Extracting Belief-Update Rules to Explain Theory-of-Mind Generalization Failures

Anonymous Author(s)
Affiliation
Address
email

Abstract

We study whether neural models learn generalizable belief-updating rules in a competitive Theory of Mind (ToM) task. Using the Standoff competitive-feeding environment, we compare a deterministic, modular ToM baseline against end-to-end transformer models. While hardcoded models produce interpretable, rule-based belief updates, neural models learn approximations that overfit, exhibiting systematic errors on unseen opponent knowledge states. Through qualitative analysis of belief state update rules, we identify systematic failure modes including violations of object symmetry, temporal invariance, and egocentric bias.

1 Introduction

In the classic Sally-Anne task, children watch Sally hide a marble and leave the room. Anne moves the marble to a new location, and Sally returns. When asked where Sally will look for the marble, children on the autism spectrum often point to the marble's actual location rather than where Sally last saw it [1]. This egocentric bias, projecting one's own knowledge onto others, reveals a fundamental challenge in distinguishing others' beliefs from one's own.

Previous computational work on a supervised learning theory of mind task found a consistent pattern: neural network models learned to predict opponent behavior for familiar tasks, but were unable to generalize their predictions to cases involving novel kinds of mental states in those opponents. Swapping in hardcoded modular components revealed an asymmetry: first person inferences (such as what the player sees or knows) generalized well to novel scenarios, but third person inferences of mental states did not. Enforcing equivalence between first and third person reasoning did not improve generalization.

Such computational models of ToM reasoning perform well on competitive feeding tasks when specific structure is imposed, while end-to-end models struggle to generalize to novel tasks. These models overfit to their training data, learning some representation of beliefs that happens to correspond to their opponents' behavior but does not generalize. Without understanding the syntax of learned behavior functions, we cannot know what errors such models must overcome to learn truly generalizable ToM capabilities.

In this paper, we: (1) present a belief-update rule analysis method that extracts belief-state graphs from model outputs; (2) identify three specific causes of generalization failure observed in our transformer models independent of learning targets; (3) describe architectural interventions to address two of the three failure modes.

Submitted to 39th Conference on Neural Information Processing Systems (NeurIPS 2025). Do not distribute.

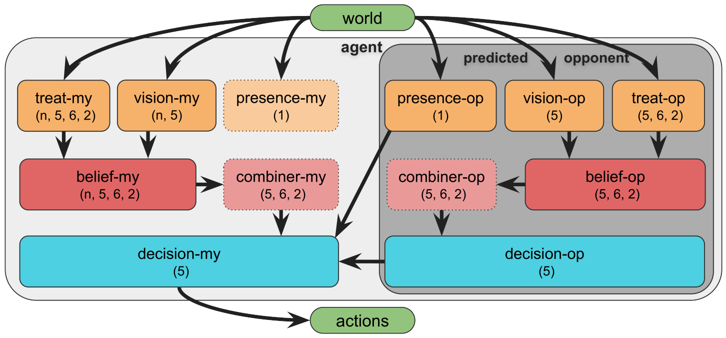

Figure 1: The hardcoded model architecture used in this study. Inputs are processed to produce: tensors indicating the treats visible at each of 5 timesteps and 6 positions (including null) of both the large and small treat (**treat**), whether the opponent's gaze is obscured at each timestep (**vision**), and whether either player is present (**presence**). **Belief** modules use the former two outputs to predict the treats' locations at the final timestep. **Combiner** modules combine n multiple uncertain beliefs into one. **Decision** modules use a belief vector and (only as the subordinate player) the opponent's decision to predict the location harboring the largest available treat.

2 Related Work

This work is heavily inspired by the ToMnet experiments of Rabinowitz et al. [7]. In their study, they implement machine learning models with explicit ToM-like representations about agents' attributes and mental states, and are able to leverage the computational setting to probe those models for representations of those features.

Recently, Horschler et al. [3] used computational modeling to investigate ToM capabilities in non-human primates, focusing on visual perspective-taking tasks similar to the one investigated by this paper. They developed seven models of varying complexity to represent different theories of primates' social cognition, and parameterize the subjects' reliance on their ToM inferences to determine how well the theories explain primate behavior.

Computational ToM skills have also been particularly well-studied recently in the context of large language models (LLMs). The ToMi dataset by Le et al. [4] consists of short, structured narratives based on the Sally-Anne false belief test. ToMi focuses primarily on first-order ToM reasoning about physical world states. Xu et al. developed OpenToM [8] to benchmark ToM capabilities in large language models using longer narratives, covering both physical and psychological aspects of ToM. Despite recent advancements, LLMs continue to underperform humans on complex ToM tasks, highlighting the difficulty in acquiring robust ToM skills in machine learning models.

2.1 Environment

The competitive feeding paradigm is a test setup designed to distinguish whether a non-verbal subject will change its behavior to account for what it believes someone else (an "opponent") *knows*, based on evidence relating to what it can perceive that the opponent *sees* [2].

In this paper, we use the Standoff environment [citation omitted], a gridworld setting that replicates the competitive feeding paradigm in the style of Penn and Povinelli [6]. Competitive feeding demands that agents reason about what an opponent believes and how those beliefs will drive strategic behavior in a competitive context. In Standoff tasks, two treats of different sizes are visibly hidden in any of five boxes, which are then shuffled around. The player's challenge is to select the box containing the best possible treat.

The environment creates four distinct opponent knowledge states regarding either of the two treats. Informed opponents have seen all changes and hold accurate beliefs about treat locations. Uninformed opponents never observed certain treats being placed and will pursue known treats or default to deterministic choices if they know about neither. Misinformed opponents hold false beliefs from seeing treats placed that were later swapped while vision was blocked. Gettier cases occur when opponents happen to be correct despite having been wrong earlier, so their belief is accidentally true.

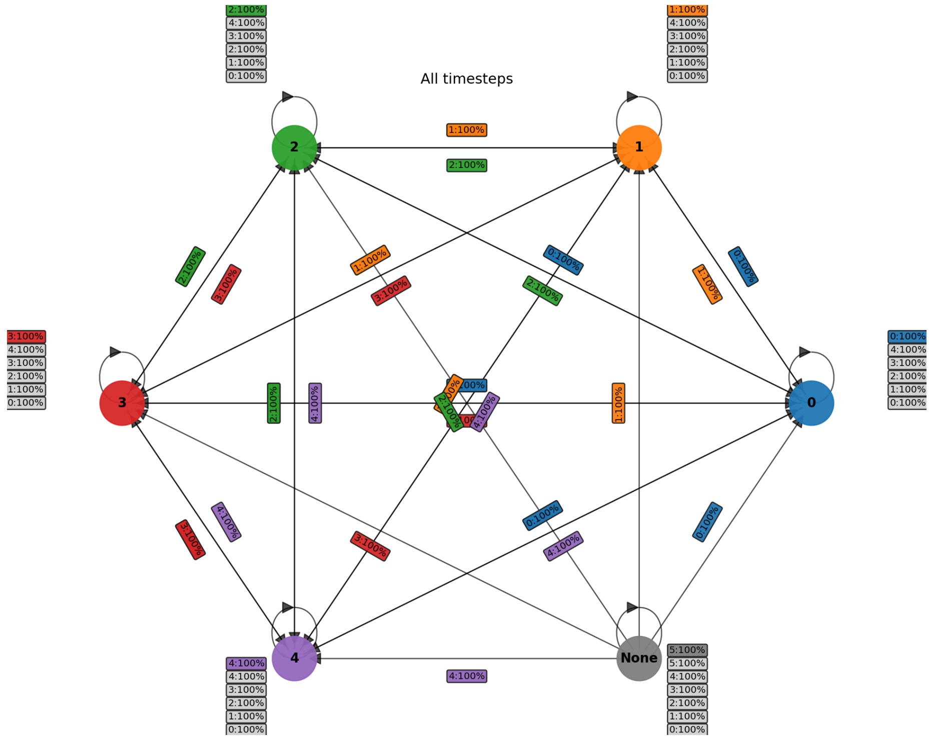

Figure 2: Belief-update rules of the hardcoded theory of mind model. Nodes 0-4 represent beliefs that a treat is in one specific state, while None represents not having observed a treat before. Edge labels describe a treat being visible in the environment at some location (0-4, with 5 being no observation), which happens when treats are placed or swapped. Percentages indicate probabilities that specific observations result in the shown transitions; for the hardcoded model, these are deterministic. Light gray edges represent observations that are occluded from the opponent's view; none of these edges should update the opponent's beliefs. All models begin each trial in the None state for both treats.

In this paper, we train models models on all but misinformed and Gettier cases, setting those aside as the core generalization challenge.

2.2 Learning Targets

Inputs to the models are (7x7x5x5) videos of the environment. They are processed by hardcoded perceptual modules for both the hardcoded model and the transformer. Outputs of perceptual modules include opponent presence, treat locations at each timestep, and whether the opponent's vision is obscured at each timestep. We train to minimize categorical cross entropy on four outputs: **my decision**, a 5-length vector of the player's optimal choice, given the opponent's decision. **opponent decision**, a (5, 6) shaped tensor of the opponent's decision at each of 5 timesteps, with the last position reserved for no decision (which occurs when no opponent is present). **my belief**, a (5, 6) shaped tensor describing the player's beliefs about the treats' locations at each timestep given their previous observations. Like opponent decision, the last position is reserved for null beliefs. Belief vectors begin as null. **opponent belief**, a (5, 6) shaped tensor describing the same as above for the opponent, who may not have viewed some timesteps.

2.3 Models

As a baseline for comparison, we use a hardcoded solution to the Standoff environment presented in [5]. This modular architecture implements simulation theory's premise that agents can use self-models to predict others' mental states by using identical modules for both self- and opponent-reasoning. Our transformers embed their inputs at each timestep into 128-dimension hidden states. A 2-layer TransformerEncoder with 4 attention heads, GELU activations, dropout of 0.1, and a causal attention

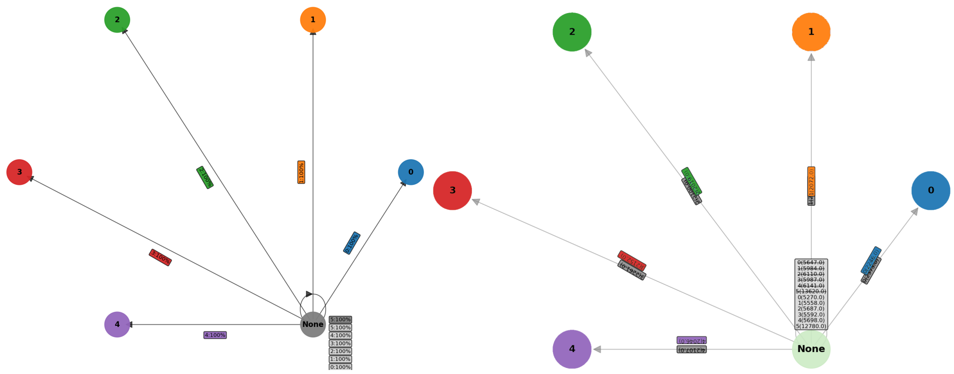

Figure 3: State transition networks for the second timestep, only for cases with present opponents. In the first timestep, treats are not placed. Left: the hardcoded model beliefs only change when a treat placement is visible to the opponent, otherwise (or if no treat is placed), the opponent belief remains None. Right: a transformer model learned to incorrectly attribute changed opponent beliefs when the opponent did not observe treats being placed, roughly half the time. This pattern was learned despite training the transformer to predict ground truth beliefs in addition to the player's behavior.

mask encodes the timestep data. The last timestep is pooled for final linear heads, each of which decodes the data for one of the learning targets: opponent per-timestep beliefs, my per-timestep beliefs, opponent's per-timestep decisions, and my decision. We train 5 models with different seeds to predict all four of the outputs referenced above, for 5k batches of size 1024, using the AdamW optimizer with default parameters.

2.4 Belief State Updates

At evaluation, over the full dataset, we convert each per-timestep belief vector into a single chosen location by taking the argmax over six classes (locations 0–4 and "None" = 5). For each timestep and for each treat (large and small) we record four values: the prior belief location (None at the first timestep), the observed location of the treat at that timestep (0–4 or 5 if it is not observed), the model's current belief, and whether the opponent could see that timestep. We accumulate counts of all prior-location × observed-location × current-location triples into a 6×6×6 table separately for each timestep, treat size, and vision condition.

3 Results and Discussion

Figure 2 shows the hardcoded model's belief-transition graph.

Our transformer models achieve high accuracy (>99.5% on novel in-distribution tasks) but lower accuracy (≈75%) on tasks from our misinformed and Gettier test set. These models learned different distributions of belief state update rules with minor variations. Qualitative analysis reveals three consistent failure modes across all our transformer models. First, the transformers learned asymmetric belief update patterns for large versus small treats despite identical ground truth rules. Swapping treats during training and minimizing belief updates based on both outputs eliminated this asymmetry. Second, they also exhibited varying transition probabilities across timesteps. We were able to address this problem to some extent by extending the length of the video and randomly shifting timesteps during training. Third, models learned an egocentric bias, shown in Figure 3: all transformers incorrectly updated opponent beliefs during vision occlusion approximately 50% of the time, projecting their own observational access onto opponents who cannot see state changes. Because the training data lacked misinformed and Gettier cases, this incorrect inference had no counterexamples and the models were not incentivized to learn the more robust solution.

Neural networks trained on ToM tasks learn belief update mechanisms that systematically conflate self and other perspectives, failing to generalize beyond familiar scenarios. These computational limitations highlight the need for architectural constraints that enforce compositional belief representations rather than learned approximations.

Reproducibility

Code and data to reproduce experiments will be released at submission including configs, seeds, and model checkpoints.

References

[1] Simon Baron-Cohen, Alan M. Leslie, and Uta Frith. Does the autistic child have a theory of mind? *Cognition*, 21(1):37–46, October 1985. ISSN 0010-0277. doi: 10.1016/0010-0277(85)90022-8.

[2] Brian Hare, Josep Call, Bryan Agnetta, and Michael Tomasello. Chimpanzees know what conspecifics do and do not see. *Animal Behaviour*, 59(4):771–785, 2000.

[3] Daniel J Horschler, Marlene D Berke, Laurie R Santos, and Julian Jara-Ettinger. Differences between human and non-human primate theory of mind: Evidence from computational modeling. *bioRxiv*, pages 2023–08, 2023.

[4] 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)*, pages 5872–5877, 2019.

[5] Joel Michelson, Deepayan Sanyal, and Maithilee Kunda. A modular framework for analyzing theory of mind learning in competitive tasks. In *Proceedings of the Annual Meeting of the Cognitive Science Society*, volume 47, 2025.

[6] Derek C Penn and Daniel J Povinelli. On the lack of evidence that non-human animals possess anything remotely resembling a 'theory of mind'. *Philosophical Transactions of the Royal Society B: Biological Sciences*, 362(1480):731–744, 2007.

[7] Neil Rabinowitz, Frank Perbet, Francis Song, Chiyuan Zhang, SM Ali Eslami, and Matthew Botvinick. Machine theory of mind. In *International conference on machine learning*, pages 4218–4227. PMLR, 2018.

[8] 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. *arXiv preprint arXiv:2402.06044*, 2024.