Language Models Use Lookbacks to Track Beliefs

Nikhil Prakash[◇], **Natalie Shapira**[◇], **Arnab Sen Sharma**[◇], **Christoph Riedl**[◇], **Yonatan Belinkov**[♠], **Tamar Rott Shaham**[♡], **David Bau**[◇], **Atticus Geiger**[♣†]

[◇]Northeastern University [♠]Technion [♡]MIT CSAIL [♣]Goodfire [†]Pr(Ai)^{2}R Group

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 LMs' ability to reason about characters' beliefs using causal mediation and abstraction. We construct a dataset, *CausalToM*, consisting of simple stories where two characters independently change the state of two objects, potentially unaware of each other's actions. Our investigation uncovers 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 their reference information, represented as Ordering IDs (OIs), 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 correct state OI and then the *answer lookback* retrieves the corresponding 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 OIs. 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 belief tracking mechanisms, taking a step toward reverse-engineering ToM reasoning in LMs.

1 Introduction

Theory of Mind (ToM), the ability to infer others' mental states, is an essential aspect of social and collective intelligence (Premack & Woodruff, 1978; Riedl et al., 2021). Recent studies have established that LMs can solve some tasks requiring ToM reasoning (Street et al., 2024; Strachan et al., 2024a; Kosinski, 2024), while others have highlighted shortcomings (Ullman, 2023; Sclar et al., 2025; Shapira et al., 2024, *inter alia*). Previous studies primarily rely on behavioral evaluations, which do not shed light on 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; Gweon et al., 2023).

In this work, we examine *how LMs internally represent and track beliefs* of characters, a core aspect of ToM (Dennett, 1981; Wimmer & Perner, 1983). A classic example is the Sally-Anne test (Baron-Cohen et al., 1985), which evaluates ToM in humans by assessing whether individuals can track conflicting beliefs: Sally's belief, which diverges from reality because of missing information, and Anne's belief, which is updated based on new observations. Our goal is to determine whether LMs learn a systematic solution to such tasks or rely on superficial statistical association.

Correspondence to prakash.nik@northeastern.edu.

39th Conference on Neural Information Processing Systems (NeurIPS 2025) Workshop: Mechanistic Interpretability Workshop.

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 and Llama-3.1-405B-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).

Our findings provide strong evidence for a systematic solution to belief tracking. We discover that LMs use a pervasive computation, which we refer to as the *lookback mechanism*, for belief tracking. This mechanism enables the model to recall important information at a later stage. In a lookback, two copies of a single piece of information are transferred to two distinct tokens. This allows attention heads at the latter token to look back at the earlier one when needed and retrieve vital information stored there, rather than transferring it directly (see Fig. 1).

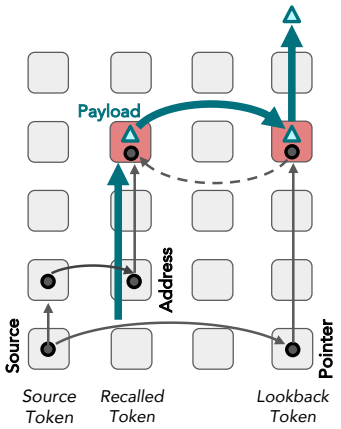

Figure 1: **The lookback mechanism** performs conditional reasoning; The *source token* contains reference information that is copied into two instances, creating a *pointer* and an *address*. Next to the address in the residual stream is a *payload*. When necessary, the model retrieves the payload by dereferencing the pointer. Solid lines represent information flow, while the dotted line indicates the attention "looking back" from pointer to address.

We identify three key lookback mechanisms that collectively perform belief tracking: 1) *Binding lookback* (Fig. 3(i)): First, the LM assigns *ordering IDs* (OIs; Dai et al. 2024) that encode whether a character, object, or state token appears first or second. Then, the character and object OIs are copied to the corresponding state token and the final token residual stream. Later, when the LM needs to answer a question about a character's beliefs, it uses this information to retrieve the answer state OI. 2) *Answer lookback* (Fig. 3(ii)): Uses the answer state OI from the binding lookback to retrieve the answer state token value. 3) *Visibility lookback* (Fig. 7): When a 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 but also uncovers a pervasive mechanism that plays a foundational role for in-context reasoning. All code and data supporting this study are available at `https://belief.baulab.info`.

2 The Lookback Mechanism

Our investigation uncovers a recurring pattern of computation that we call the *lookback mechanism*.[2] In lookback, a *source reference* is copied (via attention) into an *address* copy in the residual stream of the *recalled token* and a *pointer* copy in the residual stream of the *lookback token* that occurs later in the text. The LM places the address alongside a *payload* in the recalled token's residual stream that can be brought forward to the lookback token if necessary. Fig. 1 shows a generic 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 (which 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. The pointer and address are not necessarily exact copies of the source reference, but they do have a high dot product after being transformed by a query or key attention matrix, respectively. 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*.

[2]Although this mechanism may resemble *induction heads* (Elhage et al., 2021; Olsson et al., 2022), it differs fundamentally. In induction heads, information from a previous token occurrence is passed only to the subsequent token, 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.

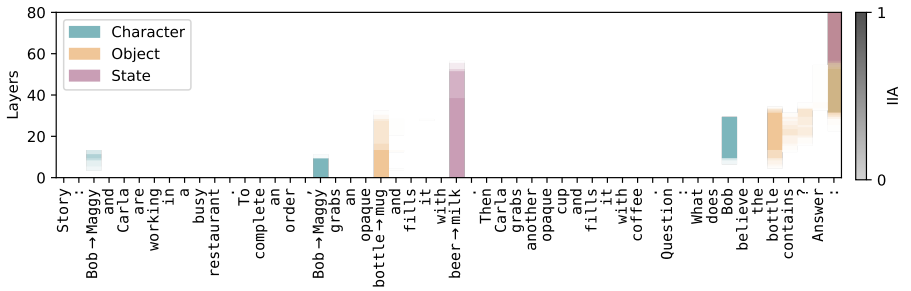

Figure 2: **Tracing information flow** of crucial input tokens using causal mediation analysis.

To develop an intuition for why an LM would learn to implement lookback mechanisms, consider that during training, LMs process text in sequence with no foreknowledge of what might come next. Instead of trying to resolve every possible future question about the current context, it would be useful to place addresses alongside payloads that might be useful to remember in the future when performing a variety of downstream tasks. In our setting, the LM constructs a representation of a story without any certainty about the questions it may later be asked about that story, so the LM localizes pivotal information in the residual stream of certain tokens, which later become payloads and addresses. When the question text is reached, pointers are constructed that reference this crucial story information and dereference it to find an answer to the question.

3 Experimental Setup: Dataset, Models, and Methods

Dataset Existing datasets for evaluating ToM capabilities of LMs are designed for behavioral testing and lack counterfactual pairs needed for causal analysis (Kim & Sundar, 2012). To address this problem, we construct *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., answer `unknown` when a character is unaware); refer to Appendices A & B for the full prompt and additional dataset details. All subsequent experiments are conducted on 80 samples that the model answers correctly. We also demonstrate generalization of the mechanism to BigToM dataset (Gandhi et al., 2024) in Appendix K.

Models Our experiments analyze Llama-3-70B-Instruct and Llama-3.1-405B-Instruct models in FP16 and INT8 precision, respectively, using *NNsight* (Fiotto-Kaufman et al., 2025). Results for Llama-3.1-405B-Instruct can be found in Appendix L. Both models demonstrate strong behavioral performance in the no-visibility and explicit-visibility settings. We do not examine smaller models, as they are unable to coherently solve the CausalToM task.

Causal Mediation Analysis Our goal is to develop a mechanistic understanding of how LMs reason about characters' beliefs and answer 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* $\mathbf{o}$ and a *counterfactual input* $\mathbf{c}$, and certain internal activations in the LM run on the original input are replaced with those computed from the counterfactual, a process also known as activation patching. We begin our analysis by tracing information flow from key input tokens to the final output, by performing interchange interventions on the residual vectors. Specifically, we construct an intervention dataset where $\mathbf{o}$ contains a question about the belief of a character not mentioned in the story, while the story in $\mathbf{c}$ includes the same queried character, as shown in Fig. 2. The expected outcome of this intervention is a change in the final output of $\mathbf{o}$ from *unknown* to a state token, such as **beer**. We conduct similar interchange interventions for object and state tokens (refer to Appendix C for details).

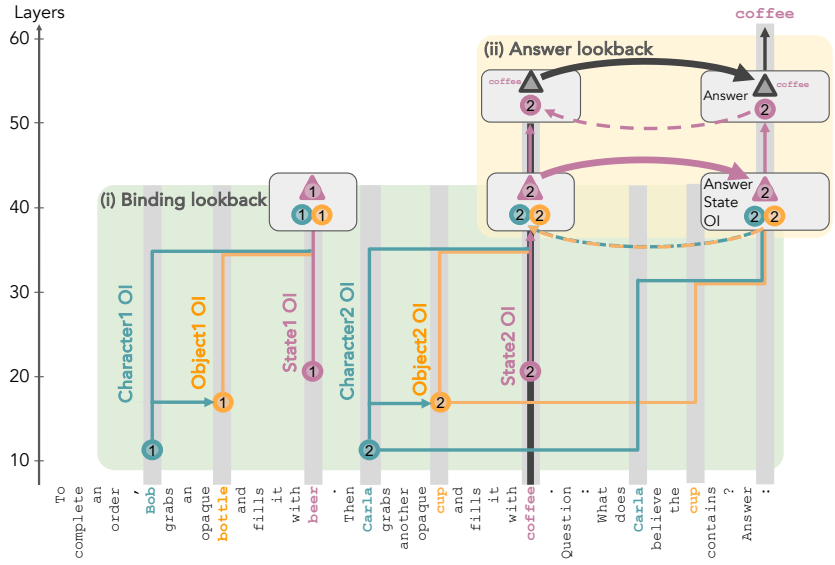

Figure 3: **Belief Tracking with *no visibility* between characters.** We hypothesize that the LM tracks beliefs using two lookback mechanisms. First, in (i) **Binding lookback**, LM binds together each character-object-state triple in the state token residual stream. When asked about a specific character-object pair, the LM looks back to the corresponding OIs to retrieve the correct state OI. Second, in (ii) **Answer lookback**, LM dereferences that state OI (used as a pointer) to retrieve the token value of the correct state. Colors indicate information type, shapes indicate role of information in lookback (see Fig. 1), e.g., state OI is a payload (△) in (i) and a pointer-address (○) in (ii).

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 the *interchange intervention accuracy* (IIA; Geiger et al., 2022). Even at this coarse level of Causal Mediation Analysis (Mueller et al., 2024; Vig et al., 2020; Meng et al., 2022), 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 where information flows, but do not identify what information is being transferred. In a transformer, the first layer represents the input and the last layer represents the output, but we wish to know: what is represented in the middle? We analyze the internal mechanism using *Causal Abstraction* (Geiger et al., 2021, 2024); First, we hypothesize a high-level causal model of the computational steps from input to output (Sec. 4), and then align its variables with the LM's internal activations (Sec. 5). We test the alignment through targeted interchange interventions on causal variables in the hypothesized model and hidden activations in the LM. If the LM produces the same output as the causal model under these aligned interventions, it provides evidence supporting the hypothesized causal model. We quantify this effect using *interchange intervention accuracy* (IIA; Geiger et al., 2022), which measures the proportion of cases where the intervened causal model and intervened LM agree. See Appendix D for more details.

In addition to measuring IIA on entire residual stream vectors, we also intervene on localized subspaces to further isolate causal variables. To identify the subspace of a specific variable, we employ *Desiderata-based Component Masking* (De Cao et al., 2020; Davies et al., 2023; Prakash et al., 2024). This method learns a sparse binary mask over the activation space that maximizes the logit of the hypothesized causal model output. We train a mask to select singular vectors of the activation space that encode a high-level variable (see Appendix F for details). Our experiments in Sec. 5 report both interventions on the full residual stream and on the identified subspaces.

4 Hypothesized High-Level Causal Model of Belief Tracking

Here we start with an overview of our hypothesized causal model of belief-tracking when characters are not aware of each other's actions. The causal model is an algorithmic process that has variables with structural roles that do not refer to the details of a transformer architecture. Appendix E presents the full pseudocode of the causal model. In Section 5, we will present experiments to verify that the causal model's variables align with representations in the transformer.

Belief tracking begins when the causal model assigns *ordering IDs* (OIs; ●, ●, ●) to each character, object, and state token, marking their order of appearance. For instance, in the example in Fig. 3, `Bob` is assigned first character OI (①), and `Carla` is assigned the second character OI(②). Then it uses these OIs in two lookback mechanisms:

(i) Binding lookback. The causal model creates address copies of each character OI (●) and object OI (●) that are bound to the state OI (Binding Payload, ▲), creating a character–object–state triple. When a question is asked about a character and object, the causal model creates pointer copies of that character and object OIs (Binding Pointers ●, ●) and dereferences them to retrieve the state OI.

(ii) Answer lookback. The causal model creates an address copy of the state OI (Answer Address ●) that is bound to the state token (Answer Payload, ▲). Through the binding lookback, a pointer copy of this OI (●) is created. The causal model dereferences the pointer to retrieves the correct state token payload as the final output.

5 Verifying the Hypothesized Causal Model of Belief Tracking

We test our hypothesized causal model by localizing its variables within the transformer's neural representations. Specifically, we localize the addresses, pointers, and payloads of the (i) binding lookback and (ii) answer lookbacks within the LM's internal activations. In Fig. 3, we show a trace of the causal model run on an input overlayed onto a schematic of a transformer architecture. This visualizes the alignment between variables in the causal model and locations in the LM residual stream that the experiments in the remaining of this paper will support. In the binding lookback, the character and object OI addresses are realized in the residual stream of the state token. The pointer copies are brought forward to the last token residual stream where they are dereferenced via attention to bring forward the correct state OI payload. In the answer lookback, the address copy of the state OI is in the state token residual stream while the pointer copy is in the last token residual stream.

Each of the following experiments localizes the presence of specific ordering IDs (OIs) and verifies their roles as hypothesized by our causal model. We do this by targeted interchange intervention experiments on the causal model and the LM. We copy hidden states between identical tokens (for example, replacing the representation of "**:**" in one context with the representation of "**:**" in another context, as in Fig. 4). When this intervention causes the LM's have the same output as the causal model under an interchange intervention on OI variables, we have evidence that the OI is carrying out the hypothesized role. Each experiment reports the effects of $n = 80$ different cases with the same structure, and the effect is measured at every layer.

Because the last step of the causal model is easiest to understand, we proceed through the experiments in reverse order, beginning with an experiment to verify the final "answer lookback" stage. After this instructive starting point, we work backward to verify the earlier steps of the model. Additional results can be found in Appendix G and H.

5.1 Step ii: Answer Lookback – Retrieving the Correct State

Localizing the Answer Payload We first verify the presence of the correct Answer Payload ▲ at the deepest layer representation of final token "**:**". To do so, we run an interchange intervention experiment shown in Fig. 4a in which the counterfactual example **c** swaps the order of the characters and objects of the original example **o** and also replaces the state (drinks) tokens with new values. If the Answer Payload is correctly localized, swapping it should cause the answer of the counterfactual (e.g., **tea**) to replace the answer of the original example (e.g., **coffee**). The gray line in Fig. 4b shows that this output change is observed in every one of $n = 80$ cases, both when intervening on the full residual stream and on the identified subspace. However, not at every layer: the information is only present after layer 56, indicating that before this stage, the transformer has not yet retrieved the

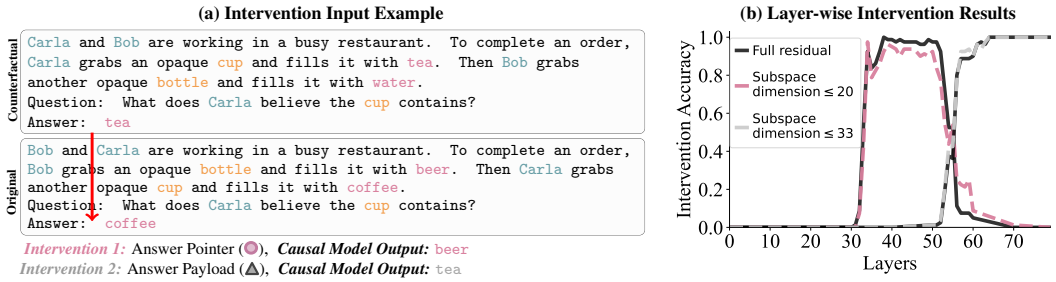

Figure 4: **Answer Lookback Pointer and Payload**: The causal model predicts that if we alter the "Answer Payload ▲" of the original to instead take the value of the counterfactual answer payload, the output should change from **coffee** to **tea**; the gray curve in the line plot shows this does occur when patching residual vectors at the ":" token beyond layer 56, providing evidence that the answer payload resides in those states. On the other hand the causal model predicts that taking the counterfactual "Answer Pointer ●" would change the original run output from **coffee** to **beer**—a new output that matches *neither* the original nor the counterfactual!—and we do see this surprising effect, again when patching layers between 34 and 52, providing strong evidence that the answer pointer is encoded at those layers. These results suggest the Answer Lookback occurs between layers 52 and 56.

correct answer payload into the residual stream. That is consistent with our hypothesis that at early steps, the OI has not yet been dereferenced. At an earlier stage, we expect to see an Answer Pointer.

Localizing the Pointer Information To identify the Answer Pointer ● before it is dereferenced to bring the payload (state token value), we examine the representations of "**:**" at layers earlier than 56. Our causal model provides the hypothesis: if the Answer Pointer is present, then patching the pointer from the counterfactual run into the original run should redirect the LM to attend to the location of the correct counterfactual state and fetch its payload. For example, in Fig. 4a the counterfactual pointer references the first presented state. When we patch it into the original story, we expect the model's answer to change to **beer** rather than **coffee**. The colored line in Fig. 4b confirms that this effect is consistently observed when patching any layer between $34 - 52$ (both when patching the full residual stream and the identified subspace), supporting our hypothesis that these layers encode the Answer Pointer information at the final token, rather than directly transferring token values.

5.2 Step i: Binding Lookback – Linking Characters, Objects, and States

Localizing the Address and Payload In this experiment, we verify the presence of the address copies of the character and object OIs as well as the payload (state OI) at the state token residual stream (recalled token, Fig. 3). As illustrated in Fig. 5a, we construct an intervention dataset where each example consists of an original input $\mathbf{o}$ with an answer that is not *unknown* and a counterfactual input $\mathbf{c}$ where the character, object, and state token values are identical, except the ordering of the two story sentences is swapped while the question remains unchanged. The expected LM's output predicted by our hypothesized causal model is the other state token in the original example, e.g., **beer**. That is because patching the address and payload values at each state token, without changing the pointer, makes the LM dereference the other state token. As a result, the model's output should flip to the other state token in the original input.

We perform the interchange intervention experiment layer-by-layer, where we replace the residual stream vector (or the identified subspace) of 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 OIs, it should not produce the expected output. (This happens in the earlier layers.) As shown in Fig. 5b, the strongest alignment occurs between layers 33 and 38, supporting our hypothesis that the state token's residual stream contains both the address (character and object OIs) and the payload information (state OI).

Localizing the Source Reference Information Next, we localize the source reference information, i.e., character and object OIs at their respective token residual stream. As illustrated in Fig. 6a, we conduct an intervention experiment with a dataset where the counterfactual example, $\mathbf{c}$, swaps the order of the characters and objects as well as replaces the state tokens with entirely new ones while keeping the question the same as in $\mathbf{o}$. Under this setup, an interchange intervention on the

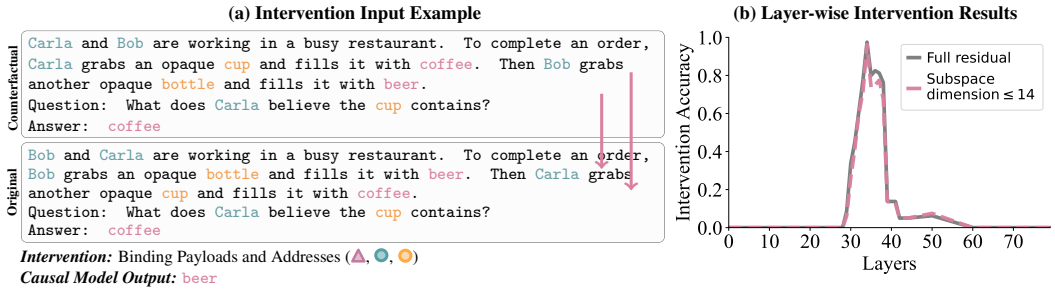

Figure 5: **Binding lookback Address and Payload:** The causal model predicts that swapping addresses (character and object OIs; and) and payloads (state OIs;) should cause the binding lookback mechanism to attend to the alternate state token and retrieve its state OI. This retrieved state OI is then dereferenced by the answer lookback, producing the corresponding token as the output (e.g., **beer** instead of **coffee**). The LM's behavior matches this prediction when we perform interchange interventions on the state token across layers 33–38. These findings support our hypothesis that both address and payload information are encoded in the residual stream of state tokens.

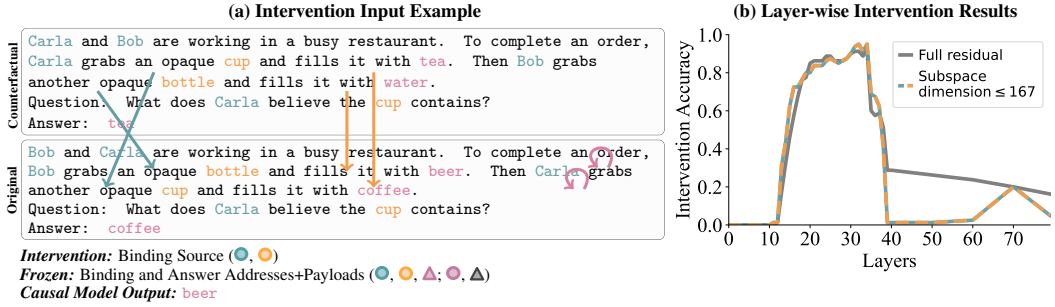

Figure 6: **Source Reference Information of Binding lookback**: The causal model predicts that swapping the source reference information (character and object OIs; ,), while freezing the addresses and payloads of the binding lookback, should cause the binding lookback mechanism to attend to the alternate state token and retrieve its state OI, which would generate alternate state token as the final output via the answer lookback (e.g., **beer** instead of **coffee**). The LM's behavior matches this prediction when we perform interchange interventions at the character and object tokens across layers 20-34. These results support our hypothesis that source reference information is encoded in the residual stream of character and object tokens.

hypothesized causal model that targets the source reference should propagate changes through both the address and the pointer, leaving the final output unchanged. However, if we instead freeze the state token residual stream, which carries both the payload and the address, the causal model produces the alternate state token (e.g., **beer** in Fig. 6), as the pointer refers to the other state's address.

In the LM, we interchange the residual streams of the character and object tokens layer-by-layer, while keeping the residual stream of the state token fixed. As shown in Fig. 6b, this experiment reveals alignment between layers 20 and 34, indicating that source reference is encoded in the residual streams of the character and object tokens within this layer range. Additional results are provided in Appendix G, where Fig. 13 shows that freezing the residual stream of the state token is necessary for this alignment to emerge. These findings support our hypothesis that source reference is present in the character and object tokens and is subsequently transferred to the recalled and lookback tokens.

Localizing the Pointer Information Finally, we localize the pointer copies of the character and object OIs to their corresponding tokens in the question and to the final token. See Appendices G & H for details of the experiments and results.

In summary, belief tracking begins in layers 20–34, where character and object OIs are encoded in their respective token representations. These OIs are transferred to the corresponding state tokens in layers 33–38. When a question is asked, pointer copies of the relevant character and object OIs are moved to the final token by layer 34, where they are dereferenced to retrieve the correct state OI. At the final token, this state OI is represented across layers 34–52, and between layers 52–56, it is dereferenced to fetch the answer from the correct state token, producing the final output.

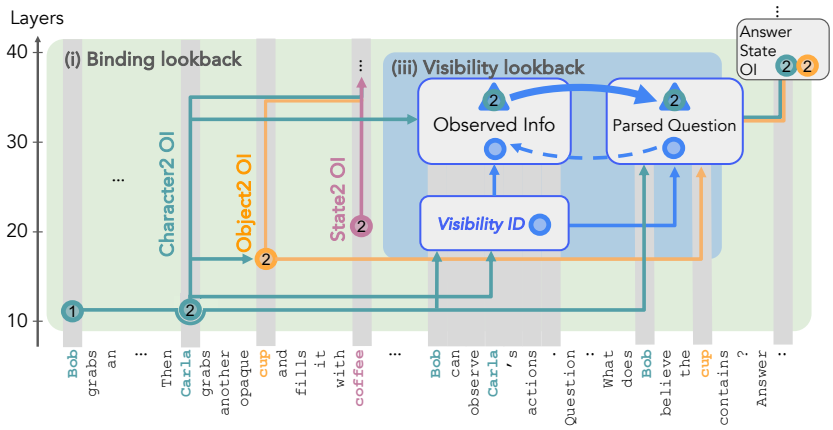

Figure 7: **Visibility Lookback** When one (observing) character can see another (observed) character, the LM assigns a visibility ID (●) to the visibility sentence where this relation is defined. An address copy of this visibility ID remains in the visibility sentence's residual stream. A pointer copy of the visibility ID is transferred to the subsequent tokens' residual stream. The LM dereferences this pointer through a QK-circuit, bringing forward the payload (▲), when processing subsequent tokens. Based on initial evidence, this payload contains the observed character's OI(●). See Appendix I for 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.

6 Impact of Visibility Conditions on Belief Tracking Mechanism

So far, we have 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 one character (*observing*) can observe the actions of the other (*observed*).

Hypothesized Visibility Lookback Mechanism We hypothesize that the LM uses an additional lookback mechanism, which we call the *Visibility Lookback*, to integrate information about the observed character when it is explicitly stated that one character can see another's action. 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 reference 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 stream of the subsequent tokens (lookback tokens). Then LM forms a QK-circuit at the lookback tokens and dereferences the visibility ID pointer to retrieve the payload.

Although our two-character setting is unable to discern the exact semantics of the payload in the visibility lookback, our observations are consistent with the payload encoding the observed character's OI. Our initial observations suggest another lookback where the story sentence associated with the observed character serves as the source reference, and its payload encodes information about the observed character. The observed character's OI appears to be retrieved by the lookback tokens of the Visibility lookback, with causal effects on the queried character's awareness (see App. I for details).

6.1 Verifying the Hypothesized Visibility Lookback

Localizing the Source Reference In this experiment, we localize the Visibility ID (●), i.e., the source reference of the Visibility lookback. We conduct an interchange intervention experiment where the counterfactual is a different story in which the characters' visibility is flipped from unobserved to observed (Fig. 8a), and we look for an output change from "unknown" to the answer that would be observed. We intervene on the representation of all the visibility sentence tokens. As shown in Fig. 8b (blue ─ line), causal effects appear between layers 10 and 23, indicating that the visibility ID remains encoded in the visibility sentence until layer 23, after which it is split into address and pointer copies that must be connected by dereference to have an effect. This pattern supports our hypothesis that the LM generates a reference to the Visibility ID.

Localizing the Payload Next, we localize the payload (▲) information using the same counterfactual dataset. However, instead of intervening on the recalled tokens, we intervene on the lookback tokens, specifically the question and answer tokens. As in the previous experiment, we replace the

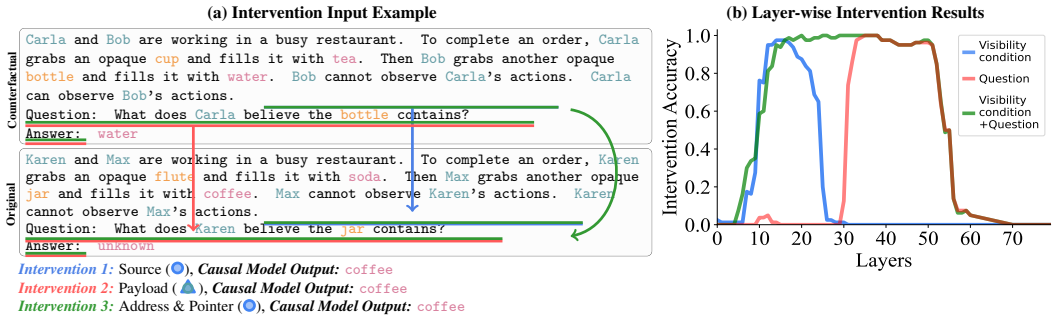

Figure 8: **Visibility Lookback**: We conduct three interchange intervention experiments to support the Visibility Lookback hypothesis: (1) *Source Alignment*: We align the source reference ( ) by intervening on the visibility sentence, replacing it with its representation from a counterfactual run where the visibility sentence causes the queried character to become aware of the queried object's contents. We observe that source reference information aligns between layers 10 and 23, after which it splits into separate address and pointer components. (2) *Payload Alignment*: To align the payload ( ), we intervene on all subsequent tokens and observe alignment only after layer 31. (3) *Address and Pointer Alignment*: When intervening on both the address and pointer information ( ), we observe alignment across a broader range of layers, particularly between layers 24 and 31, because of the enhanced alignment between the address and pointer copies at the recalled and lookback tokens.

residual vectors of these tokens in the original run with those from the counterfactual run. As shown in Fig. 8b (red — line), alignment occurs after layer 31, indicating that the information causing the queried character's awareness is present in the lookback tokens after this layer.

Localizing Address and Pointer The previous two experiments indicate the absence of both the source and payload information between layers 24 and 31. We hypothesize that this lack of signal is due to a mismatch between the address and pointer information that inhibits a lookback dereference. 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, a mismatch in either case. 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. 8b (green — line), alignment occurs after layer 10 and remains stable, providing evidence that a lookback now occurs between layers 24 and 31. This intervention replaces both the address and pointer copies of the visibility IDs, enabling the LM to form a QK-circuit and resolve the visibility lookback.

7 Related Work

Theory of mind in LMs Theory of mind in LMs has been widely benchmarked (Le et al., 2019; Shapira et al., 2023; Wu et al., 2023; Kim et al., 2023; Xu et al., 2024; Jin et al., 2024; Chan et al., 2024; Strachan et al., 2024b). However, these benchmarks lack adequate counterfactuals for the binding manipulations we need, so we made CausalToM (Section B).

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; Feng & Steinhardt, 2023; Kim & Schuster, 2023; Prakash et al., 2024; Feng et al., 2024; Dai et al., 2024; Wu et al., 2025). 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 Few studies explored the underlying mechanisms of ToM of LM (Zhu et al., 2024; Bortoletto et al., 2024; Herrmann & Levinstein, 2024). Those studies use probing (Alain, 2016; Belinkov, 2022) to identify internal representations of beliefs and steering (Rimsky et al., 2023; Li et al., 2024) to control LMs 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.

8 Conclusion

Through a series of interchange intervention experiments, we have mapped the end-to-end underlying mechanism responsible for the processing of partial knowledge and false beliefs in a set of simple stories. We are surprised 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 belief states. Our improved understanding of these fundamental computations gives us optimism that it may be possible to fully reveal the algorithms underlying not only the theory of mind, but also other capabilities in LMs.

9 Ethics Statement

This work involves experiments conducted exclusively on synthetic text generated by the authors. No human subjects, personal data, or sensitive user information were collected or analyzed. The models we analyze are publicly released LLMs (Llama-3-70B-Instruct and Llama-3.1-405B-Instruct). All interventions and analyses were performed locally without querying proprietary APIs. We acknowledge that research on ToM in LMs may be misinterpreted as suggesting human-like cognition or intentionality. We explicitly caution that our findings demonstrate internal computational mechanisms, not conscious reasoning, and should not be construed as evidence of sentience or moral agency in LMs. Our causal intervention techniques reveal latent structures within pretrained models but do not modify the underlying weights or enable novel capabilities. Nevertheless, reverse-engineering latent belief representations could, in principle, be misused for behavioral steering or manipulation. To mitigate this, we provide our code strictly for transparency and further scientific audit, rather than for application in deployed systems.

10 Reproducibility Statement

To facilitate reproducibility, we release the full *CausalToM* dataset, including all story templates and the code used to generate the various story instances that serve as counterfactual variants in our experiments. The repository, that can be accessed at `https://belief.baulab.info`, contains all scripts required to construct the dataset, extract activations, perform interchange interventions, and compute causal mediation metrics, along with the hyperparameters and random seeds used for subspace identification via DCM. All experiments were conducted using publicly available open-weight models (Llama-3-70B-Instruct and Llama-3.1-405B-Instruct). The interchange intervention experiments can be reproduced either by hosting these models locally or by using remotely hosted instances via NDIF (Fiotto-Kaufman et al., 2025), if sufficient local compute is not available.

11 The Use of Large Language Models

We used LLMs as a writing assistant to correct grammatical and typographical errors; beyond this, they did not contribute to any stage of the research.

12 Acknowledgement

This research was supported in part by Open Philanthropy (N.P., N.S., A.S.S., D.B., A.G., Y.B.), the NSF National Deep Inference Fabric award #2408455 (D.B.), the Israel Council for Higher Education (N.S.), the Zuckerman STEM Leadership Program (T.R.S.), the Israel Science Foundation (grant No. 448/20; Y.B.), an Azrieli Foundation Early Career Faculty Fellowship (Y.B.), a Google Academic Gift (Y.B.), and a Google Gemma Academic Award (D.B.). This research was partly funded by the European Union (ERC, Control-LM, 101165402). Views and opinions expressed are however those of the author(s) only and do not necessarily reflect those of the European Union or the European Research Council Executive Agency. Neither the European Union nor the granting authority can be held responsible for them.

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, Zhengxuan Wu, Hanson Lu, Josh Rozner, Elisa Kreiss, Thomas Icard, Noah Goodman, and Christopher Potts. Inducing causal structure for interpretable neural networks. In Kamalika Chaudhuri, Stefanie Jegelka, Le Song, Csaba Szepesvari, Gang Niu, and Sivan Sabato (eds.), *Proceedings of the 39th International Conference on Machine Learning*, volume 162 of *Proceedings of Machine Learning Research*, pp. 7324–7338. PMLR, 17–23 Jul 2022. URL https://proceedings.mlr.press/v162/geiger22a.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.

Hyowon Gweon, Judith Fan, and Been Kim. Socially intelligent machines that learn from humans and help humans learn. *Philosophical Transactions of the Royal Society A*, 381(2251):20220048, 2023.

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

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 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*, 2023.

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.

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, 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*, 8(7):1285–1295, 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.

Tomer Ullman. Large language models fail on trivial alterations to theory-of-mind tasks. *arXiv preprint arXiv:2302.08399*, 2023.

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.

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.

Yiwei Wu, Atticus Geiger, and Raphaël Millière. How do transformers learn variable binding in symbolic programs? *arXiv preprint arXiv:2505.20896*, 2025.

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.

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 the bottle 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.
Question: What does Bob believe the cup contains?
Answer:
```

B The CausalToM Dataset

We needed to construct a new dataset because we required a task that models could reliably solve. In contrast, most existing ToM datasets remain challenging for LMs. Additionally, we needed a dataset in which each sample is paired with multiple counterfactuals, enabling causal computations and the extraction of the underlying mechanism. The only dataset that met both criteria was BigToM, which we used in our study. However, even BigToM was insufficient for investigating the full range of factors influencing the mechanism, such as the relationship between a character and their object. Hence, we needed to simplify the task to allow for additional counterfactuals. To test the effect of a specific element, we required the ability to modify only that element without altering the rest of the story or creating an incoherent scenario. For example, consider a BigToM story where a flood occurs, and opening a gate releases the water. In the counterfactual scenario where the gate remains closed, the story's continuation becomes unintelligible, with the occurrence of a flood.

To address this, we developed CausalToM, which features simple stories accompanied by a range of counterfactuals. Key features include: (1) two characters, objects, and states, (2) the ability to modify each of them independently, and (3) control over whether characters witness each other's actions. The dataset comprises four templates, one without visibility statements and three with explicit visibility statements. Each template supports four types of questions (e.g., "CharacterX asked about ObjectY"). We used lists of 103 characters, 21 objects, and 23 states. For our interchange intervention experiments, we randomly sampled 80 pairs of original and counterfactual stories.

C Causal Mediation Analysis

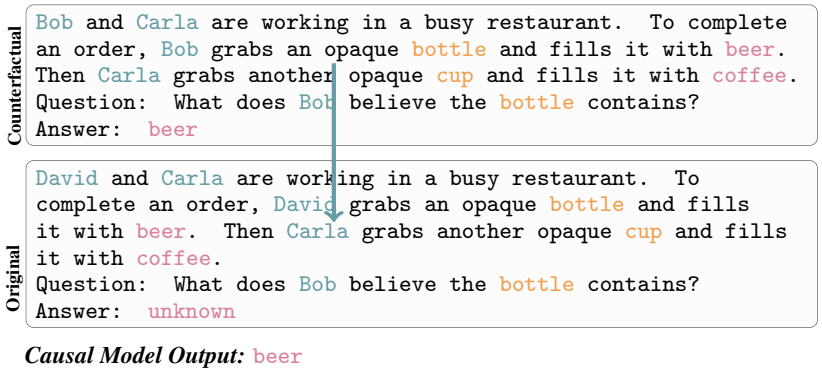

Figure 9: **Causal Mediation Analysis**: The original example produces the output *unknown* because *Bob* is not mentioned in the story, leaving the model without any information about his beliefs. However, when the residual stream vectors corresponding to *Bob* from the counterfactual run are patched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

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 causal model output. The results of these experiments are collectively presented in Fig.2, with separate heatmaps shown in Fig. 10, 11, 12.

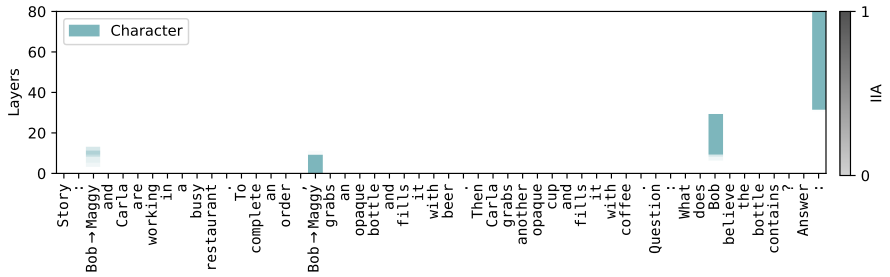

Figure 10: Information flow of character input tokens using causal mediation analysis.

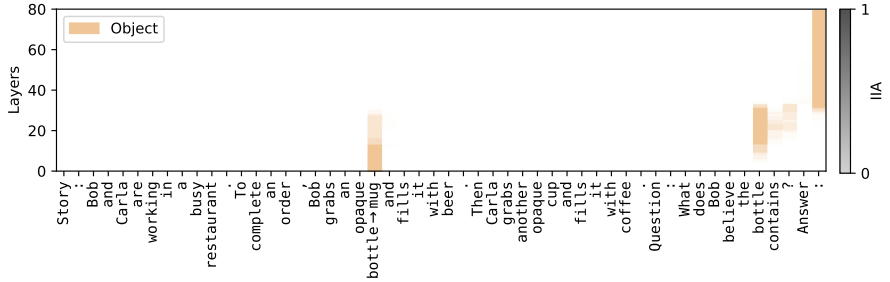

Figure 11: Information flow of object input tokens using causal mediation analysis.

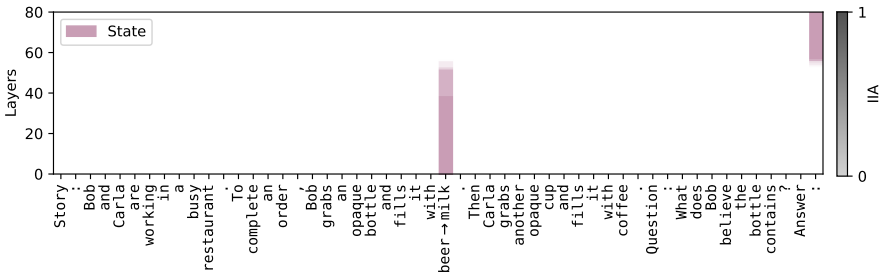

Figure 12: Information flow of state input tokens using causal mediation analysis.

D 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 *original input* $\mathbf{o}$ to a model under an interchange intervention $\mathcal{M}_{A \leftarrow \mathsf{Get}(\mathcal{M}(\mathbf{c}), A))}(\mathbf{o})$.

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 $\mathbf{F}_\mathbf{h} \leftarrow \mathbf{f}$ edits the mechanism of a hidden vector $\mathbf{h}$ to fix the value of features $\mathbf{F}_\mathbf{h}$ 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 $\mathbf{F}^A_\mathbf{h}$, 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 $\mathbf{F}^A_\mathbf{h}$ with a counterfactual input $\mathbf{c}$ and original input $\mathbf{b}$, we write

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

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 Singular Vector Decomposition (SVD) to featurize residual stream vectors, i.e., features are the orthogonal singular vectors. For a given transformer layer and token location, we collect the residual stream vectors across a large number of examples and compute the singular vectors. Given singular vector features $\mathbf{F}_\mathbf{h}$ 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 Component Masking (DCM) (De Cao et al., 2020; Davies et al., 2023; Prakash et al., 2024). Given original input $\mathbf{o}$ and counterfactual input $\mathbf{c}$, we train a mask $\mathbf{m} \in [0,1]^{|\mathbf{F_h}|}$ on the following objective

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

E Pseudocode for the Belief Tracking High-Level Causal Model

Algorithm 2 High-level causal model for the no visibility

procedure BELIEFTRACKING($c_1, o_1, s_1, c_2, o_2, s_2, q_c, q_o$)
    Ordering ID assignment
    $c_1^{OI}, o_1^{OI}, s_1^{OI} \leftarrow$ AssignOIs($[c_1, o_1, s_1], 1$)
    $c_2^{OI}, o_2^{OI}, s_2^{OI} \leftarrow$ AssignOIs($[c_2, o_2, s_2], 2$)

    Binding lookback mechanism
    binding_address$_1 \leftarrow$ (copy(c_1^{OI}), copy(o_1^{OI}))
    binding_address$_2 \leftarrow$ (copy(c_2^{OI}), copy(o_2^{OI}))

    $q_c^{OI} \leftarrow$ copy($\{c_1 : c_1^{OI}, c_2 : c_2^{OI}\}[q_c]$)
    $q_o^{OI} \leftarrow$ copy($\{o_1 : o_1^{OI}, o_2 : o_2^{OI}\}[q_o]$)
    binding_pointer $\leftarrow (q_c^{OI}, q_o^{OI})$

    if binding_address$_1$ = binding_pointer **then**
        binding_payload $\leftarrow$ copy(s_1^{OI})
    else if binding_address$_2$ = binding_pointer **then**
        binding_payload $\leftarrow$ copy(s_2^{OI})
    end if

    Answer lookback mechanism
    answer_pointer $\leftarrow$ binding_payload
    answer1_address $\leftarrow s_1^{OI}$
    answer2_address $\leftarrow s_2^{OI}$
    if answer1_address = answer_pointer **then**
        answer_payload $\leftarrow s_1$
    else if answer2_address = answer_pointer **then**
        answer_payload $\leftarrow s_2$
    end if
    return answer_payload
end procedure

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 (De Cao et al., 2020; Davies et al., 2023; Prakash et al., 2024), which learns a sparse binary mask $\mathbf{m}$ over the singular vectors of the LM's internal activations. We first cache the internal activations from 500 samples at the token positions specified in the main text for each experiment. Next, we apply *Singular Value Decomposition* to compute the singular vectors as a matrix $V \in \mathbb{R}^{d \times 500}$ where d is the dimensionality of the residual stream. We then masked this matrix using a learnable binary vector $\mathbf{m} \in [0, 1]^d$ to choose a subset of singular vectors

$$V_{masked} = V\mathbf{m} \tag{3}$$

The chosen subset of vectors is used to construct a *projection matrix* $W_{proj} \in \mathbb{R}^{d \times d}$.

$$W_{proj} = V_{masked} V_{masked}^T \tag{4}$$

Then, we perform subspace-level interchange interventions (rather than replacing the entire residual vector) using the following equations:

$$h_{new} = W_{proj} h_c + (I - W_{proj}) h_o \tag{5}$$

where h_o is the full residual stream of the original run, h_c is the full residual stream of the counterfactual run, and h_{new} is the intervened vector where the chosen subspace of h_o is replaced with that of h_c.

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.

In order to find the optimal subspace, we optimize $\mathbf{m}$ to maximize the agreement between the causal model output and the LM's output. To do so, we train the mask for each experiment on 80 examples of the same counterfactual datasets specified in the main text and use another 80 samples as the validation set. We use the following objective function, which maximizes the logit of the causal model output token:

$$\mathcal{L} = -\text{logit}_{\text{causal\_model\_output\_under\_intervention}} + \lambda \sum \mathbf{m} \tag{6}$$

Where λ is a hyperparameter used to control the rank of the subspace and $\mathbf{m}$ is the learnable mask. See Appendix D for details on how the causal model output under intervention are computed. We trained $\mathbf{m}$ for one epoch with ADAM optimizer, on batches of size 4 and a learning rate of 0.01. During training, the parameters of $\mathbf{m}$ are continuous and constrained to lie within the range $[0, 1]$. To enforce this constraint, we clamp their values after each gradient update. During evaluation, we binarize the mask by rounding each parameter to the nearest integer, i.e., 0 or 1.

G Aligning Character and Object OIs

As mentioned in section 5.2, the source reference information, consisting of character and object OI, is duplicated to form the address and pointer of the binding lookback. 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. 6, but without freezing the residual vectors at the state tokens. Based on 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 original QK-circuit.

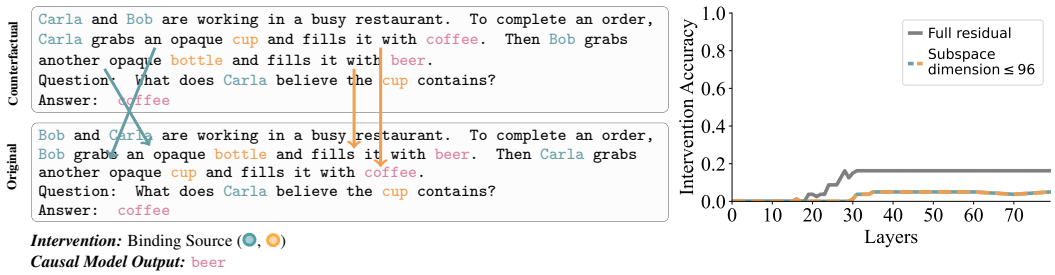

Figure 13: **Source Reference Information** of Binding lookback: In this interchange intervention experiment, the source information, i.e., the character and object OIDs (●, ●), is modified, while the address and payload (●, ●, ▲) are recomputed based on the modified source. Since both the address and pointer information are derived from the altered source, the binding lookback ultimately retrieves the same original state token as the payload. As a result, we do not observe high intervention accuracy.

In section 5.2, we identified the source of the information but did not fully determine the locations of each character and object OI. To address this, we now localize the character and object OIs separately to gain a clearer understanding of the layers at which they appear in the residual streams of their respective tokens, as shown in Fig.14 and Fig.15.

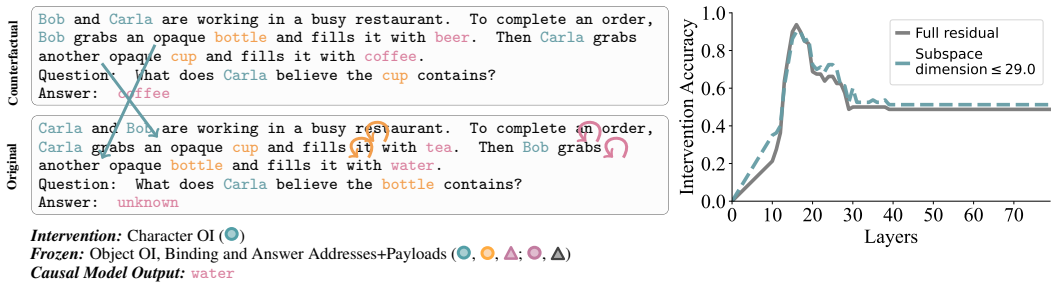

Figure 14: **Character OI**: This interchange intervention experiment swaps the character OI (), while freezing the object OI as well as binding lookback address and payload (, ,). Swapping the character OIs in the story tokens changes the queried character OI to the other one. Hence, the final output changes from *unknown* to **water**.

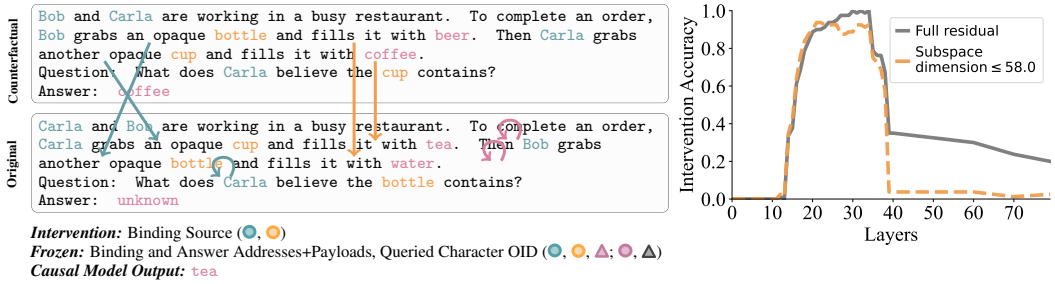

Figure 15: **Object OI**: This interchange intervention experiment swaps both the character and object OIs (,), while freezing the address and payload of binding lookback (, ,) as well as queried character OI (). Swapping both character and object OIs in the story tokens ensures that the queried object gets the other OI. Hence, the final output changes from *unknown* to **tea**.

H Aligning Query Character and Object OIs

In section 5.2, we localized the pointer information of binding lookback. 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 OIs of the queried character and queried object, as shown in Fig. 16 and Fig. 17.

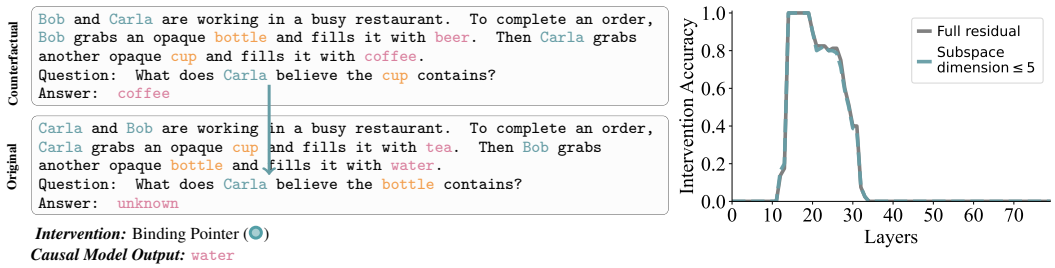

Figure 16: **Query Character OI**: This interchange intervention experiment alters the OI of the queried character () to the other one. Hence, the final output changes from *unknown* to **water**.

I Speculated Payload in Visibility Lookback

As mentioned in section 6, 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

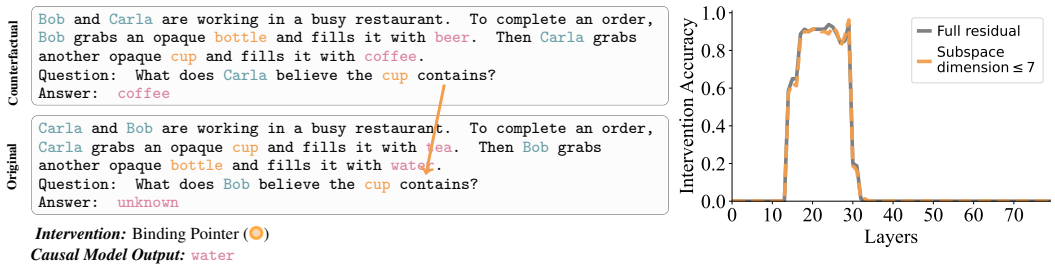

Figure 17: **Query Object OI**: This interchange intervention experiment alters the OI of the queried object ( ) to the other one. Hence, the final output changes from *unknown* to **water**.

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.

If the LM is fetching vital information from the knocked-out sentence, the interchange intervention accuracy (IIA) post-knockout will decrease. Therefore, a decrease 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.

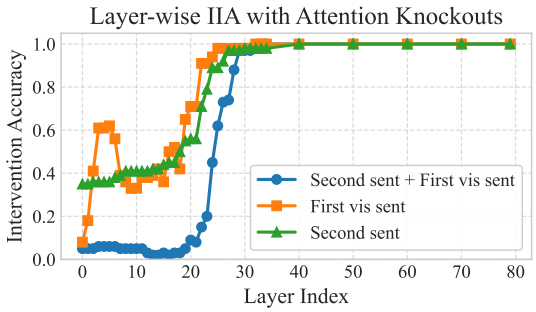

Figure 18: At the second visibility sentence, attention heads are restricted to retrieve information from one of three prior contexts: (1) both the second story sentence and the first visibility sentence (— line), (2) only the first visibility sentence (— line), or (3) only the second story sentence (— line).

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 OIs. Specifically, the second visibility sentence mentions the first and second characters, whose character OIs must be fetched before the model can perform any further operations. Therefore, we believe the decrease in IIA until layer 15, when the character OIs are formed (based on the results from Section G), can be attributed to the model being restricted from fetching the character OIs. 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 OI, which is later used to fetch the correct state OI.

J Correlation Analysis of Causal Subspaces and Attention Heads

This section identifies the attention heads that align with the causal subspaces discovered in the previous sections. Specifically, first we focus on attention heads whose query projections are aligned with the subspaces—characterized by the relevant singular vectors—that contain the correct answer state OI. To quantify this alignment between attention heads and causal subspaces, we use the following computation.

Let $Q \in \mathbb{R}^{d_{\text{model}} \times d_{\text{model}}}$ denote the query projection weight matrix for a given layer:

We normalize Q column-wise:

$$\tilde{Q}_{:,j} = \frac{Q_{:,j}}{\|Q_{:,j}\|} \quad \text{for each column } j \tag{7}$$

Let $S \in \mathbb{R}^{d_{\text{model}} \times k}$ represent the matrix of k singular vectors (i.e., the causal subspace basis). We project the normalized query weights onto this subspace:

$$Q_{\text{sv}} = \tilde{Q} \cdot S \tag{8}$$

We then reshape the resulting projection into per-head components. Assuming $Q_{\text{sv}} \in \mathbb{R}^{d_{\text{model}} \times k}$, and each attention head has dimensionality d_h, we write:

$$Q_{\text{head}}^{(i)} = Q_{\text{sv}}^{(i)} \in \mathbb{R}^{d_h \times k} \quad \text{for } i = 1, \ldots, n_{\text{heads}} \tag{9}$$

Finally, we compute the norm of each attention head's projection:

$$\text{head\_norm}_i = \left\| Q_{\text{head}}^{(i)} \right\|_F \quad \text{for } i = 1, \ldots, n_{\text{heads}} \tag{10}$$

We compute the $head\_norm$ for each attention head in every layer, which quantifies how strongly a given head reads from the causal subspace present in the residual stream. The results are presented in Fig. 19, and they align with our previous findings: attention heads in the later layers form the QK-circuit by using pointer and address information to retrieve the payload during the Answer lookback.

We perform a similar analysis to check which attention heads' value projection matrix align with the causal subspace that encodes the payload of the Answer lookback. Results are shown in Fig. 20, indicating that attention heads at later layers primarily align with causal subspace containing the answer token.

K Belief Tracking Mechanism in BigToM Benchmark

This section presents preliminary evidence that the mechanisms outlined in Sections 5 and 6 generalize to other benchmark datasets. Specifically, we demonstrate that Llama-3-70B-Instruct answers the belief questions (true belief and false belief) in the BigToM dataset Gandhi et al. (2024) in a manner similar to that observed for CausalToM: by first converting token values to their corresponding OIs and then performing logical operations on them using lookbacks. However, as noted in Section 3, BigToM—like other benchmarks—lacks the coherent structure necessary for causal analysis. As a result, we were unable to replicate all experiments conducted on CausalToM. Thus, the results reported here provide only preliminary evidence of a similar underlying mechanism.

To justify the presence of OIs, we conduct an interchange intervention experiment, similar to the one described in Section H, aiming to localize the character OI at the character token in the question sentence. We construct an original sample by replacing its question sentence with that of a counterfactual sample, selected directly from the unaltered BigToM dataset. Consequently, when processing the original sample, the model has no information about the queried character and, as a result, produces unknown as the final output. However, if we replace the residual vector at the queried character token in the original sample with the corresponding vector from the counterfactual

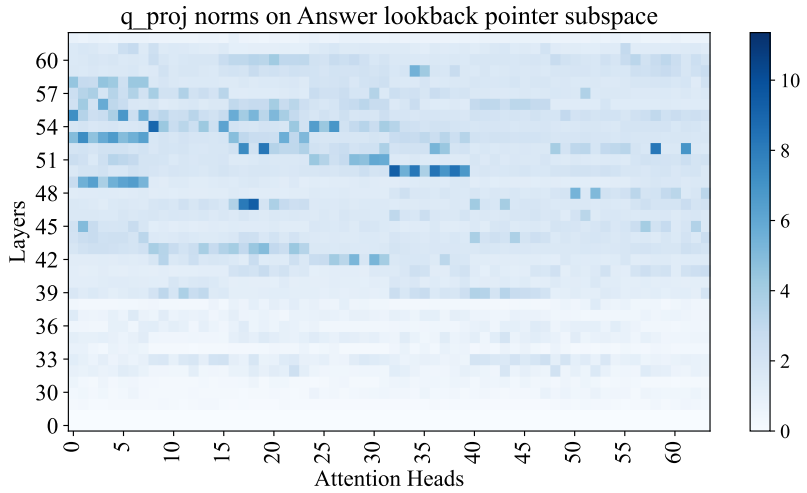

Figure 19: Alignment between the Answer lookback pointer causal subspace and the query projection matrix in Llama-3-70B-Instruct.

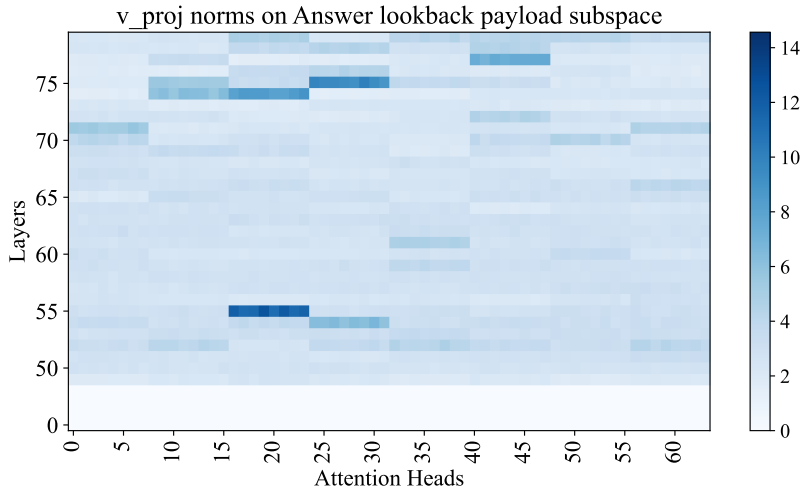

Figure 20: Alignment between the Answer lookback payload causal subspace and the value projection matrix in Llama-3-70B-Instruct.

sample (which contains the character OI), the model's output changes from unknown to the state token(s) associated with the queried object. This is because inserting the character OI at the queried token provides the correct pointer information, aligning with the address information at the correct state token(s), thereby enabling the model to form the appropriate QK-circuit and retrieve the state's OI. As shown in Fig. 21, we observe a high IIA between layers $9 - 28$—similar to the pattern seen in CausalToM—suggesting that the queried character token encodes the character OI in its residual vector within these layers.

Next, we investigate the Answer lookback mechanism in BigToM, focusing specifically on localizing the pointer and payload information at the final token position. To localize the pointer information, which encodes the correct state OI, we construct original and counterfactual samples by selecting two completely different examples from the BigToM dataset, each with different ordered states as the correct answer. For example, as illustrated in Fig.22, the counterfactual sample designates the first state as the answer, **thrilling plot**, whereas the original sample designates the second state, **almond milk**. We perform an intervention by swapping the residual vector at the last token position from the counterfactual sample into the original run. The causal model outcome of this intervention is that the model will output the alternative state token from the original sample, **oat milk**. As shown in Fig.22,

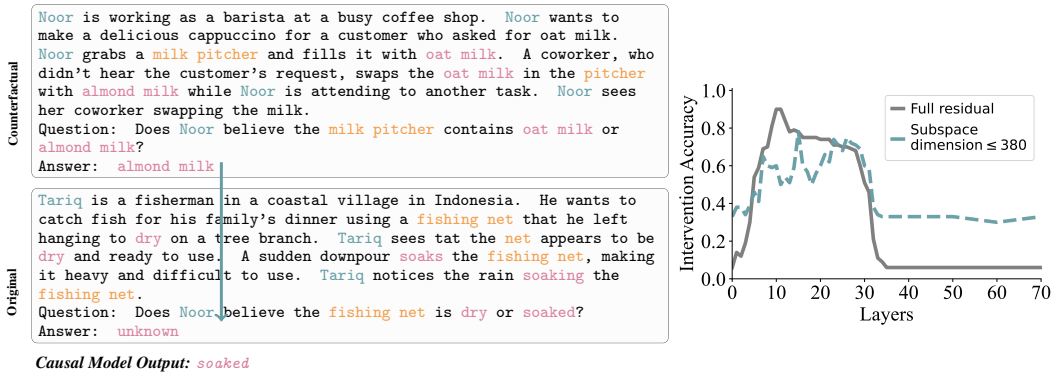

Figure 21: **Query Character OI in BigToM**: This interchange intervention experiment inserts the first character's OI into the residual stream at the queried character token (), resulting in the movement of pointer information to the last token that aligns with the address information of binding lookback mechanism. Consequently, the model is able to form the appropriate QK-circuit from the last token to predict the correct state answer token(s) as the final output, instead of unknown.

this alignment occurs between layers 33 and 51, similar to the layer range observed for the pointer information in the Answer lookback of CausalToM.

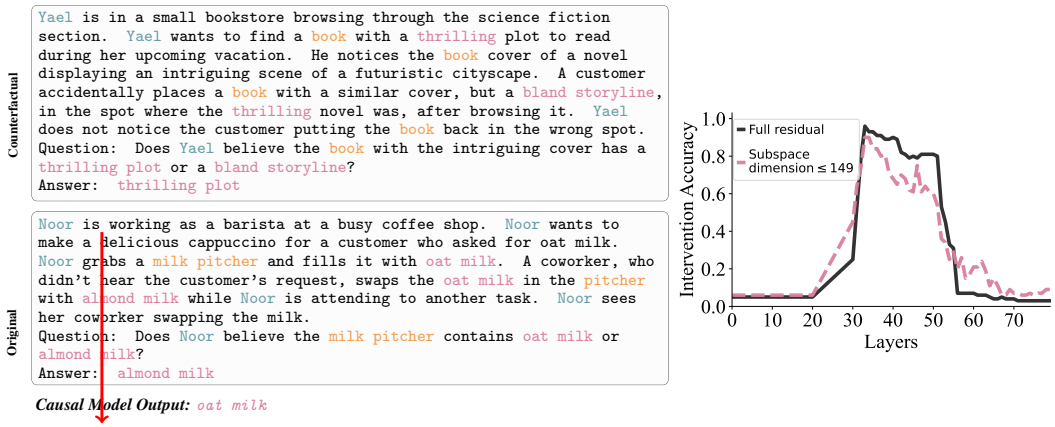

Figure 22: **Answer Lookback Pointer in BigToM**: This interchange intervention experiment modifies the pointer information () of the Answer lookback, thereby altering the subsequent QK-circuit to attend to the other state (e.g., **oat milk**) instead of the original one (e.g., **almond milk**). As a result, the model retrieves the token value corresponding to the other state to answer the question.

Further, to localize the payload of the Answer lookback in BigToM, we perform an interchange intervention experiment using the same original and counterfactual samples as mentioned in the previous experiment, but with a different expected output—namely, the correct state from the counterfactual sample instead of the other state from the original sample. As shown in Fig. 23, alignment emerges after layer 59, consistent with the layer range observed for the Answer lookback payload in CausalToM.

Finally, we investigate the impact of the visibility condition on the underlying mechanism and find that, similar to CausalToM, the model uses the Visibility lookback to enhance the observing character's awareness based on the observed character's actions. To localize the effect of the visibility condition, we perform an interchange intervention in which the original and counterfactual samples differ in belief type—that is, if the original sample involves a false belief, the counterfactual involves a true belief, and vice versa. The expected output of this experiment is the other (incorrect) state of the original sample. Following the methodology in Section 6, we conduct three types of interventions: (1) only at the visibility condition sentence, (2) only at the subsequent question sentence, and (3) at

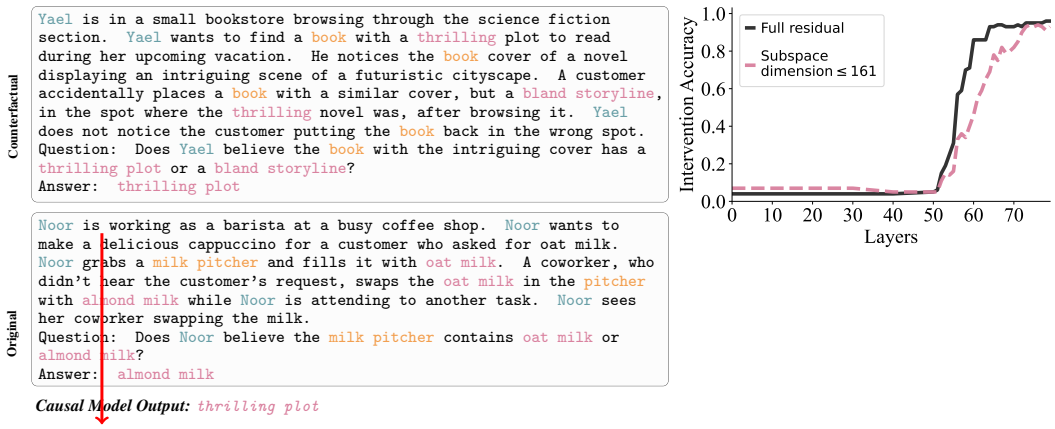

Figure 23: **Answer Lookback Payload in BigToM**: This interchange intervention experiment directly modifies the payload information (▲) of the Answer lookback, which is fetched from the corresponding state tokens and predicted as the next token(s). Thus, replacing its value in the original run, e.g. **almond milk**, with that from the counterfactual run, e.g. **thrilling plot**, causes the model's next predicted tokens to correspond to the correct answer of the counterfactual sample.

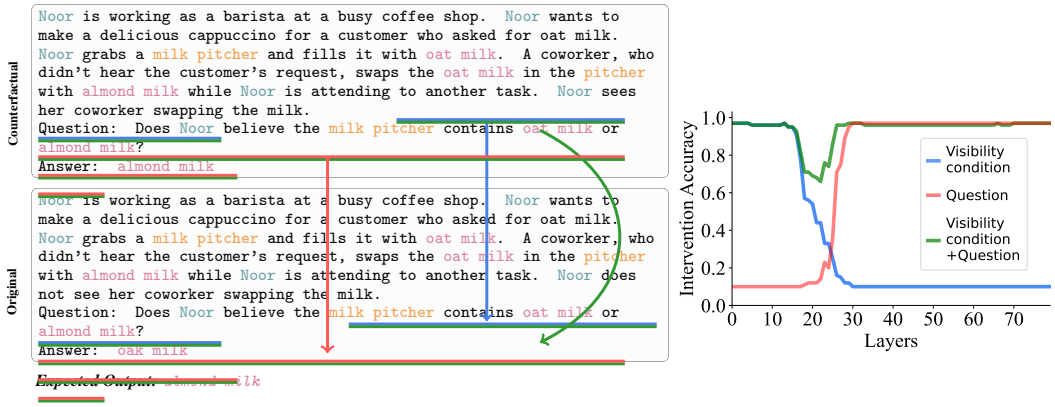

Figure 24: **Visibility Lookback in BigToM**: We perform three interchange interventions to establish the presence of the Visibility ID, which serves as both address and pointer information. When intervening at the source (●)—i.e., the visibility sentence—both the address and pointer are updated, resulting in alignment across layers. Intervening only at the subsequent question tokens leads to alignment only at later layers, after the model has already fetched the payload (▲). However, intervening at both the visibility and question sentences results in alignment across all layers, as the address and pointer remain consistent throughout.

both the visibility condition and the question sentence. As shown in Fig. 24, intervening only at the visibility sentence results in alignment at early layers, up to layer 17, while intervening only at the subsequent question sentence leads to alignment after layer 26. Intervening on both the visibility and question sentences results in alignment across all layers. These results align with those found in the CausalToM setting shown in the Fig. 8.

Previous experiments suggest that the underlying mechanisms responsible for answering belief questions in BigToM are similar to those in CausalToM. However, we observed that the subspaces encoding various types of information are not shared between the two settings. For example, although the pointer information in the Answer lookback encodes the correct state's OI in both cases, the specific subspaces that represent this information at the final token position differ significantly. We leave a deeper investigation of this phenomenon—shared semantics across distinct subspaces in different distributions—for future work.

L Generalization of Belief Tracking Mechanism on CausalToM to Llama-3.1-405B-Instruct

This section presents all the interchange intervention experiments described in the main text, conducted using the same set of counterfactual examples on Llama-3.1-405B-Instruct, using NDIF Fiotto-Kaufman et al. (2025). Each experiment was performed on 80 samples. Due to computational constraints, subspace interchange intervention experiments were not conducted. The results indicate that Llama-3.1-405B-Instruct employs the same underlying mechanism as Llama-3-70B-Instruct to reason about belief and answer related questions. This suggests that the identified belief-tracking mechanism generalizes to other models capable of reliably performing the task.

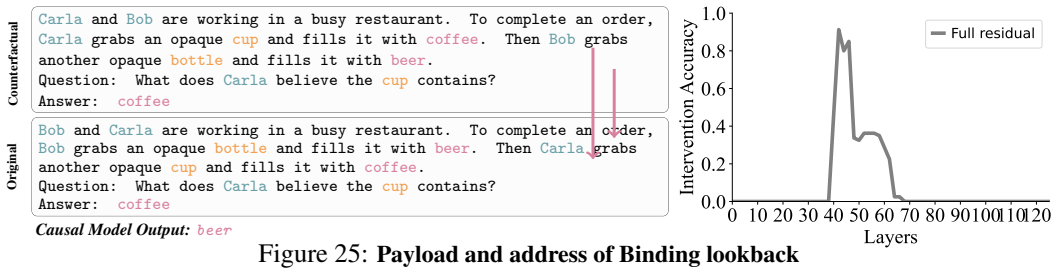

Figure 25: **Payload and address of Binding lookback**

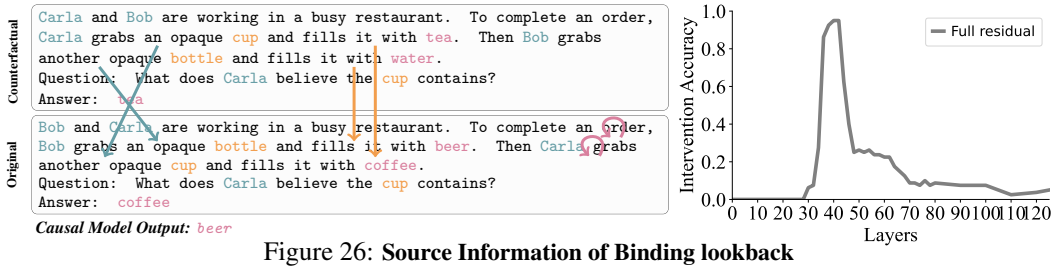

Figure 26: **Source Information of Binding lookback**

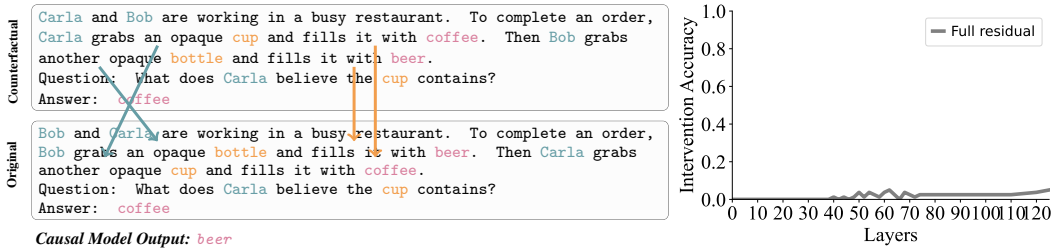

Figure 27: **Source Reference Information of Binding lookback without freezing address and payload**

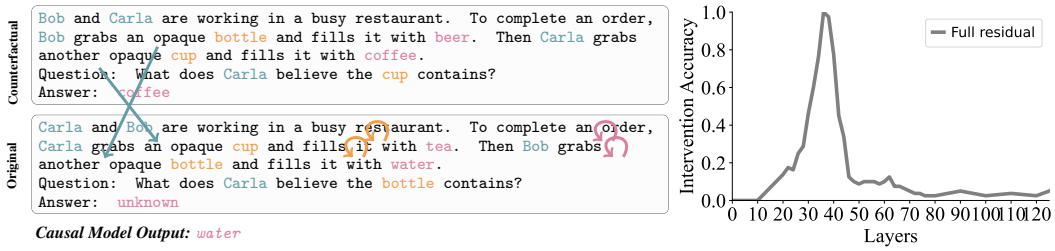

Figure 28: **Character OI**

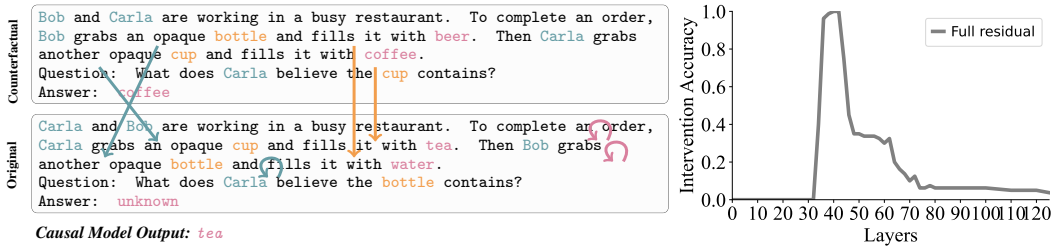

Figure 29: **Object OI**

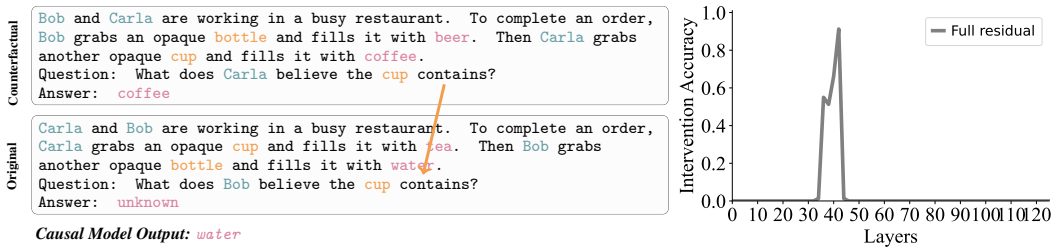

Figure 30: **Query Object OI**

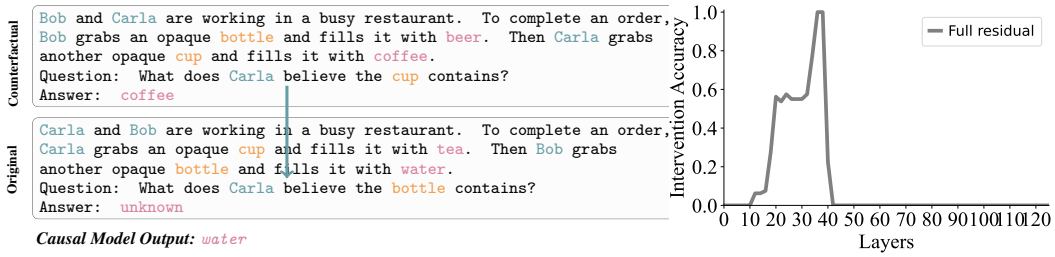

Figure 31: **Query Character OI**

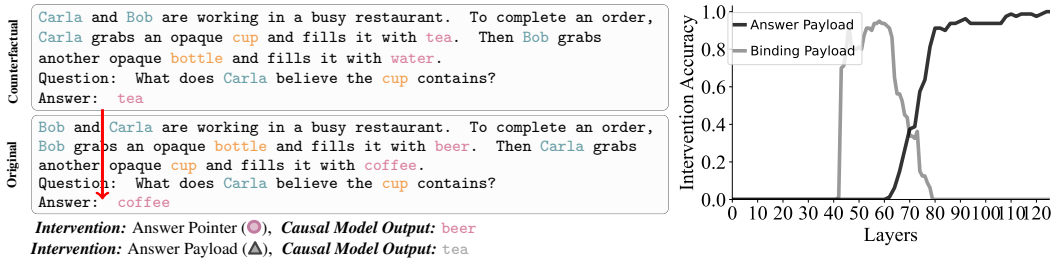

Figure 32: **Answer Lookback Pointer and Payload**

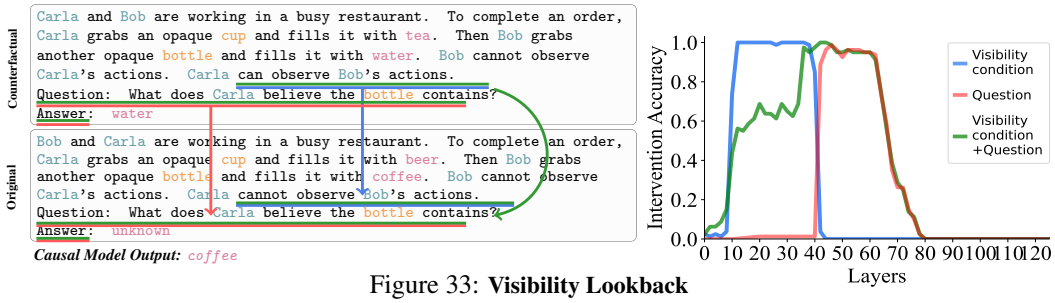

Figure 33: **Visibility Lookback**