Fig. 3. We construct a counterfactual dataset where each example consists of a original input **o** with an answer that isn't *unknown* and a counterfactual input **c** where the character, object, and state tokens are identical, except the ordering of the two sentences is swapped while the question remains unchanged, as illustrated in Fig. 4.2. The expected outcome predicted by our high-level causal model under intervention is the other state token from the original example, because reversing the address and payload values without changing the pointer flips the output. In the low-level LM, the QK-circuit, formed using the pointer at the lookback token, attends to the other state token and retrieves its state OID as the payload.

We perform an interchange intervention experiment layer-by-layer, where we replace the residual stream vectors at the first state token in the original run with that of the second state token in the counterfactual run and vice versa for the other state token. It is important to note that if the intervention targets state token values instead of their OIDs, it should not produce the expected output.

As shown in Fig. 4.2, the strongest alignment occurs between layers 33 and 38, confirming our hypothesis that the state token's residual stream contains both the address information

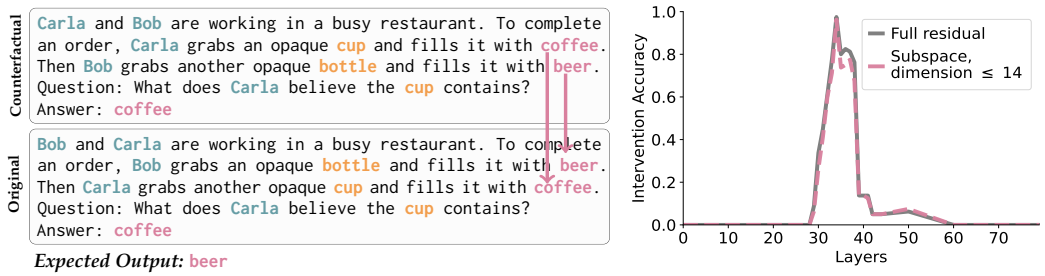

Figure 4: **Payload and address** of Binding lookback: We perform interchange interventions on the residual stream vectors of the state tokens, one layer at a time.

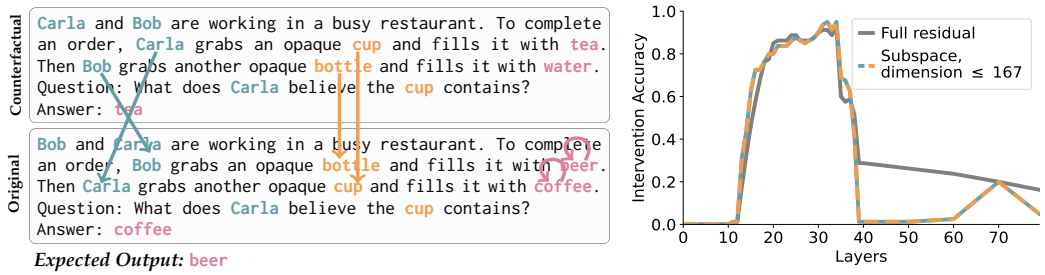

Figure 5: **Source Information** of Binding lookback: We perform interchange interventions on the residual stream vectors up to a given layer (represented by the x-axis) at the character and object tokens, while keeping all residual vectors of the state token frozen.

(character and object OIDs) and the payload information (state OID). These components are subsequently used to form the appropriate QK and OV circuits.

Localizing the Source Information Shown in Fig. 3, the source information is copied as both the address and the pointer at different token positions. As such, to localize the source information, we conduct two intervention experiments: 1) interchanging the residual stream vectors of the source tokens (characters and objects) and 2) interchanging the source tokens while freezing the residual stream of the recalled tokens (state), which contain the address.

We generate a dataset where the counterfactual example, **c**, swaps the order of the characters and objects and replaces the state tokens with entirely new ones, while keeping the question the same as in **o**. In the high-level causal model, the expected outcome for the first experiment is the same token, e.g., coffee, because the address and the pointer are both flipped, resulting in no change. The expected outcome of the second experiment is the other state token, e.g., beer in Fig. 5. In the low-level LM, when neither the address nor the pointer is frozen, both are updated through the intervention, causing the QK-circuit at the lookback token to attend to the same state token and retrieve its state OID as the payload.

As shown in Fig. 5, we observe alignment in the second experiment between layers $20 - 34$, indicating that the source information, specifically the character and object OIDs, is present in their respective token residual streams between these layers. As expected, no alignment is observed in the first experiment, as illustrated in Fig. 13. These results not only confirm the presence of source information but also establish its transfer to the recalled and lookback tokens as addresses and pointers, respectively. We provide more experimental results in Appendix G on localizing character and object OIDs separately.

Localizing the Pointers The pointer copies of the character and object OID are first formed at the character and object tokens in the question before being moved again to the final token for dereferencing. Refer to Appendix H for experiments and more details.

4.3 Uncovering the Answer Lookback Mechanism

The LM answers the question using the *Answer Lookback*. The state OID of the correct answer serves as the source information, which is copied into two instances. One instance, the address copy of the state OID, is in the residual stream of the state token (the recalled token)

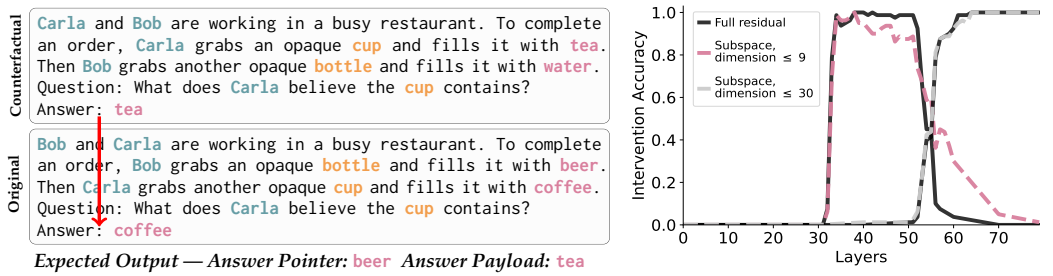

Figure 6: **Answer Lookback Pointer and Payload**: The state OID payload of the binding lookback is the pointer for the answer lookback. We perform interchange interventions on the residual stream of the final token with two expected outputs. The expected output for the pointer is the other state token in the original, whereas the expected output for the payload is the correct state token of the counterfactual.

with the state token itself as the payload. The other instance, the pointer copy of the state OID, is transferred to the residual stream of the final token as the binding lookback payload. This pointer is then dereferenced, bringing the state token payload into the residual stream of the final token. See Fig. 3 for the answer lookback and Fig. 1 for the general mechanism.

Localizing the Pointers We first localize the pointer of the answer lookback, which is the payload of the binding lookback. To achieve this, we conduct another interchange intervention experiment where the residual vectors at the final token position in the original run are replaced with those from the counterfactual run, one layer at a time. The counterfactual inputs have swapped objects and characters and randomly sampled states. If the answer pointer is targeted in the high-level causal model, the expected output is the other state in the original input, e.g., **beer**. As shown in Fig. 6, alignment begins at layer 34, indicating that this layer contains the pointer information, in low-rank subspace, used to retrieve the correct state token as the payload, which remains causally relevant until layer 52.

Localizing the Payload To determine where the model uses the correct state OID pointer to retrieve the correct state token, we use the same interchange intervention experiment. However, in this case, the expected output is the correct state token from the counterfactual example, rather than the state token from the original example, as illustrated in Fig. 6. The alignment occurs after layer 56, indicating that the model retrieves the correct state token (payload) into the final token's residual stream at layer 56 and beyond, where it is subsequently used to generate the final output.

5 Impact of Visibility Conditions on Belief Tracking Mechanism

In the previous section, we demonstrated how the LM uses ordering IDs and two lookback mechanisms to track the beliefs of characters that cannot observe each other. Now, we explore how the LM updates the beliefs of characters when provided with additional information—that one of the characters (*observing*) can observe the actions of others (*observed*). We hypothesize that the LM employs another lookback mechanism, which we refer to as the *Visibility Lookback*, to incorporate information about the observed character.

As illustrated in Fig. 7, we hypothesize that the LM first generates a *Visibility ID* at the residual stream of the visibility sentence, serving as the source information. The address copy of the visibility ID remains in the residual stream of the visibility sentence, while its pointer copy gets transferred to the residual streams of the question tokens, which are the lookback tokens. Then LM forms a QK-circuit at the lookback token and dereferences the visibility ID pointer to bring forward the payload.

Although we were unable to determine the exact semantics of the payload in this lookback, we speculate that it represents the character OID of the observed character from the visibility sentence. We propose the existence of another lookback, where the story sentence associated with the observed character serves as the source, and its payload encodes information about the observed character. This information is then retrieved by the lookback tokens of the Visibility lookback, with the payload also containing the observed character's OID,

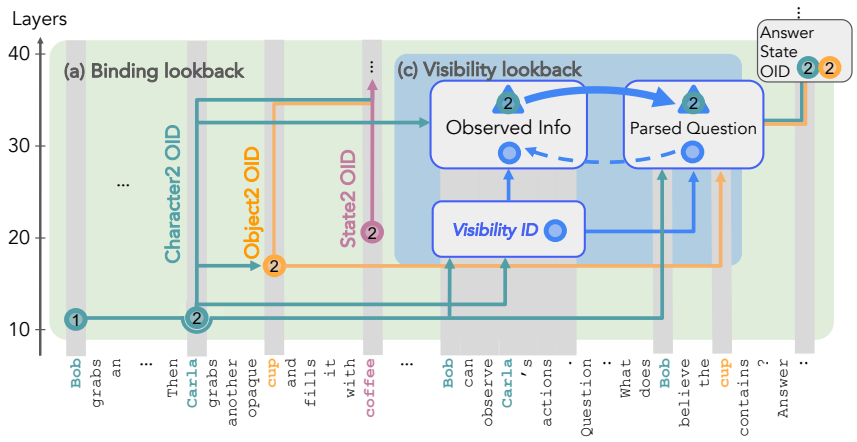

Figure 7: **Visbility Lookback** When one character (the observing character) can see another (the observed character), the LM assigns a visibility ID to the visibility sentence. An address copy of this visibility ID remains in the visibility sentence's residual stream. A pointer copy of the visibility ID is transferred to the question's residual stream (lookback tokens). During processing, the model dereferences this pointer through a QK-circuit, bringing forward the payload. Based on initial evidence, this payload contains the observed character's OID. Refer to Appendix I for more details. This mechanism allows the model to incorporate the observed character's knowledge into the observing character's belief state, enabling more complex belief reasoning.

which contributes to the queried character's enhanced awareness. For more details on the speculated lookback, please refer to Appendix I.

5.1 Uncovering the Visibility Lookback Mechanism

Localizing the Source Information To localize the source information, we conduct an interchange intervention experiment using the same story sentences but with different state tokens and visibility information. In the original example **o**, the first character cannot observe the second character's actions, whereas in the counterfactual example **c**, the first character can observe them, as illustrated in Fig. 8. The interchange intervention is executed on visibility sentence tokens by replacing their residual vectors in the original run with those from the counterfactual run. The expected outcome of this intervention is a change in the final output of the original run from "unknown" to the state token associated with the queried object. As shown in Fig. 8 (— line), alignment occurs between layers 10 and 23, indicating that the visibility ID remains encoded in the visibility sentence until layer 23, after which it is transferred to subsequent tokens.

Localizing the Payload To localize the payload information, we use the same counterfactual dataset. However, instead of intervening on the source or recalled tokens, we intervene on the lookback tokens, specifically the question and answer tokens. As in the previous experiment, we replace the residual vectors of these tokens in the original run with those from the counterfactual run. As shown in Fig. 8 (— line), alignment occurs only after layer 31, indicating that the information enhancing the queried character's awareness is present in the lookback tokens only after this layer.

Localizing the Address and Pointer The previous two experiments suggest the presence of a lookback mechanism, as there is no signal indicating that the source or payload have been formed between layers 24 and 31. We hypothesize that this lack of signal is due to a mismatch between the address and pointer information at the recalled and lookback tokens. Specifically, when intervening only on the recalled token after layer 25, the pointer is not updated, whereas intervening only on the lookback tokens leaves the address unaltered, leading to the mismatch. To test this hypothesis, we conduct another intervention using the same counterfactual dataset, but this time, we intervene on the residual vectors of both the recalled and lookback tokens, i.e., the visibility sentence, as well as the question and answer tokens. As shown in Fig. 8 (— line), alignment occurs after layer 10 and remains

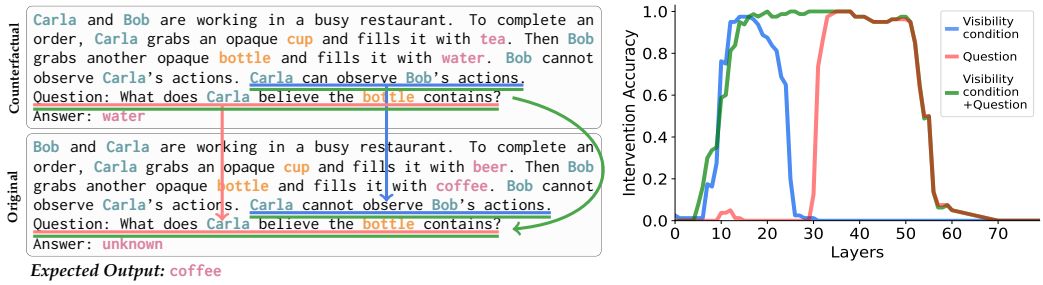

Figure 8: **Visibility Lookback.** The interventions on the visibility sentence are successful until the visibility ID source information is split into two copies and the interventions on the question sentences are successful once the payload of the observed character OID is brought over. The gap in layers where interventions on the visibility sentence stop working and interventions on the question sentence start working is exactly where the visibility lookback is hypothesized to take place.

stable, supporting our hypothesis. This intervention replaces both the address and pointer copies of the visibility IDs, enabling the LM to form a QK-circuit and retrieve the payload.

6 Related Work

Theory of mind in LMs A large body of work has focused on benchmarking different aspects of ToM through various tasks that attempt to assess LMs' performance such as Le et al. (2019); Xu et al. (2024); Shapira et al. (2023); Jin et al. (2024); Wu et al. (2023); Kim et al. (2023b); Chan et al. (2024); Strachan et al. (2024b) and many more. In addition, there are various methods tailored to improve ToM ability in LMs through prompting (e.g., Sclar et al., 2023; Zhou et al., 2023; Wilf et al., 2024; Moghaddam & Honey, 2023; Hou et al., 2024). Only a few works relate to counterfactual inputs needed for causal analysis (Gandhi et al., 2024; Shapira et al., 2024).

Entity tracking in LMs Entity tracking and variable binding are crucial abilities for LMs to exhibit not only coherent ToM capabilities, but also neurosymbolic reasoning. Many existing works have attempted to decipher this ability in LMs (Li et al., 2021; Davies et al., 2023; Kim & Schuster, 2023; Prakash et al., 2024; Feng & Steinhardt, 2023; Feng et al., 2024; Dai et al., 2024). Our work builds on their empirical insights and extends the current understanding of how LMs bind various entities defined in context.

Mechanistic interpretability of theory of mind Only a few empirical studies explored the underlying mechanisms of ToM of LM (Zhu et al., 2024; Bortoletto et al., 2024) (Herrmann & Levinstein, 2024, is a notable theoretical paper). Those studies focus on probing techniques (Belinkov, 2022; Alain, 2016) to identify internal representations of beliefs and used steering techniques (Li et al., 2024; Rimsky et al., 2023) to improve LM performance by manipulating their activations. However, the mechanism by which LMs solve those tasks remains a black box, limiting our ability to understand, predict, and control LMs' behaviors.

7 Conclusion

Through a series of desiderata-based patching experiments, we have mapped the mechanisms underlying the processing of partial knowledge and false beliefs in a set of simple stories. We are struck by the pervasive appearance of a single recurring computational pattern: the lookback, which resembles a pointer dereference inside a transformer. The LMs use a combination of several lookbacks to reason about nontrivial visibility and belief states. Our improved understanding of these fundamental computations gives us optimism that it will be possible to fully reveal the algorithms underlying Theory of Mind in LMs.

References

Guillaume Alain. Understanding intermediate layers using linear classifier probes. *arXiv preprint arXiv:1610.01644*, 2016.

Simon Baron-Cohen, Alan M Leslie, and Uta Frith. Does the autistic child have a "theory of mind"? *Cognition*, 21(1):37–46, 1985.

Yonatan Belinkov. Probing classifiers: Promises, shortcomings, and advances. *Computational Linguistics*, 48(1):207–219, 2022.

Matteo Bortoletto, Constantin Ruhdorfer, Lei Shi, and Andreas Bulling. Benchmarking mental state representations in language models. *arXiv preprint arXiv:2406.17513*, 2024.

Chunkit Chan, Cheng Jiayang, Yauwai Yim, Zheye Deng, Wei Fan, Haoran Li, Xin Liu, Hongming Zhang, Weiqi Wang, and Yangqiu Song. Negotiationtom: A benchmark for stress-testing machine theory of mind on negotiation surrounding. *arXiv preprint arXiv:2404.13627*, 2024.

Qin Dai, Benjamin Heinzerling, and Kentaro Inui. Representational analysis of binding in language models. In Yaser Al-Onaizan, Mohit Bansal, and Yun-Nung Chen (eds.), *Proceedings of the 2024 Conference on Empirical Methods in Natural Language Processing*, pp. 17468–17493, Miami, Florida, USA, November 2024. Association for Computational Linguistics. doi: 10.18653/v1/2024.emnlp-main.967. URL https://aclanthology.org/2024.emnlp-main.967/.

Xander Davies, Max Nadeau, Nikhil Prakash, Tamar Rott Shaham, and David Bau. Discovering variable binding circuitry with desiderata, 2023. URL https://arxiv.org/abs/2307.03637.

Nicola De Cao, Michael Sejr Schlichtkrull, Wilker Aziz, and Ivan Titov. How do decisions emerge across layers in neural models? interpretation with differentiable masking. In *Proceedings of the 2020 Conference on Empirical Methods in Natural Language Processing (EMNLP)*, pp. 3243–3255, Online, November 2020. Association for Computational Linguistics. doi: 10.18653/v1/2020.emnlp-main.262. URL https://aclanthology.org/2020.emnlp-main.262.

Daniel Clement Dennett. *The Intentional Stance*. MIT Press, 1981.

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. *Transformer Circuits Thread*, 2021. https://transformer-circuits.pub/2021/framework/index.html.

Jiahai Feng and Jacob Steinhardt. How do language models bind entities in context? *arXiv preprint arXiv:2310.17191*, 2023.

Jiahai Feng, Stuart Russell, and Jacob Steinhardt. Monitoring latent world states in language models with propositional probes. *CoRR*, abs/2406.19501, 2024. doi: 10.48550/ARXIV.2406.19501. URL https://doi.org/10.48550/arXiv.2406.19501.

Matthew Finlayson, Aaron Mueller, Sebastian Gehrmann, Stuart M. Shieber, Tal Linzen, and Yonatan Belinkov. Causal analysis of syntactic agreement mechanisms in neural language models. In Chengqing Zong, Fei Xia, Wenjie Li, and Roberto Navigli (eds.), *Proceedings of the 59th Annual Meeting of the Association for Computational Linguistics and the 11th International Joint Conference on Natural Language Processing, ACL/IJCNLP 2021, (Volume 1: Long Papers), Virtual Event, August 1-6, 2021*, pp. 1828–1843. Association for Computational Linguistics, 2021. doi: 10.18653/V1/2021.ACL-LONG.144. URL https://doi.org/10.18653/v1/2021.acl-long.144.

Jaden Fried Fiotto-Kaufman, Alexander Russell Loftus, Eric Todd, Jannik Brinkmann, Koyena Pal, Dmitrii Troitskii, Michael Ripa, Adam Belfki, Can Rager, Caden Juang, Aaron Mueller, Samuel Marks, Arnab Sen Sharma, Francesca Lucchetti, Nikhil Prakash, Carla E. Brodley, Arjun Guha, Jonathan Bell, Byron C Wallace, and David Bau. NNsight and NDIF: Democratizing access to open-weight foundation model internals. In *The Thirteenth International Conference on Learning Representations*, 2025. URL https://openreview.net/forum?id=MxbEiFRf39.

Kanishk Gandhi, Jan-Philipp Fränken, Tobias Gerstenberg, and Noah Goodman. Understanding social reasoning in language models with language models. *Advances in Neural Information Processing Systems*, 36, 2024.

Atticus Geiger, Kyle Richardson, and Christopher Potts. Neural natural language inference models partially embed theories of lexical entailment and negation. In Afra Alishahi, Yonatan Belinkov, Grzegorz Chrupała, Dieuwke Hupkes, Yuval Pinter, and Hassan Sajjad (eds.), *Proceedings of the Third BlackboxNLP Workshop on Analyzing and Interpreting Neural Networks for NLP*, pp. 163–173, Online, November 2020. Association for Computational Linguistics. doi: 10.18653/v1/2020.blackboxnlp-1.16. URL https://aclanthology.org/2020.blackboxnlp-1.16.

Atticus Geiger, Hanson Lu, Thomas Icard, and Christopher Potts. Causal abstractions of neural networks. In Marc'Aurelio Ranzato, Alina Beygelzimer, Yann N. Dauphin, Percy Liang, and Jennifer Wortman Vaughan (eds.), *Advances in Neural Information Processing Systems 34: Annual Conference on Neural Information Processing Systems 2021, NeurIPS 2021, December 6-14, 2021, virtual*, pp. 9574–9586, 2021. URL https://proceedings.neurips.cc/paper/2021/hash/4f5c422f4d49a5a807eda27434231040-Abstract.html.

Atticus Geiger, Duligur Ibeling, Amir Zur, Maheep Chaudhary, Sonakshi Chauhan, Jing Huang, Aryaman Arora, Zhengxuan Wu, Noah Goodman, Christopher Potts, and Thomas Icard. Causal abstraction: A theoretical foundation for mechanistic interpretability, 2024. URL https://arxiv.org/abs/2301.04709.

Mor Geva, Jasmijn Bastings, Katja Filippova, and Amir Globerson. Dissecting recall of factual associations in auto-regressive language models, 2023. URL https://arxiv.org/abs/2304.14767.

Aaron Grattafiori, Abhimanyu Dubey, Abhinav Jauhri, Abhinav Pandey, Abhishek Kadian, Ahmad Al-Dahle, Aiesha Letman, Akhil Mathur, Alan Schelten, Alex Vaughan, Amy Yang, Angela Fan, Anirudh Goyal, Anthony Hartshorn, Aobo Yang, Archi Mitra, Archie Sravankumar, Artem Korenev, Arthur Hinsvark, Arun Rao, Aston Zhang, Aurelien Rodriguez, Austen Gregerson, Ava Spataru, Baptiste Roziere, Bethany Biron, Binh Tang, Bobbie Chern, Charlotte Caucheteux, Chaya Nayak, Chloe Bi, Chris Marra, Chris McConnell, Christian Keller, Christophe Touret, Chunyang Wu, Corinne Wong, Cristian Canton Ferrer, Cyrus Nikolaidis, Damien Allonsius, Daniel Song, Danielle Pintz, Danny Livshits, Danny Wyatt, David Esiobu, Dhruv Choudhary, Dhruv Mahajan, Diego Garcia-Olano, Diego Perino, Dieuwke Hupkes, Egor Lakomkin, Ehab AlBadawy, Elina Lobanova, Emily Dinan, Eric Michael Smith, Filip Radenovic, Francisco Guzmán, Frank Zhang, Gabriel Synnaeve, Gabrielle Lee, Georgia Lewis Anderson, Govind Thattai, Graeme Nail, Gregoire Mialon, Guan Pang, Guillem Cucurell, Hailey Nguyen, Hannah Korevaar, Hu Xu, Hugo Touvron, Iliyan Zarov, Imanol Arrieta Ibarra, Isabel Kloumann, Ishan Misra, Ivan Evtimov, Jack Zhang, Jade Copet, Jaewon Lee, Jan Geffert, Jana Vranes, Jason Park, Jay Mahadeokar, Jeet Shah, Jelmer van der Linde, Jennifer Billock, Jenny Hong, Jenya Lee, Jeremy Fu, Jianfeng Chi, Jianyu Huang, Jiawen Liu, Jie Wang, Jiecao Yu, Joanna Bitton, Joe Spisak, Jongsoo Park, Joseph Rocca, Joshua Johnstun, Joshua Saxe, Junteng Jia, Kalyan Vasuden Alwala, Karthik Prasad, Kartikeya Upasani, Kate Plawiak, Ke Li, Kenneth Heafield, Kevin Stone, Khalid El-Arini, Krithika Iyer, Kshitiz Malik, Kuenley Chiu, Kunal Bhalla, Kushal Lakhotia, Lauren Rantala-Yeary, Laurens van der Maaten, Lawrence Chen, Liang Tan, Liz Jenkins, Louis Martin, Lovish Madaan, Lubo Malo, Lukas Blecher, Lukas Landzaat, Luke de Oliveira, Madeline Muzzi, Mahesh Pasupuleti, Mannat Singh, Manohar Paluri, Marcin Kardas, Maria Tsimpoukelli, Mathew Oldham, Mathieu Rita, Maya Pavlova, Melanie Kambadur, Mike Lewis, Min Si, Mitesh Kumar Singh, Mona Hassan, Naman Goyal,

Narjes Torabi, Nikolay Bashlykov, Nikolay Bogoychev, Niladri Chatterji, Ning Zhang, Olivier Duchenne, Onur Çelebi, Patrick Alrassy, Pengchuan Zhang, Pengwei Li, Petar Vasic, Peter Weng, Prajjwal Bhargava, Pratik Dubal, Praveen Krishnan, Punit Singh Koura, Puxin Xu, Qing He, Qingxiao Dong, Ragavan Srinivasan, Raj Ganapathy, Ramon Calderer, Ricardo Silveira Cabral, Robert Stojnic, Roberta Raileanu, Rohan Maheswari, Rohit Girdhar, Rohit Patel, Romain Sauvestre, Ronnie Polidoro, Roshan Sumbaly, Ross Taylor, Ruan Silva, Rui Hou, Rui Wang, Saghar Hosseini, Sahana Chennabasappa, Sanjay Singh, Sean Bell, Seohyun Sonia Kim, Sergey Edunov, Shaoliang Nie, Sharan Narang, Sharath Raparthy, Sheng Shen, Shengye Wan, Shruti Bhosale, Shun Zhang, Simon Vandenhende, Soumya Batra, Spencer Whitman, Sten Sootla, Stephane Collot, Suchin Gururangan, Sydney Borodinsky, Tamar Herman, Tara Fowler, Tarek Sheasha, Thomas Georgiou, Thomas Scialom, Tobias Speckbacher, Todor Mihaylov, Tong Xiao, Ujjwal Karn, Vedanuj Goswami, Vibhor Gupta, Vignesh Ramanathan, Viktor Kerkez, Vincent Gonguet, Virginie Do, Vish Vogeti, Vítor Albiero, Vladan Petrovic, Weiwei Chu, Wenhan Xiong, Wenyin Fu, Whitney Meers, Xavier Martinet, Xiaodong Wang, Xiaofang Wang, Xiaoqing Ellen Tan, Xide Xia, Xinfeng Xie, Xuchao Jia, Xuewei Wang, Yaelle Goldschlag, Yashesh Gaur, Yasmine Babaei, Yi Wen, Yiwen Song, Yuchen Zhang, Yue Li, Yuning Mao, Zacharie Delpierre Coudert, Zheng Yan, Zhengxing Chen, Zoe Papakipos, Aaditya Singh, Aayushi Srivastava, Abha Jain, Adam Kelsey, Adam Shajnfeld, Adithya Gangidi, Adolfo Victoria, Ahuva Goldstand, Ajay Menon, Ajay Sharma, Alex Boesenberg, Alexei Baevski, Allie Feinstein, Amanda Kallet, Amit Sangani, Amos Teo, Anam Yunus, Andrei Lupu, Andres Alvarado, Andrew Caples, Andrew Gu, Andrew Ho, Andrew Poulton, Andrew Ryan, Ankit Ramchandani, Annie Dong, Annie Franco, Anuj Goyal, Aparajita Saraf, Arkabandhu Chowdhury, Ashley Gabriel, Ashwin Bharambe, Assaf Eisenman, Azadeh Yazdan, Beau James, Ben Maurer, Benjamin Leonhardi, Bernie Huang, Beth Loyd, Beto De Paola, Bhargavi Paranjape, Bing Liu, Bo Wu, Boyu Ni, Braden Hancock, Bram Wasti, Brandon Spence, Brani Stojkovic, Brian Gamido, Britt Montalvo, Carl Parker, Carly Burton, Catalina Mejia, Ce Liu, Changhan Wang, Changkyu Kim, Chao Zhou, Chester Hu, Ching-Hsiang Chu, Chris Cai, Chris Tindal, Christoph Feichtenhofer, Cynthia Gao, Damon Civin, Dana Beaty, Daniel Kreymer, Daniel Li, David Adkins, David Xu, Davide Testuggine, Delia David, Devi Parikh, Diana Liskovich, Didem Foss, Dingkang Wang, Duc Le, Dustin Holland, Edward Dowling, Eissa Jamil, Elaine Montgomery, Eleonora Presani, Emily Hahn, Emily Wood, Eric-Tuan Le, Erik Brinkman, Esteban Arcaute, Evan Dunbar, Evan Smothers, Fei Sun, Felix Kreuk, Feng Tian, Filippos Kokkinos, Firat Ozgenel, Francesco Caggioni, Frank Kanayet, Frank Seide, Gabriela Medina Florez, Gabriella Schwarz, Gada Badeer, Georgia Swee, Gil Halpern, Grant Herman, Grigory Sizov, Guangyi, Zhang, Guna Lakshminarayanan, Hakan Inan, Hamid Shojanazeri, Han Zou, Hannah Wang, Hanwen Zha, Haroun Habeeb, Harrison Rudolph, Helen Suk, Henry Aspegren, Hunter Goldman, Hongyuan Zhan, Ibrahim Damlaj, Igor Molybog, Igor Tufanov, Ilias Leontiadis, Irina-Elena Veliche, Itai Gat, Jake Weissman, James Geboski, James Kohli, Janice Lam, Japhet Asher, Jean-Baptiste Gaya, Jeff Marcus, Jeff Tang, Jennifer Chan, Jenny Zhen, Jeremy Reizenstein, Jeremy Teboul, Jessica Zhong, Jian Jin, Jingyi Yang, Joe Cummings, Jon Carvill, Jon Shepard, Jonathan McPhie, Jonathan Torres, Josh Ginsburg, Junjie Wang, Kai Wu, Kam Hou U, Karan Saxena, Kartikay Khandelwal, Katayoun Zand, Kathy Matosich, Kaushik Veeraraghavan, Kelly Michelena, Keqian Li, Kiran Jagadeesh, Kun Huang, Kunal Chawla, Kyle Huang, Lailin Chen, Lakshya Garg, Lavender A, Leandro Silva, Lee Bell, Lei Zhang, Liangpeng Guo, Licheng Yu, Liron Moshkovich, Luca Wehrstedt, Madian Khabsa, Manav Avalani, Manish Bhatt, Martynas Mankus, Matan Hasson, Matthew Lennie, Matthias Reso, Maxim Groshev, Maxim Naumov, Maya Lathi, Meghan Keneally, Miao Liu, Michael L. Seltzer, Michal Valko, Michelle Restrepo, Mihir Patel, Mik Vyatskov, Mikayel Samvelyan, Mike Clark, Mike Macey, Mike Wang, Miquel Jubert Hermoso, Mo Metanat, Mohammad Rastegari, Munish Bansal, Nandhini Santhanam, Natascha Parks, Natasha White, Navyata Bawa, Nayan Singhal, Nick Egebo, Nicolas Usunier, Nikhil Mehta, Nikolay Pavlovich Laptev, Ning Dong, Norman Cheng, Oleg Chernoguz, Olivia Hart, Omkar Salpekar, Ozlem Kalinli, Parkin Kent, Parth Parekh, Paul Saab, Pavan Balaji, Pedro Rittner, Philip Bontrager, Pierre Roux, Piotr Dollar, Polina Zvyagina, Prashant Ratanchandani, Pritish Yuvraj, Qian Liang, Rachad Alao, Rachel Rodriguez, Rafi Ayub, Raghotham Murthy, Raghu Nayani, Rahul Mitra, Rangaprabhu Parthasarathy, Raymond Li, Rebekkah Hogan, Robin Battey, Rocky Wang, Russ Howes, Ruty Rinott, Sachin Mehta, Sachin Siby, Sai Jayesh

Bondu, Samyak Datta, Sara Chugh, Sara Hunt, Sargun Dhillon, Sasha Sidorov, Satadru Pan, Saurabh Mahajan, Saurabh Verma, Seiji Yamamoto, Sharadh Ramaswamy, Shaun Lindsay, Shaun Lindsay, Sheng Feng, Shenghao Lin, Shengxin Cindy Zha, Shishir Patil, Shiva Shankar, Shuqiang Zhang, Shuqiang Zhang, Sinong Wang, Sneha Agarwal, Soji Sajuyigbe, Soumith Chintala, Stephanie Max, Stephen Chen, Steve Kehoe, Steve Satterfield, Sudarshan Govindaprasad, Sumit Gupta, Summer Deng, Sungmin Cho, Sunny Virk, Suraj Subramanian, Sy Choudhury, Sydney Goldman, Tal Remez, Tamar Glaser, Tamara Best, Thilo Koehler, Thomas Robinson, Tianhe Li, Tianjun Zhang, Tim Matthews, Timothy Chou, Tzook Shaked, Varun Vontimitta, Victoria Ajayi, Victoria Montanez, Vijai Mohan, Vinay Satish Kumar, Vishal Mangla, Vlad Ionescu, Vlad Poenaru, Vlad Tiberiu Mihailescu, Vladimir Ivanov, Wei Li, Wenchen Wang, Wenwen Jiang, Wes Bouaziz, Will Constable, Xiaocheng Tang, Xiaojian Wu, Xiaolan Wang, Xilun Wu, Xinbo Gao, Yaniv Kleinman, Yanjun Chen, Ye Hu, Ye Jia, Ye Qi, Yenda Li, Yilin Zhang, Ying Zhang, Yossi Adi, Youngjin Nam, Yu, Wang, Yu Zhao, Yuchen Hao, Yundi Qian, Yunlu Li, Yuzi He, Zach Rait, Zachary DeVito, Zef Rosnbrick, Zhaoduo Wen, Zhenyu Yang, Zhiwei Zhao, and Zhiyu Ma. The llama 3 herd of models, 2024. URL https://arxiv.org/abs/2407.21783.

Daniel A Herrmann and Benjamin A Levinstein. Standards for belief representations in llms. *arXiv preprint arXiv:2405.21030*, 2024.

Guiyang Hou, Wenqi Zhang, Yongliang Shen, Linjuan Wu, and Weiming Lu. TimeToM: Temporal space is the key to unlocking the door of large language models' theory-of-mind. In Lun-Wei Ku, Andre Martins, and Vivek Srikumar (eds.), *Findings of the Association for Computational Linguistics ACL 2024*, pp. 11532–11547, Bangkok, Thailand and virtual meeting, August 2024. Association for Computational Linguistics. URL https://aclanthology.org/2024.findings-acl.685.

Jennifer Hu, Felix Sosa, and Tomer Ullman. Re-evaluating theory of mind evaluation in large language models. *arXiv preprint arXiv:2502.21098*, 2025.

Chuanyang Jin, Yutong Wu, Jing Cao, Jiannan Xiang, Yen-Ling Kuo, Zhiting Hu, Tomer Ullman, Antonio Torralba, Joshua Tenenbaum, and Tianmin Shu. MMToM-QA: Multimodal theory of mind question answering. In Lun-Wei Ku, Andre Martins, and Vivek Srikumar (eds.), *Proceedings of the 62nd Annual Meeting of the Association for Computational Linguistics (Volume 1: Long Papers)*, pp. 16077–16102, Bangkok, Thailand, August 2024. Association for Computational Linguistics. URL https://aclanthology.org/2024.acl-long.851.

Hyunwoo Kim, Melanie Sclar, Xuhui Zhou, Ronan Bras, Gunhee Kim, Yejin Choi, and Maarten Sap. FANToM: A benchmark for stress-testing machine theory of mind in interactions. In Houda Bouamor, Juan Pino, and Kalika Bali (eds.), *Proceedings of the 2023 Conference on Empirical Methods in Natural Language Processing*, pp. 14397–14413, Singapore, December 2023a. Association for Computational Linguistics. doi: 10.18653/v1/2023.emnlp-main.890. URL https://aclanthology.org/2023.emnlp-main.890/.

Hyunwoo Kim, Melanie Sclar, Xuhui Zhou, Ronan Le Bras, Gunhee Kim, Yejin Choi, and Maarten Sap. Fantom: A benchmark for stress-testing machine theory of mind in interactions. *arXiv preprint arXiv:2310.15421*, 2023b.

Najoung Kim and Sebastian Schuster. Entity tracking in language models. *arXiv preprint arXiv:2305.02363*, 2023.

Youjeong Kim and S Shyam Sundar. Anthropomorphism of computers: Is it mindful or mindless? *Computers in Human Behavior*, 28(1):241–250, 2012.

Michal Kosinski. Evaluating large language models in theory of mind tasks. *Proceedings of the National Academy of Sciences*, 121(45), October 2024. ISSN 1091-6490. doi: 10.1073/pnas.2405460121. URL http://dx.doi.org/10.1073/pnas.2405460121.

Matthew Le, Y-Lan Boureau, and Maximilian Nickel. Revisiting the evaluation of theory of mind through question answering. In *Proceedings of the 2019 Conference on Empirical Methods in Natural Language Processing and the 9th International Joint Conference on Natural Language Processing (EMNLP-IJCNLP)*, pp. 5872–5877, 2019.

Belinda Z. Li, Maxwell Nye, and Jacob Andreas. Implicit representations of meaning in neural language models. In Chengqing Zong, Fei Xia, Wenjie Li, and Roberto Navigli (eds.), *Proceedings of the 59th Annual Meeting of the Association for Computational Linguistics and the 11th International Joint Conference on Natural Language Processing (Volume 1: Long Papers)*, pp. 1813–1827, Online, August 2021. Association for Computational Linguistics. doi: 10.18653/v1/2021.acl-long.143. URL https://aclanthology.org/2021.acl-long.143.

Kenneth Li, Oam Patel, Fernanda Viégas, Hanspeter Pfister, and Martin Wattenberg. Inference-time intervention: Eliciting truthful answers from a language model. *Advances in Neural Information Processing Systems*, 36, 2024.

Tom Lieberum, Matthew Rahtz, János Kramár, Neel Nanda, Geoffrey Irving, Rohin Shah, and Vladimir Mikulik. Does circuit analysis interpretability scale? evidence from multiple choice capabilities in chinchilla, 2023. URL https://arxiv.org/abs/2307.09458.

Kevin Meng, David Bau, Alex Andonian, and Yonatan Belinkov. Locating and editing factual associations in GPT. *Advances in Neural Information Processing Systems*, 36, 2022.

Shima Rahimi Moghaddam and Christopher J Honey. Boosting theory-of-mind performance in large language models via prompting. *arXiv preprint arXiv:2304.11490*, 2023.

Aaron Mueller, Jannik Brinkmann, Millicent Li, Samuel Marks, Koyena Pal, Nikhil Prakash, Can Rager, Aruna Sankaranarayanan, Arnab Sen Sharma, Jiuding Sun, Eric Todd, David Bau, and Yonatan Belinkov. The quest for the right mediator: A history, survey, and theoretical grounding of causal interpretability, 2024. URL https://arxiv.org/abs/2408.01416.

Catherine Olsson, Nelson Elhage, Neel Nanda, Nicholas Joseph, Nova DasSarma, Tom Henighan, Ben Mann, Amanda Askell, Yuntao Bai, Anna Chen, Tom Conerly, Dawn Drain, Deep Ganguli, Zac Hatfield-Dodds, Danny Hernandez, Scott Johnston, Andy Jones, Jackson Kernion, Liane Lovitt, Kamal Ndousse, Dario Amodei, Tom Brown, Jack Clark, Jared Kaplan, Sam McCandlish, and Chris Olah. In-context learning and induction heads. *Transformer Circuits Thread*, 2022. https://transformer-circuits.pub/2022/in-context-learning-and-induction-heads/index.html.

Nikhil Prakash, Tamar Rott Shaham, Tal Haklay, Yonatan Belinkov, and David Bau. Fine-tuning enhances existing mechanisms: A case study on entity tracking. In *Proceedings of the 2024 International Conference on Learning Representations*, 2024. arXiv:2402.14811.

David Premack and Guy Woodruff. Does the chimpanzee have a theory of mind? *Behavioral and Brain Sciences*, 1(4):515–526, 1978. doi: 10.1017/S0140525X00076512.

Christoph Riedl, Young Ji Kim, Pranav Gupta, Thomas W Malone, and Anita Williams Woolley. Quantifying collective intelligence in human groups. *Proceedings of the National Academy of Sciences*, 118(21):e2005737118, 2021.

Nina Rimsky, Nick Gabrieli, Julian Schulz, Meg Tong, Evan Hubinger, and Alexander Matt Turner. Steering llama 2 via contrastive activation addition. *arXiv preprint arXiv:2312.06681*, 2023.

Naomi Saphra and Sarah Wiegreffe. Mechanistic? In Yonatan Belinkov, Najoung Kim, Jaap Jumelet, Hosein Mohebbi, Aaron Mueller, and Hanjie Chen (eds.), *Proceedings of the 7th BlackboxNLP Workshop: Analyzing and Interpreting Neural Networks for NLP*, pp. 480–498, Miami, Florida, US, November 2024. Association for Computational Linguistics. doi: 10.18653/v1/2024.blackboxnlp-1.30. URL https://aclanthology.org/2024.blackboxnlp-1.30/.

Melanie Sclar, Sachin Kumar, Peter West, Alane Suhr, Yejin Choi, and Yulia Tsvetkov. Minding language models'(lack of) theory of mind: A plug-and-play multi-character belief tracker. *arXiv preprint arXiv:2306.00924*, 2023.

Melanie Sclar, Jane Yu, Maryam Fazel-Zarandi, Yulia Tsvetkov, Yonatan Bisk, Yejin Choi, and Asli Celikyilmaz. Explore theory of mind: Program-guided adversarial data generation for theory of mind reasoning. *ICLR*, 2025.

Natalie Shapira, Guy Zwirn, and Yoav Goldberg. How well do large language models perform on faux pas tests? In *Findings of the Association for Computational Linguistics: ACL 2023*, pp. 10438–10451, Toronto, Canada, July 2023. Association for Computational Linguistics. URL https://aclanthology.org/2023.findings-acl.663.

Natalie Shapira, Mosh Levy, Seyed Hossein Alavi, Xuhui Zhou, Yejin Choi, Yoav Goldberg, Maarten Sap, and Vered Shwartz. Clever hans or neural theory of mind? stress testing social reasoning in large language models. In Yvette Graham and Matthew Purver (eds.), *Proceedings of the 18th Conference of the European Chapter of the Association for Computational Linguistics (Volume 1: Long Papers)*, pp. 2257–2273, St. Julian's, Malta, March 2024. Association for Computational Linguistics. URL https://aclanthology.org/2024.eacl-long.138.

Paul Smolensky. Neural and conceptual interpretation of PDP models. In James L. McClelland, David E. Rumelhart, and the PDP Research Group (eds.), *Parallel Distributed Processing: Explorations in the Microstructure of Cognition: Psychological and Biological Models*, volume 2, pp. 390–431. MIT Press, 1986.

James W. A. Strachan, Dalila Albergo, Giulia Borghini, Oriana Pansardi, Eugenio Scaliti, Saurabh Gupta, Krati Saxena, Alessandro Rufo, Stefano Panzeri, Guido Manzi, Michael S. A. Graziano, and Cristina Becchio. Testing theory of mind in large language models and humans. *Nature Human Behaviour*, 8(7):1285–1295, Jul 2024a. ISSN 2397-3374. doi: 10.1038/s41562-024-01882-z. URL https://doi.org/10.1038/s41562-024-01882-z.

James WA Strachan, Dalila Albergo, Giulia Borghini, Oriana Pansardi, Eugenio Scaliti, Saurabh Gupta, Krati Saxena, Alessandro Rufo, Stefano Panzeri, Guido Manzi, et al. Testing theory of mind in large language models and humans. *Nature Human Behaviour*, pp. 1–11, 2024b.

Winnie Street, John Oliver Siy, Geoff Keeling, Adrien Baranes, Benjamin Barnett, Michael McKibben, Tatenda Kanyere, Alison Lentz, Blaise Aguera y Arcas, and Robin I. M. Dunbar. Llms achieve adult human performance on higher-order theory of mind tasks, 2024. URL https://arxiv.org/abs/2405.18870.

Jesse Vig, Sebastian Gehrmann, Yonatan Belinkov, Sharon Qian, Daniel Nevo, Yaron Singer, and Stuart Shieber. Investigating gender bias in language models using causal mediation analysis. In H. Larochelle, M. Ranzato, R. Hadsell, M.F. Balcan, and H. Lin (eds.), *Advances in Neural Information Processing Systems*, volume 33, pp. 12388–12401. Curran Associates, Inc., 2020. URL https://proceedings.neurips.cc/paper_files/paper/2020/file/92650b2e92217715fe312e6fa7b90d82-Paper.pdf.

Kevin Ro Wang, Alexandre Variengien, Arthur Conmy, Buck Shlegeris, and Jacob Steinhardt. Interpretability in the wild: a circuit for indirect object identification in GPT-2 small. In *The Eleventh International Conference on Learning Representations, ICLR 2023, Kigali, Rwanda, May 1-5, 2023*. OpenReview.net, 2023. URL https://openreview.net/forum?id=NpsVSN6o4ul.

Alex Wilf, Sihyun Lee, Paul Pu Liang, and Louis-Philippe Morency. Think twice: Perspective-taking improves large language models' theory-of-mind capabilities. In Lun-Wei Ku, Andre Martins, and Vivek Srikumar (eds.), *Proceedings of the 62nd Annual Meeting of the Association for Computational Linguistics (Volume 1: Long Papers)*, pp. 8292–8308, Bangkok, Thailand, August 2024. Association for Computational Linguistics. URL https://aclanthology.org/2024.acl-long.451.

Heinz Wimmer and Josef Perner. Beliefs about beliefs: Representation and constraining function of wrong beliefs in young children's understanding of deception. *Cognition*, 13 (1):103–128, 1983.

Yufan Wu, Yinghui He, Yilin Jia, Rada Mihalcea, Yulong Chen, and Naihao Deng. Hi-ToM: A benchmark for evaluating higher-order theory of mind reasoning in large language

models. In Houda Bouamor, Juan Pino, and Kalika Bali (eds.), *Findings of the Association for Computational Linguistics: EMNLP 2023*, pp. 10691–10706, Singapore, December 2023. Association for Computational Linguistics. doi: 10.18653/v1/2023.findings-emnlp.717. URL https://aclanthology.org/2023.findings-emnlp.717.

Hainiu Xu, Runcong Zhao, Lixing Zhu, Jinhua Du, and Yulan He. OpenToM: A comprehensive benchmark for evaluating theory-of-mind reasoning capabilities of large language models. In Lun-Wei Ku, Andre Martins, and Vivek Srikumar (eds.), *Proceedings of the 62nd Annual Meeting of the Association for Computational Linguistics (Volume 1: Long Papers)*, pp. 8593–8623, Bangkok, Thailand, August 2024. Association for Computational Linguistics. URL https://aclanthology.org/2024.acl-long.466.

Pei Zhou, Aman Madaan, Srividya Pranavi Potharaju, Aditya Gupta, Kevin R McKee, Ari Holtzman, Jay Pujara, Xiang Ren, Swaroop Mishra, Aida Nematzadeh, et al. How far are large language models from agents with theory-of-mind? *arXiv preprint arXiv:2310.03051*, 2023.

Wentao Zhu, Zhining Zhang, and Yizhou Wang. Language models represent beliefs of self and others. *arXiv preprint arXiv:2402.18496*, 2024.

A Full prompt

No Visibility

Instruction: 1. Track the belief of each character as described in the story. 2. A character's belief is formed only when they perform an action themselves or can observe the action taking place. 3. A character does not have any beliefs about the container and its contents which they cannot observe. 4. To answer the question, predict only what is inside the queried container, strictly based on the belief of the character, mentioned in the question. 5. If the queried character has no belief about the container in question, then predict 'unknown'. 6. Do not predict container or character as the final output.
Story: Bob and Carla are working in a busy restaurant. To complete an order, Bob grabs an opaque bottle and fills it with beer. Then Carla grabs another opaque cup and fills it with coffee.
Question: What does Bob believe cup contains?
Answer:

Explicit Visibility

Instruction: 1. Track the belief of each character as described in the story. 2. A character's belief is formed only when they perform an action themselves or can observe the action taking place. 3. A character does not have any beliefs about the container and its contents which they cannot observe. 4. To answer the question, predict only what is inside the queried container, strictly based on the belief of the character, mentioned in the question. 5. If the queried character has no belief about the container in question, then predict 'unknown'. 6. Do not predict container or character as the final output.
Story: Bob and Carla are working in a busy restaurant. To complete an order, Bob grabs an opaque bottle and fills it with beer. Then Carla grabs another opaque cup and fills it with coffee. Bob can observe Carla's actions. Carla cannot observe Bob's actions. Bob cannot observe Carla's actions. Carla can observe Bob's actions.
Question: What does Bob believe cup contains?
Answer:

B The CausalToM Dataset

In total, there are 4 templates (one without and 3 with explicit visibility statements). Each template allows 4 different types of questions (CharacterX asked about ObjectY). we used lists of 103 Characters, 21 Objects, and 30 States. In our patching experiments (Sec. 4.2), We randomly sample 80 pairs of an original and a counterfactual stories in total.

C Desiderate Based Patching Via Causal Abstraction

Causal Models and Interventions A deterministic causal model $\mathcal{M}$ has *variables* that take on *values*. Each variable has a *mechanism* that determines the value of the variable based on the values of *parent variables*. Variables without parents, denoted $\mathbf{X}$, can be thought of as inputs that determine the setting of all other variables, denoted $\mathcal{M}(\mathbf{x})$. A *hard intervention* $A \leftarrow a$ overrides the mechanisms of variable A, fixing it to a constant value a.

Interchange Interventions We perform *interchange interventions* (Vig et al., 2020; Geiger et al., 2020) where a variable (or set of features) A is fixed to be the value it would take on if the LM were processing *counterfactual input* $\mathbf{c}$. We write $A \leftarrow \mathsf{Get}(\mathcal{M}(\mathbf{c}), A)$ where $\mathsf{Get}(\mathcal{M}(\mathbf{c}), A)$ is the value of variable A when $\mathcal{M}$ processes input $\mathbf{c}$. In experiments, we will feed a *base input* $\mathbf{b}$ to a model under an interchange intervention $\mathcal{M}_{A \leftarrow \mathsf{Get}(\mathcal{M}(\mathbf{c}), A))}(\mathbf{b})$.

Featurizing Hidden Vectors The dimensions of hidden vectors are not an ideal unit of analysis (Smolensky, 1986), and so it is typical to *featurize* a hidden vector using some invertible function, e.g., an orthogonal matrix, to project a hidden vector into a new variable

space with more interpretable dimensions called "features"(Mueller et al., 2024). A feature intervention $\mathbf{F_h} \leftarrow \mathbf{f}$ edits the mechanism of a hidden vector $\mathbf{h}$ to fix the value of features $\mathbf{F_h}$ to $\mathbf{f}$.

Alignment The LM is a *low-level causal model* $\mathcal{L}$ where variables are dimensions of hidden vectors and the hypothesis about LM structure is a *high-level causal model* $\mathcal{H}$. An *alignment* Π assigns each high-level variable A to features of a hidden vector $\mathbf{F}_\mathbf{h}^A$, e.g., orthogonal directions in the activation space of $\mathbf{h}$. To evaluate an alignment, we perform intervention experiments to evaluate whether high-level interventions on the variables in $\mathcal{H}$ have the same effect as interventions on the aligned features in $\mathcal{L}$.

Causal Abstraction We use interchange interventions to reveal whether the hypothesized causal model $\mathcal{H}$ is an abstraction of an LM $\mathcal{L}$. To simplify, assume both models share an input and output space. The high-level model $\mathcal{H}$ is an abstraction of the low-level model $\mathcal{L}$ under a given alignment when each high-level interchange intervention and the aligned low-level intervention result in the same output. For a high-level intervention on A aligned with low-level features $\mathbf{F}_\mathbf{h}^A$ with a counterfactual input $\mathbf{c}$ and base input $\mathbf{b}$, we write

$$\mathsf{GetOutput}(\mathcal{L}_{\mathbf{F}_\mathbf{h}^A \leftarrow \mathsf{Get}(\mathcal{L}(\mathbf{c}), \mathbf{F}_\mathbf{h}^A))}(\mathbf{b})) = \mathsf{GetOutput}(\mathcal{H}_{A \leftarrow \mathsf{Get}(\mathcal{H}(\mathbf{c}), A))}(\mathbf{b})) \tag{1}$$

If the low-level interchange intervention on the LM produces the same output as the aligned high-level intervention on the algorithm, this is a piece of evidence in favor of the hypothesis. This extends naturally to multi-variable interventions (Geiger et al., 2024).

Graded Faithfulness Metric We construct *counterfactual datasets* for each causal variable where an example consists of a base prompt and a counterfactual prompt . The *counterfactual label* is the expected output of the algorithm after the high-level interchange intervention, i.e., the right-side of Equation 1. The interchange intervention accuracy is the proportion of examples for which Equation 1 holds, i.e., the degree to which $\mathcal{H}$ faithfully abstracts $\mathcal{L}$.

Aligning Features to Causal Variables In our experiments, we use principal component analysis (PCA) to featurize residual stream vectors, i.e., features are the orthogonal principal components. For a given transformer layer and token location, we collect the residual stream vectors across a large number of examples and compute the principal components. Given PCA features $\mathbf{F_h}$ of a hidden vector $\mathbf{h}$ in the residual stream of the LM $\mathcal{L}$, we select features to align with a causal variable A in causal model $\mathcal{H}$ using Desiderata-Based Masking (DBM) (De Cao et al., 2020; Davies et al., 2023; Prakash et al., 2024). Given base input $\mathbf{b}$ and counterfactual input $\mathbf{c}$, we train a mask $\mathbf{m} \in [0,1]^{|\mathbf{F_h}|}$ on the objective

$$\mathsf{CE}\Big(\mathsf{GetLogits}\big(\mathcal{L}_{\mathbf{F_h} \leftarrow \mathbf{m} \circ \mathsf{Get}(\mathcal{L}(\mathbf{c}), \mathbf{F_h}))}(\mathbf{b})\big), \mathsf{GetLogits}\big(\mathcal{H}_{A \leftarrow \mathsf{Get}(\mathcal{H}(\mathbf{c}), A))}(\mathbf{b})\big)\Big) \tag{2}$$

D Pseudocode for the Belief Tracking High-Level Causal Model

Algorithm 2 High-level causal model for the no visibility

procedure BELIEFREPRESENTATION($c_1, o_1, s_1, c_2, o_2, s_2, q_c, q_o$)
  Ordering ID assignment
  $c_1^{OID}, o_1^{OID}, s_1^{OID} \leftarrow \text{AssignOIDS}([c_1, o_1, s_1], 1)$
  $c_2^{OID}, o_2^{OID}, s_2^{OID} \leftarrow \text{AssignOIDS}([c_2, o_2, s_2], 2)$

  Binding lookback mechanism
  $\text{binding\_address}_1 \leftarrow (\text{copy}(c_1^{OID}), \text{copy}(o_1^{OID}))$
  $\text{binding\_address}_2 \leftarrow (\text{copy}(c_2^{OID}), \text{copy}(o_2^{OID}))$

  $q_c^{OID} \leftarrow \text{copy}(\{c_1 : c_1^{OID}, c_2 : c_2^{OID}\}[q_c])$
  $q_o^{OID} \leftarrow \text{copy}(\{o_1 : o_1^{OID}, o_2 : o_2^{OID}\}[q_o])$
  $\text{binding\_pointer} \leftarrow (q_c^{OID}, q_o^{OID})$

  if $\text{binding\_address}_1 = \text{binding\_pointer}$ **then**
    $\text{binding\_payload} \leftarrow \text{copy}(s_1^{OID})$
  else if $\text{binding\_address}_2 = \text{binding\_pointer}$ **then**
    $\text{binding\_payload} \leftarrow \text{copy}(s_2^{OID})$
  end if

  Answer lookback mechanism
  $\text{answer\_pointer} \leftarrow \text{binding\_payload}$
  $\text{answer1\_address} \leftarrow s_1^{OID}$
  $\text{answer2\_address} \leftarrow s_2^{OID}$
  if $\text{answer1\_address} = \text{answer\_pointer}$ **then**
    $\text{answer\_payload} \leftarrow s_1$
  else if $\text{answer2\_address} = \text{answer\_pointer}$ **then**
    $\text{answer\_payload} \leftarrow s_2$
  end if
  return answer_payload
end procedure

E Causal Mediation Analysis

In addition to the experiment shown in Fig.9, we conduct similar experiments for the object and state tokens by replacing them in the story with random tokens, which alters the original example's final output. However, patching the residual stream vectors of these tokens from the counterfactual run restores the relevant information, enabling the model to predict the expected output. The results of these experiments are collectively presented in Fig.2, with separate heatmaps shown in Fig. 10, 11, 12.

F Desiderata-based Component Masking

While interchange interventions on residual vectors reveal where a causal variable might be encoded in the LM's internal activations, they do not localize the variable to specific subspaces. To address this, we apply the *Desiderata-based Component Masking* technique, which learns a sparse binary mask over the singular vectors of the LM's internal activations. First, we cache the internal activations from 500 samples at the token positions where residual-level interchange interventions align with the expected output. Next, we apply *Singular Value Decomposition* to compute the singular vectors, which are then used to construct a *projection matrix*. Rather than replacing the entire residual vector with that

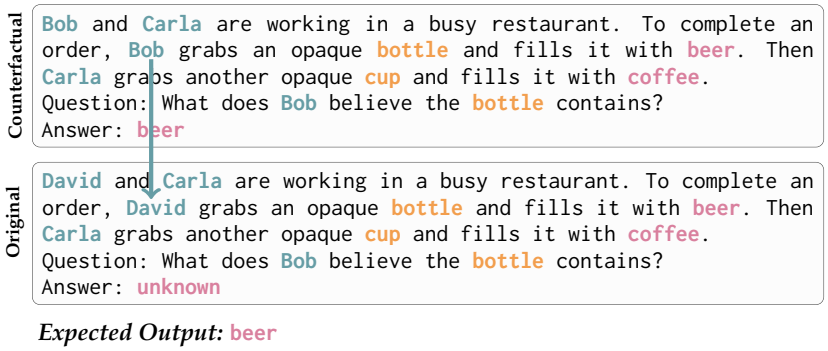

Figure 9: Causal Mediation Analysis: The original example produces the output *unknown* because *Bob* is not mentioned in the story, leaving the model without any information about his beliefs. However, when the residual stream vectors corresponding to *Bob* from the counterfactual run are patched into the original run, the model acquires the necessary information about that character and consequently updates its output to *beer*.

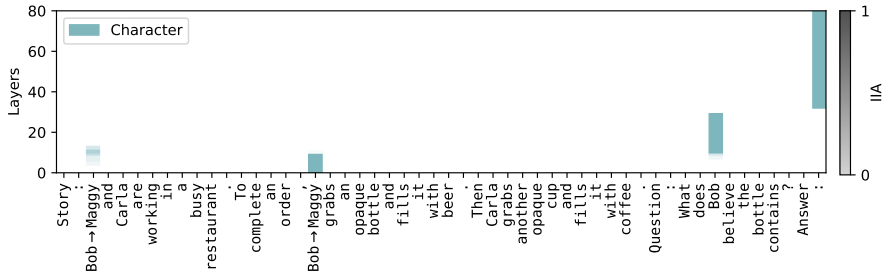

Figure 10: Information flow of character input tokens using causal mediation analysis.

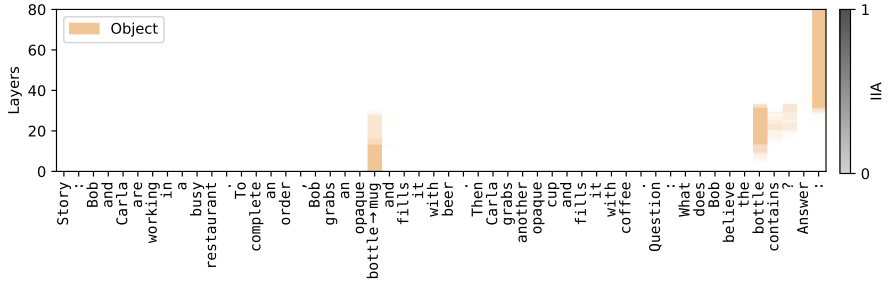

Figure 11: Information flow of object input tokens using causal mediation analysis.

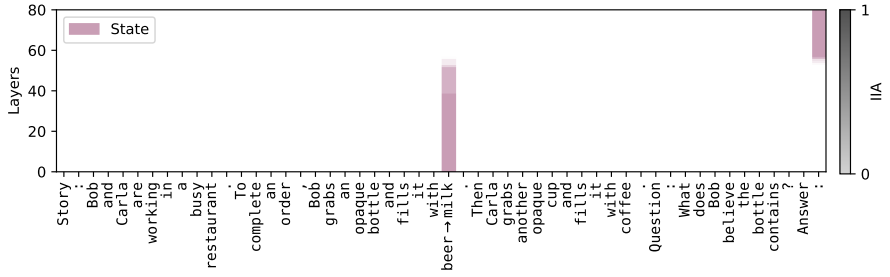

Figure 12: Information flow of state input tokens using causal mediation analysis.

from the counterfactual run, we perform subspace-level interchange interventions using the following equations:

$$W_{proj} = V * V^T \tag{3}$$

$$h_{org} \leftarrow W_{proj} h_{counterfactual} + (I - W_{proj}) h_{original} \tag{4}$$

Here, V is a matrix containing stacked singular vectors, while $h_{counterfactual}$ and $h_{original}$ represent the residual stream vectors from the counterfactual and original runs, respectively. The core idea is to first remove the existing information from the subspace defined by the projection matrix and then insert the counterfactual information into that same subspace using the same projection matrix. However, in DCM, instead of utilizing the entire internal activation space, we learn a binary mask over the projection matrix to identify the desired subspace. Specifically, before applying the intervention, we use the following equations to select the relevant subspace:

$$W_{proj} \leftarrow W_{proj} * mask \tag{5}$$

We train the mask on 80 examples of the same counterfactual dataset and use another 80 as the validation set. We use the following objective function, which maximizes the logit of the expected token:

$$\mathcal{L} = -\text{logit}_{expected\_output} + \lambda \sum 1 - W \tag{6}$$

Where λ is a hyperparameter used to control the rank of the subspace and W is the parameter of the learnable mask. We trained it for one epoch with ADAM optimizer, a batch size of 4 and a learning rate of 0.01.

G Aligning Character and Object OIDs

As mentioned in section 4.2, the source information, consisting of character and object OID, is transferred to the recalled token (state token) to form the address. Here, we describe another experiment to verify that the source information is copied to both the address and the pointer. More specifically, we conduct the same interchange intervention experiment as described in Fig. 5, but without freezing the residual vectors at the state tokens. Based our hypothesis, this intervention will not be able to change the state of the original run, since the intervention at the source information will affect both address and pointer, hence making the model form the correct QK-circuit.

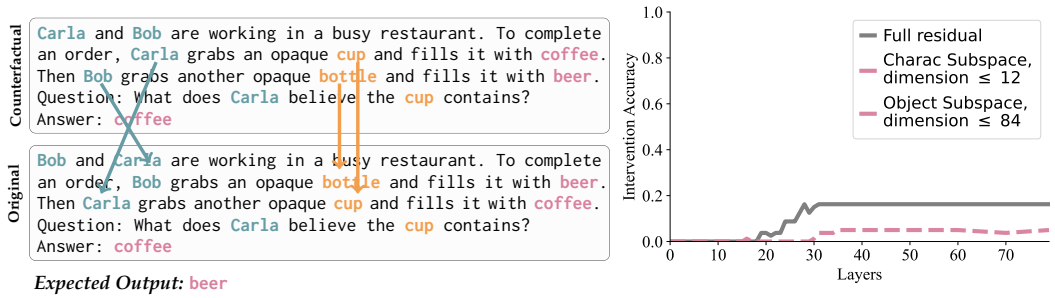

Figure 13: **Intervention Source Information** without freezing address and pointer: To swap the *source information* of the binding lookback, i.e., the initial character and object OIDs, we perform interchange interventions on their respective residual stream vectors up to a given layer (represented by the x-axis).

In section 4.3, we localized the source information, but it did not provide complete details about the location of each character and object OID. Therefore, in this section, we will

localize both separately to better understand at which layers they appear in the residual streams of their respective tokens, as shown in Fig.14 and Fig.15.

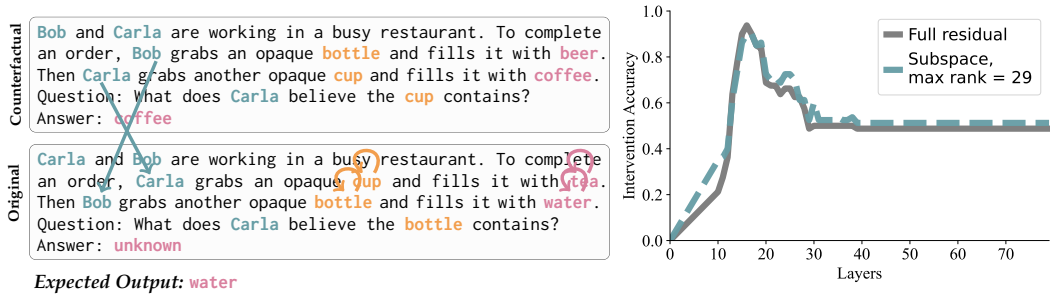

Figure 14: **Character OID**: This interchange intervention experiment swaps the character OID, while freezing the object and state OIDs. Swapping the character OIDs in the story tokens changes the queried character OID to the other one. Hence, the final output changes from *unknown* to **water**.

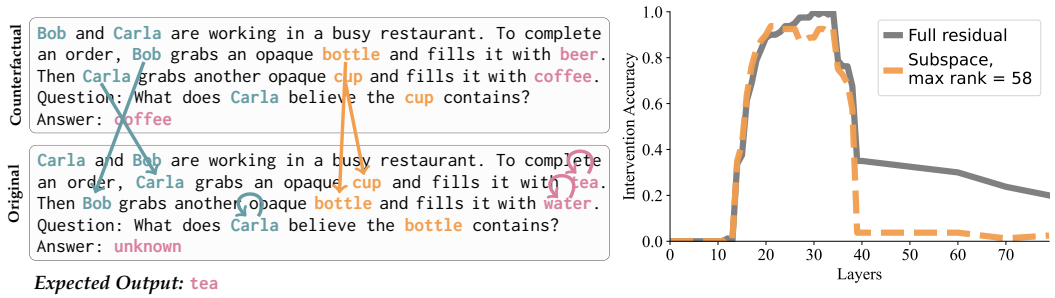

Figure 15: **Object OID**: This interchange intervention experiment swaps both the character and object OIDs, while freezing the state OID. Swapping both character and object OIDs in the story tokens ensures that the queried object gets the other OID. Hence, the final output changes from *unknown* to **water**.

H Aligning Query Character and Object OIDs

In section 4.3, we localized the pointer information. However, we found that this information is transferred to the lookback token (last token) through two intermediate tokens: the queried character and the queried object. In this section, we separately localize the OIDs of the queried character and queried object, as shown in Fig. 16 and Fig. 17.

I Speculated Payload in Visibility Lookback

As mentioned in section 5, the payload of the Visibility lookback remains undetermined. In this section, we attempt to disambiguate its semantics using the Attention Knockout technique introduced in (Geva et al., 2023), which helps reveal the flow of crucial information. We apply this technique to understand which previous tokens are vital for the formation of the payload information. Specifically, we "knock out" all attention heads at all layers of the second visibility sentence, preventing them from attending to one or more of the previous sentences. Then, we allow the attention heads to attend to the knocked-out sentence one layer at a time.

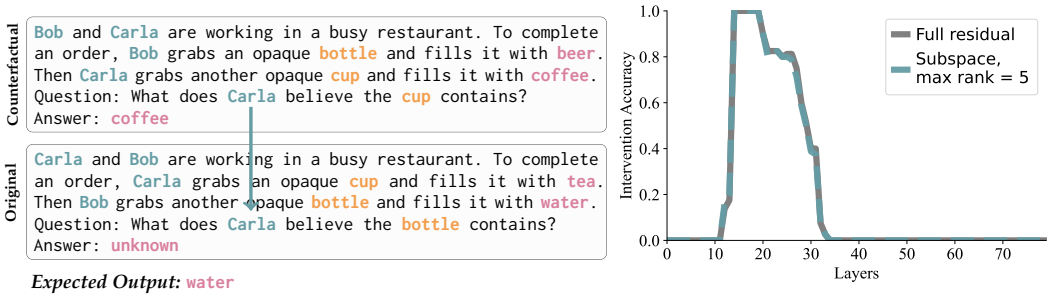

Figure 16: **Query Character OID**: This interchange intervention experiment alters the OID of the queried character to the other one. Hence, the final output changes from *unknown* to **water**.

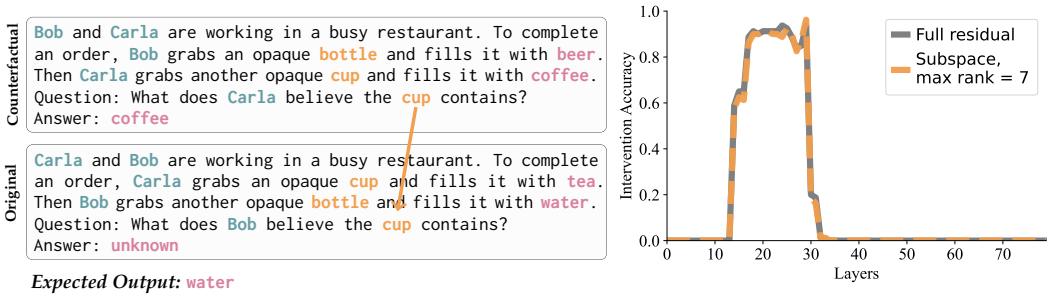

Figure 17: **Query Object OID**: This interchange intervention experiment alters the OID of the queried object to the other one. Hence, the final output changes from *unknown* to **water**.

If the LM is fetching vital information from the knocked-out sentence, the interchange intervention accuracy (IIA) post-knockout will decrease. Therefore, an increase in IIA will indicate which attention heads, at which layers, are bringing in the vital information from the knocked-out sentence. If, however, the model is not fetching any critical information from the knocked-out sentence, then knocking it out should not affect the IIA.

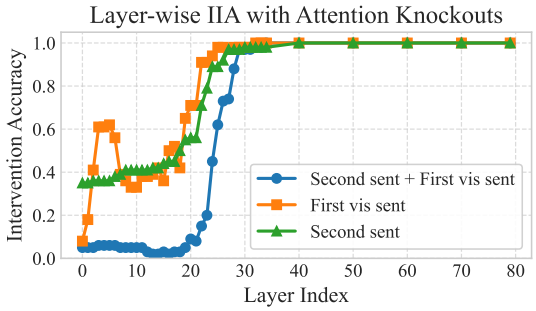

Figure 18: **Attention Knockout Results**:

To determine if any vital information is influencing the formation of the Visibility lookback payload, we perform three knockout experiments: 1) Knockout attention heads from the second visibility sentence to both the first visibility sentence and the second story sentence (which contains information about the observed character), 2) Knockout attention heads from the second visibility sentence to only the first visibility sentence, and 3) Knockout attention heads from the second visibility sentence to the second story sentence. In each experiment, we measure the effect of the knockout using IIA.

Fig.18 shows the experimental results. Knocking out any of the previous sentences affects the model's ability to produce the correct output. The decrease in IIA in the early layers can be explained by the restriction on the movement of character OIDs. Specifically, the second visibility sentence mentions the first and second characters, whose character OIDs must be fetched before the model can perform any further operations. Therefore, we believe the decrease in IIA until layer 15, when the character OIDs are formed (based on the results from Section G), can be attributed to the model being restricted from fetching the character OIDs. However, the persistently low IIA even after this layer—especially when both the second and first visibility sentences are involved—indicates that some vital information is being fetched by the second visibility sentence, which is essential for forming the coherent Visibility lookback payload. Thus, we speculate that the Visibility payload encodes information about the observed character, specifically their character OID, which is later used to fetch the correct state OID.