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.

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).

Correspondence to prakash.nik@northeastern.edu.

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).

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.

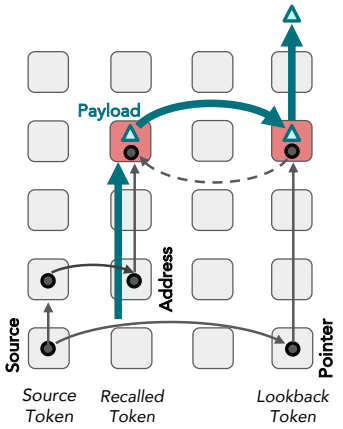

Figure 1: **The lookback mechanism** performs conditional reasoning; The *source token* contains reference information that is copied into two instances, creating a *pointer* and an *address*. Next to the address in the residual stream is a *payload*. When necessary, the model retrieves the payload by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

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*.[1] 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*.

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

[1]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.

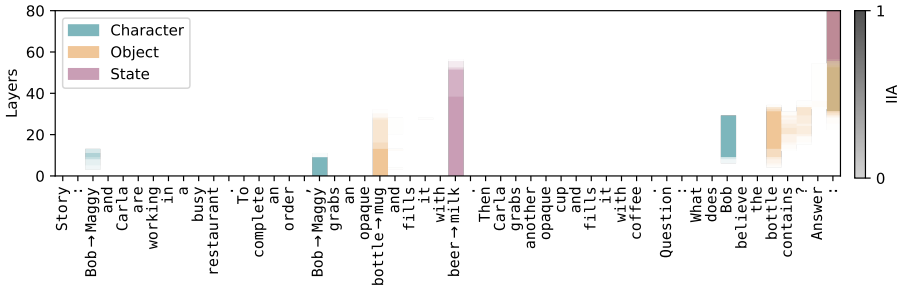

Figure 2: **Tracing information flow** of crucial input tokens using causal mediation analysis.

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 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.,