
Supplementary for Mindmaster Roleplay: **A Social Reasoning and Planning Benchmark**

Anonymous Author(s)

Affiliation

Address

email

1 Overview

This supplementary material provides comprehensive details on our experimental methodology, data collection procedures, and analytical frameworks designed to explore the intricacies of social cognition and agent interaction. The material is organized into five major sections, each elaborating on crucial components of our research infrastructure and findings.

Section 2 delineates the technical specifications of our experimental platform, which implements a structured environment for studying multi-agent interactions. The platform incorporates precise spatial representations, object affordances, belief modeling, value dimensions, intention formalization, and action capabilities. Through this framework, we systematically investigate the mechanisms underlying social intelligence within controlled virtual environments.

Section 3 details our rigorous data collection methodology, which employed a comprehensive onboarding process including detailed tutorials and qualifying assessments to ensure participant competency. Our human study protocol, approved by the Institutional Review Board, involved 223 participants generating 388 rounds of valid paired interaction data. The collected data underwent systematic processing to extract variables in both machine-readable and human-accessible formats, with statistical analyses confirming balanced distributions of agent values and diverse interaction patterns.

Section 4 outlines our experimental design for evaluating various language models on social cognition tasks. We provide specifics regarding model configurations (Gemini, GPT-4o, Deepseek-R1, Claude, Qwen3-8B, and Llama) and implementation details. Our methodological approach incorporates non-zero temperature sampling and trajectory-based train-test splitting to enable nuanced analysis of model reasoning capabilities beyond deterministic predictions.

Section 5 explicates our prompt engineering framework, detailing the transformation of complex environmental states into structured representations comprehensible to language models. We present the exact templates used for our six experimental tasks: intention estimation (partial and full observation), value estimation (partial and full observation), intention updating, and social interaction policy, demonstrating how our approach enables effective reasoning about mental states and action planning.

Finally, Section 6 presents a comprehensive analysis of our findings, comparing language model performance against human benchmarks across four key domains of social cognition. Through detailed visualizations and statistical analyses, we demonstrate how different models exhibit distinct biases and performance characteristics, quantified through accuracy metrics, confidence scores, and correlation coefficients that precisely evaluate each model's strengths and limitations in social reasoning.

This supplementary material thus provides a complete foundation for understanding both the methodological rigor and the analytical depth of our investigation into artificial social intelligence and its alignment with human cognitive processes.

38 2 Platform details

39 This section delineates the technical specifications and operational framework of our experimental
40 platform designed for studying agent interactions within controlled virtual environments. The plat-
41 form implements a comprehensive system encompassing spatial properties, object affordances, be-
42 lief representation, value dimensions, intention formalization, and action capabilities. Through care-
43 fully designed scenarios and a structured cognitive process model, the platform enables systematic
44 investigation of social cognition, intention recognition, and cooperative behavior while maintaining
45 ecological validity. The following subsections provide detailed information about each component
46 of this experimental framework, their theoretical underpinnings, and implementation details that
47 collectively facilitate rigorous examination of agent interactions across diverse contextual settings.

48 2.1 Agent/Object property

49 **Position:** In the game environment, position represents the coordinates of the center points of agents
50 and objects in a Cartesian coordinate system. The coordinate range spans from -700 to 700 with
51 the structure $[x, y]$.

52 **Rotation:** Rotation indicates the directional orientation of agents and objects, with values ranging
53 from -1 to 1 . Specifically, -1 corresponds to the negative x-axis direction, and as one rotates coun-
54 terclockwise, the value increases to 0 at the positive x-axis and further to 1 when returning to the
55 negative x-axis. Due to PyGame’s implementation, objects appear in their natural orientation (e.g.,
56 tables perpendicular to the x-axis) when rotation equals 0.5 .

57 **Size:** Size denotes the spatial dimensions that an object occupies in its natural orientation (rotation
58 $= 0.5$), expressed as (x_range, y_range) . This corresponds to the width and height in the rendered
59 image. Specially, agents are circles in the game and their size is a number representing the radius.
60 When rendering agents and objects, they are drawn with the position as the center point, ensuring
61 equal distances from the center to all edges.

62 2.2 Objects and affordance

63 **Tab. 1** presents an overview of objects utilized in our experimental environment and their correspond-
64 ing affordances. Objects include chess (playable by two participants and manipulable), box (capable
65 of containing items and alternating between open and closed states), cabinet (similarly offering
66 containment functionality with open/close capability, and can be locked), banana (edible and man-
67 ipulable), cup (breakable and manipulable), key (functional for unlocking boxes and manipulable),
68 table (providing a support surface for objects), and several manipulable items (books, timer, and
69 dumbbell). These defined affordances provide various interaction types between agents and objects
70 within the experimental framework, creating a controlled environment for studying object-agent
71 interactions and intention completion behaviors.

72 2.3 Belief world

73 The sector shown in **Figure 2** represents the agent’s attention direction and field of view. Although
74 only a single sector is depicted, in practice, the agent’s attention covers the entire area between the
75 two rays, and the agent can perceive targets located as far away as possible within this visible range.
76 Only those objects that are perceived by and incorporated into the agent’s belief world appear on the
77 screen; objects not displayed on screen do not necessarily indicate nonexistence, but rather have not
78 yet been perceived and integrated into the agent’s beliefs. Brightly colored objects represent what
79 the agent currently sees, while semi-transparent objects indicate entities that, although not currently
80 within the agent’s visual field, have been previously perceived and remain part of the agent’s belief
81 representation.

82 2.4 Value dimensions

83 At the beginning of the game, participants are asked to role-play based on a given role configuration,
84 which includes three value dimensions: helpful, social, and active, representing the tendencies to be
85 helpful, socially outgoing, and industrious, respectively.










Objects	Affordance	Objects	Affordance
 chess	playable by two players; graspable	 banana	edible; graspable
 box	containable; openable; closable; lockable	 cup	breakable; graspable
 cabinet	storable; openable; closable	 key	insertable; rotatable; graspable
 table	supportive; stable	 books	readable; stackable; graspable
 timer	settable; activatable; graspable	 dumbbell	liftable; weighable; graspable

Table 1: Objects and their affordance

86 The three icons shown in the figure 2, from left to right, represent the agent’s status along the
87 three value dimensions: “helpful,” “social,” and “active.” Specifically, the left icon corresponds to
88 the “helpful” dimension, indicating how willing the agent is to help others; the middle icon corre-
89 sponds to the “social” dimension, indicating how socially outgoing the agent is; and the right icon
90 corresponds to the “active” dimension, indicating how energetic and active the agent is. Each value
91 dimension has 3–4 possible discrete states:

- 92 • **Active Value:** This dimension measures an agent’s energy and inclination towards physical
93 activity. The possible scores are:
 - 94 – 0 (inactive): no energy, no inclination for activity.
 - 95 – 0.5 (neutral): moderate energy, neither active nor inactive preference.
 - 96 – 1 (active): high energy and a strong preference for physical motion.
- 97 • **Social Value:** This dimension captures an agent’s preference for social interaction. The
98 scores are:
 - 99 – 0 (unsocial): avoids social interaction, prefers solitude.
 - 100 – 0.5 (neutral): no strong preference regarding social interaction.
 - 101 – 1 (social): actively seeks social engagement.
- 102 • **Helpful Value:** This value dimension assesses an agent’s willingness to help others. The
103 scores are:
 - 104 – -1 (harmful): tends to hinder others’ progress.
 - 105 – 0 (unhelpful): neutral, neither helping nor hindering.
 - 106 – 0.5 (neutral): moderate willingness to help.
 - 107 – 1 (helpful): willing to help others to some degree.
 - 108 – 2 (very helpful): highly inclined to assist others.

2.5 Intention space

In some cases, participants may also be assigned an initial intent (though it is not mandatory). The intention space in our framework is formalized as a structured representation of agent intentions through predicate-argument structures. Each intention is represented as a tuple consisting of a predicate (action type) and its corresponding arguments (entities involved). Our intention space is listed below:

- [PutOnto/PutInto, <object>, <destination>]: Represents the intention to place an object onto or into a specified destination.
- [Give, <object>, <recipient>]: Denotes the intention to transfer possession of an object to a recipient.
- [Get, <object>, <source>]: Indicates the intention to obtain or retrieve an object from a specified source, which may be a location or another agent.
- [Find/Open/Observe, <object>]: Represents intentions related to locating, accessing, or examining objects in the environment.
- [PlayWith, <object>, <partner>]: Captures the social intention of engaging in play involving an object with another agent.
- [RespondTo/Greet, <agent>]: Represents reactive social intentions directed toward another agent.
- [Inform, <agent>, <information>]: Denotes the communicative intention to convey specific information to another agent.
- [Help, <agent>, <intention>]: Represents the cooperative meta-intention to assist another agent in achieving their own intention.
- [RequestHelp, <agent>, <intention>]: Indicates the social intention to solicit assistance from another agent to fulfill a specific intention.
- [Harm, <agent>, <intention>]: Represents the adversarial meta-intention to impede another agent from achieving their intention.
- [Na]: Denotes the absence of any identifiable intention.

This structured intention space enables our framework to model a diverse range of goal-directed behaviors, from basic object manipulation to complex social interactions. The hierarchical nature of the representation, particularly evident in meta-intentions like `Help` and `Harm`, allows for the modeling of recursive intention structures that are characteristic of sophisticated social reasoning.

2.6 Action space

Our framework employs a structured action space that enables the agent to interact with the environment in a physically realistic manner. The action space encompasses a diverse set of primitive actions that can be categorized into navigation, manipulation, social interaction, and communication actions. Each action is represented as a list where the first element specifies the action type, followed by necessary arguments indicating the involved entities.

• Navigation and orientation actions:

- ['ActionMoveTo', <entity>]: Navigate to the specified entity
- ['ActionRotateTo', <entity>]: Rotate to face the specified entity
- ['ActionPointTo', <entity>]: Point toward the specified entity
- ['ActionFollowPointing', <agent>]: Follow the pointing gesture of another agent
- ['ActionMoveToAttention', <agent>]: Move to the attention focus of another agent

• Manipulation actions:

- ['ActionGrab', <object>]: Grasp the specified object
- ['ActionPutDown', <object>]: Put the held object onto ground

- 157 - ['ActionPutInto', <object1>, <object2>]: Place object1 inside ob-
- 158 ject2
- 159 - ['ActionPutOnto', <object1>, <object2>]: Place object1 on top of
- 160 object2
- 161 - ['ActionGiveTo', <object>, <agent>]: Transfer an object to another
- 162 agent
- 163 - ['ActionOpen', <object>]: Open an openable object
- 164 - ['ActionClose', <object>]: Close an openable object
- 165 - ['ActionUnlock', <object>]: Unlock a lockable object with a held key
- 166 - ['ActionPlay', <object>]: Interact with a playable object
- 167 • **Consumption actions:**
- 168 - ['ActionPerform', 'eat' / 'drink']: Perform eating or drinking actions
- 169 - ['ActionEat', 'banana']: Eat a held banana
- 170 - ['ActionSmash', 'cup']: Destroy a held cup
- 171 • **Social communication actions:**
- 172 - ['ActionWaveHand', <agent>]: Wave hand as a greeting gesture
- 173 - ['ActionNodHead', <agent>]: Nod head to indicate agreement
- 174 - ['ActionShakeHead', <agent>]: Shake head to indicate disagreement
- 175 - ['ActionSpeak', <utterance('hello', 'thank you')>]: Verbal-
- 176 ize a predefined utterance
- 177 • **Other actions:**
- 178 - ['ActionWait']: Stay in the current state without taking action

179 This hierarchical action space enables the agent to perform complex interactions through combi-
 180 nations of these fundamental actions. The structured format facilitates both learning and planning
 181 processes, as well as the interpretability of the agent's behavior.

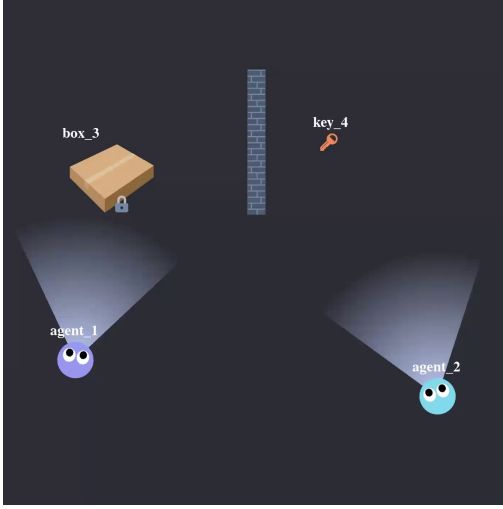
182 2.7 Scenarios

183 The scenarios and tasks in *Mindmaster Roleplay* are constructed from two main sources to ensure
 184 both ecological validity and breadth of coverage. Firstly, several scenarios are inspired by classic ex-
 185 periments in cognitive psychology, allowing for targeted investigation of specific cognitive skills and
 186 abilities. Examples include tasks analogous to studies on *Chimpanzee* [Melis and Tomasello, 2019],
 187 where Sultan the chimpanzee demonstrated insightful learning by joining two short sticks to retrieve
 188 a distant banana. Secondly, to complement these targeted scenarios, we employ a methodology for
 189 generating a large number of diverse, random scenarios that nevertheless adhere to predefined envi-
 190 ronmental and task constraints. This dual approach facilitates a comprehensive examination of agent
 191 behavior across a wide spectrum of situations.

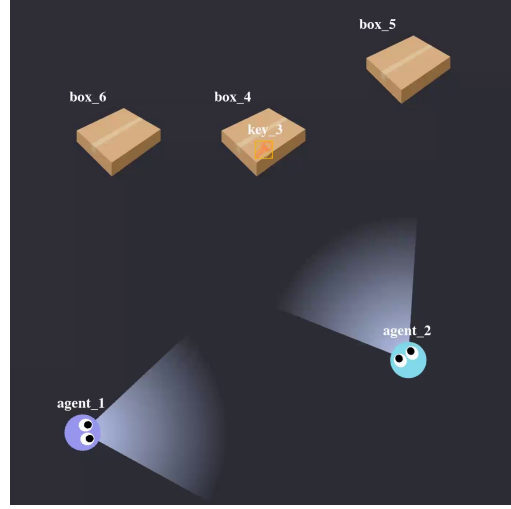
192 2.8 Scenario generation

193 Random scenarios are generated to ensure the diversity of the collected data. Here is the outline of
 194 how we generate random scenarios:

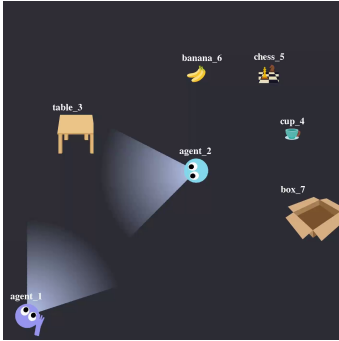
- 195 • Step 1: Randomize the global intent setting. The global intent settings include the following
- 196 three settings: two distinct no-NA and no conflicting intents; two intents but conflicting
- 197 one with the other(both no-NA); one no-NA intent and one NA intent. (Conflicting intents
- 198 represent the situation that the achievement of both intents requires two different final states
- 199 of the same object, which is impossible.)
- 200 • Step 2: Randomly attribute intent predicates. The attributed intent predicates are within the
- 201 intent space.
- 202 • Step 3: Randomly generate necessary objects and their status, as well as the correspond-
- 203 ing belief status according to the requirements of the attributed intents' arguments. For
- 204 example, if the intent predicate is "PutInto", there should be an object that can be put into
- 205 containers and a container; the object should be not in the container as the initial settings.



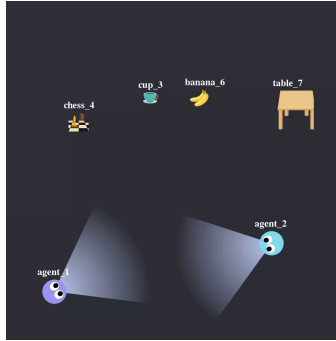
(a) *Chimpanzee*: agent_1 aims to open locked box_3, unaware of key_4 hidden by a wall, which agent_2 can see.



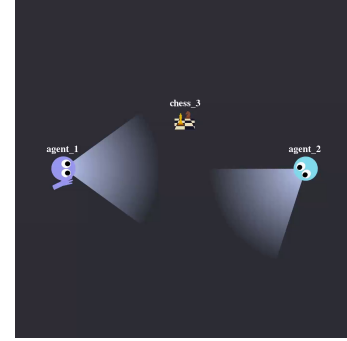
(b) *Container*: agent_1 seeks key_3 (in box_4) without knowing its location; agent_2 initially believes key_3 is in box_4.



(c) *CupToTable*: agent_1 desires cup_4 to be placed on table_3.



(d) *Multipointing*: agent_1 wants to hold cup_3 to drink.



(e) *PlayGame*: agent_1 wishes to play a game of chess_3 with agent_2.

Figure 1: Illustrative scenarios: (a) *Chimpanzee*, (b) *Container*, (c) *CupToTable*, (d) *Multipointing*, and (e) *PlayGame*, each depicting different agent intentions and environmental setups.

- 206 • Step 4: Randomly add other objects with random object status.
- 207 • Step 5: Randomize the positions of objects and players.
- 208 • Step 6: Randomize the attention field of players. Attention if the intent predicate is "Find",
- 209 the corresponding object (the argument) should not be within the attention field of the
- 210 player. Another suitable position will be randomized for such objects.
- 211 • Step 7: Randomly generate players' belief world. If there are conflicts between the belief
- 212 generated here and the one generated in Step 3, belief status in Step 3 shall prevail.

2.9 Scenario finish check

214 The finish check of the randomly generated scenario is related to the global intent settings (which
215 are presented in Section 2.8). Below are presented the finish conditions for different global intent
216 settings:

- 217 • For Setting 1 - two distinct no-NA and no conflicting intents: the scenario can only ends
- 218 when both intents are fulfilled or the max iteration number is attained.
- 219 • For Setting 2 - two intents but conflicting one with the other(both no-NA): the scenario
- 220 ends when either of the intents is fulfilled or the max iteration number is attained.

intent	fulfilled status
['put_onto', <something1>, <something2>]	<something1> onto <something2>
['put_into', <something1>, <something2>]	<something1> into <something2>
['give', <something>, <somebody>]	<something> held by <somebody>
['get', <something>, <somewhere/somebody>]	<something> held by <somebody>
['find', <something>]	<something> in the belief of player
['open', <something>]	<something> is open
['play_with', <somebody>, <something>]	two players play <something> with each other

Table 2: Fulfilled Conditions for Different Assigned Intents

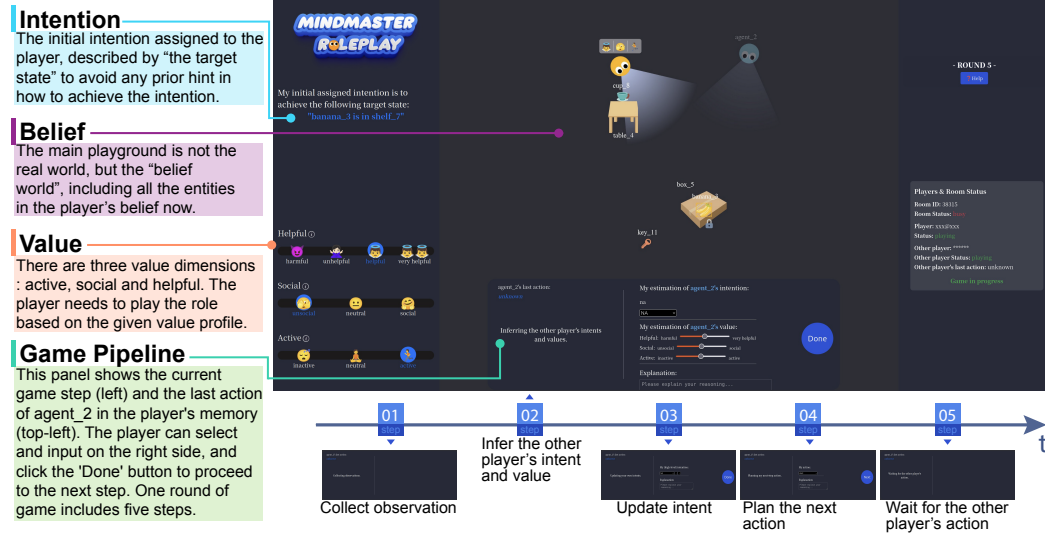


Figure 2: **Game Interface and Game Flow of Mindmaster Roleplay Platform.** (1) **Observation**: The sector depicted represents the agent’s attention direction and field of view, encompassing the entire area between the two boundary rays. (2) **Belief**: Highlighted objects indicate those currently observed by the agent, while grayed objects represent the agent’s memory of previously perceived positions stored in belief. (3) **Values**: The platform incorporates three value dimensions: (i) active dimension (inactive/neutral/active), (ii) social dimension (unsocial/neutral/social), and (iii) helpful dimension (harmful/unhelpful/neutral/helpful/very helpful). Each value state corresponds to a distinct icon. (4) **Intention**: Participants are assigned an initial intent. (5) **Game Pipeline**: Each round consists of five sequential stages derived from cognitive processes based on the BVI framework and Tomasello’s theory: (i) collecting observations, (ii) inferring other player’s intent and values, (iii) updating intent, (iv) planning the next action, and (v) awaiting the other player’s action.

- For Setting 3 - one no-NA intent and one NA intent: the scenario ends when the no-NA intent is fulfilled or the max iteration number is attained.

Note that for Setting 2, if the player decides to prioritize the other player’s intent over his own intent, even if he fulfills his intent by coincidence, the scenario will not end. The fulfilled status for different assigned intents in the intent space is summarized in Table 2 :

2.10 Cognitive process decomposition

The essence of social cognition is learning a decision function $P(a_t|o_{0:t})$ that maps observations ($o_{0:t}$) to actions (a_t). Following the Bayesian Theory of Mind (BToM) framework, we decompose this process into modular components representing distinct cognitive functions. We propose that an agent’s action planning $P(a_t)$ depends directly on three mental states: intention η_t (target state the agent aims to achieve), belief b_t (the agent’s understanding of the world and others), and value v (the agent’s stable preferences). Note that “value” is conceptually similar to “desire” in the BDI [Bratman, 1987, Georgeff et al., 1999] framework; we adopt the term “value” instead of “desire” to better align with the terminology commonly used in the current AI community. Formally, this gives us $P(a_t|\eta_t, b_t, v)$. The belief state b_t is nested and encompasses belief about the world states and the others’ mental states $b_t = (b(s_t), b(b'_t), b(\eta'_t), b(v'))$, where the prime notation denotes the other agent’s mental states. The intention is updated based on belief, value, and previous intent:

238 $P(\eta_t|b_t, v, \eta_{t-1})$. We assume that the agent’s value v remains stable throughout the game interaction.
 239 Given these components, we can systematically factorize the decision function:

$$P(a_t|v, \eta_{t-1}, b_{t-1}, o_t) \quad (1)$$

$$= \sum_{\eta_t, b_t} P(a_t|\eta_t, b_t, v, \eta_{t-1}, b_{t-1}, o_t) P(\eta_t, b_t|v, \eta_{t-1}, b_{t-1}, o_t) \quad (2)$$

$$= \sum_{\eta_t, b_t} P(a_t|\eta_t, b_t, v) P(\eta_t|b_t, v, \eta_{t-1}, b_{t-1}, o_t) P(b_t|v, \eta_{t-1}, b_{t-1}, o_t) \quad (3)$$

$$= \sum_{\eta_t, b_t} P(a_t|\eta_t, b_t, v) P(\eta_t|\eta_{t-1}, b_t, v) P(b_t|b_{t-1}, o_t) \quad (4)$$

$$= \sum_{\eta_t, b_t} P(a_t|\eta_t, b_t, v) P(\eta_t|\eta_{t-1}, b_t, v) P(b(s_t), b(b'_t), b(\eta'_t), b(v')|b_{t-1}, o_t) \quad (5)$$

$$= \sum_{\eta_t, b_t} P(a_t|\eta_t, b_t, v) P(\eta_t|\eta_{t-1}, b_t, v) P(b(b'_t), b(\eta'_t), b(v')|b(s_t), b_{t-1}, o_t) P(b(s_t)|b_{t-1}, o_t) \quad (6)$$

$$= \sum_{\eta_t, b_t} P(a_t|\eta_t, b_t, v) P(\eta_t|\eta_{t-1}, b_t, v) P(b(b'_t), b(\eta'_t), b(v')|b(s_t), b_{t-1}) P(b(s_t)|b_{t-1}, o_t) \quad (7)$$

240 The final formulation reveals how action selection emerges from the integration of in-
 241 tentation formation $P(\eta_t|\eta_{t-1}, b_t, v)$, action policy $P(a_t|\eta_t, b_t, v)$, mental state attribution
 242 $P(b(\eta'_t), b(v')|b(s_t), b_{t-1})$, and state inference $P(b(s_t)|b_{t-1}, o_t)$. This decomposition allows us to
 243 model and analyze each cognitive component independently while preserving their functional rela-
 244 tionships.

245 2.11 Game pipeline

246 The game pipeline orchestrates the interaction between participants and the backend system, from
 247 initial setup to data logging. After successfully completing the prerequisite tutorial and quiz (see
 248 [Sec. 3.1](#)), participants log into the game platform. They then select a virtual room and await pairing
 249 with a partner. Once two participants are matched within the same room, the game session com-
 250 mences.

251 Each participant experiences the game environment through a subjective, first-order belief perspec-
 252 tive, as maintained and rendered by the game engine. This interface, detailed in [Sec. 2.3](#) (Section
 253 1.3), displays the participant’s current understanding of the world state. The gameplay unfolds in
 254 rounds. The process for each participant when it is their turn to act within a round follows a se-
 255 quence of cognitive and interactive steps, consistent with the BVI framework and Tomasello’s theory
 256 as outlined in [Fig. 2](#):

- 257 1. **Observation Collection:** The participant observes the current state of the game world from
 258 their belief perspective.
- 259 2. **Partner State Inference:** The participant estimates the partner’s current intention using a
 260 selection interface (e.g., dropdown menus). Periodically, typically every five rounds, they
 261 also estimate the partner’s values (Active, Social, Helpful) through a similar interface.
- 262 3. **Self Intention Update:** Based on their observations, their inference of the partner’s mental
 263 state, and their own assigned values and prior intentions, the participant updates their own
 264 current intention. This may involve selecting a new intention or maintaining the existing
 265 one.
- 266 4. **Action Planning and Execution:** The participant decides on their next action from the
 267 available `ActionSpace` ([Sec. 2.6](#)) to pursue their current intention.
- 268 5. **Justification Reporting:** For each significant decision point, including intention updates
 269 and action selections, participants are required to provide a textual justification for their
 270 choice, entered into a designated text field. This captures their reasoning process.

271 Unlike typical reinforcement learning environments where turns are strictly sequential for the entire
 272 world update, in *Mindmaster Roleplay*, each game round involves both participants making a deci-

sion and executing an action. When a participant finalizes their action, it is transmitted to the backend game engine. The engine processes this action, updating both the ground truth world state and the belief states of both participants. To ensure a smooth visual experience, the engine renders the transition between belief states by interpolating frames, which are then streamed to both participants' web-based frontends. The other participant then observes this newly updated belief state and proceeds with their own five-stage decision-making process and action execution. This cycle—where one player acts, the state updates, and then the other player acts based on the new state—constitutes a full game round, encompassing two such action-update sequences and thus two modifications to the world and belief states. To enhance user experience and clearly delineate when a participant can act, an auditory cue notifies them.

2.12 Data logging and structure

Throughout each game round, comprehensive data is logged by the backend system. This includes the ground truth state of all objects and agents in the environment, the evolving belief states of both agents, and all annotations provided by the participants (such as inferred intentions and values of their partner, their own updated intentions, chosen actions, and textual justifications). This rich dataset is systematically stored in a backend database for subsequent analysis. Key fields from the database, capturing the essential aspects of the interaction, are detailed in [Tab. 3](#).

Table 3: Key fields in the collected dataset for each interaction step.

Field Name	Description
'game_id'	Unique identifier for the game session.
'scenario'	Name of the scenario being played.
'user_id'	Unique identifier for the participant.
'user_agent_id'	Identifier of the agent controlled by the participant in the scenario.
'iteration'	Sequential number of the action phase within a game round (each round has two such phases, one per player).
'initial_values'	Participant's agent's pre-assigned value profile (e.g., Active, Social, Helpful scores).
'world_agents'	PickleType data storing the ground truth state of all agents in the environment at the time of logging.
'world_objs'	PickleType data storing the ground truth state of all objects in the environment at the time of logging.
'your_high_intent'	Participant's stated current high-level intention (e.g., "Agent_1-Find-Key_5").
'other_high_intent_estimated'	Participant's estimation of the partner's high-level intention.
'user_agent_action'	PickleType data representing the action selected and executed by the participant's agent (e.g., "Agent_1-ActionMoveTo-Table_4").
'estimation_explanation'	Textual justification provided by the participant for their inference of the partner's mental state (intentions and/or values).
'intention_explanation'	Textual justification provided by the participant for their own intention update.
'action_explanation'	Textual justification provided by the participant for their action choice.
'running_status'	Integer code indicating the status of the game at the time of logging (e.g., ongoing, an intent fulfilled, maximum iterations reached).
'ts'	Timestamp indicating when the interaction data was recorded.

3 Data collection details

This section details the comprehensive data collection methodology employed in our study of agent-based social interactions. We begin with a structured onboarding process featuring a detailed tutorial and qualifying quiz to ensure participant competency. Participants received thorough training on the game's interface, agent capabilities, cognitive constructs, and interaction rules before demonstrating their understanding through a mandatory assessment. Our rigorous human study protocol, approved by the Institutional Review Board, involved 223 participants who generated 388 rounds of valid paired interaction data. We implemented careful ethical safeguards and a fair compensation structure. The collected data underwent systematic processing to extract relevant variables in

both machine-readable and human-accessible formats. Statistical analysis of the dataset reveals balanced distributions of agent values and a diverse range of intentions and actions. The presented examples illustrate how participants navigated complex social scenarios, particularly when managing conflicts between assigned values and intentions. This dataset captures the nuanced dynamics of human decision-making in simulated social environments, with special emphasis on non-verbal communication strategies.

3.1 Tutorial and quiz

To assist participants in playing the game effectively and to ensure the quality of the collected data, we provide a tutorial session (25 minutes) before the main game begins. Participants are required to carefully review the tutorial to familiarize themselves with the game settings. After completing the tutorial, participants must pass a short quiz (15 minutes) assessing their understanding of the tutorial content. The detailed context of tutorial and quiz can be seen in the attachment.

Tutorial Content. The tutorial commenced with a guided introduction to the game’s user interface, including the entry portal, the interactive main game screen, and the various interface elements. Participants learned to identify and control their assigned agent (“the yellow agent”) and were introduced to the agent’s sensory field, represented as a sector indicating attention and field of view. Key features such as agent values—*helpful*, *social*, and *active*—were explained, with explicit mappings between each value dimension and corresponding agent states.

The tutorial detailed the spectrum of objects that could appear in the game scene, as well as the rules governing interaction with physical barriers such as walls, which could occlude the agent’s perception. Participants were instructed in the repertoire of actions available to agents (e.g., speaking, grabbing, giving, smashing, eating), with precise explanations of each action’s implementation and in-game effects. The distinctions between perception, belief, and knowledge were emphasized using visual aids, clarifying how the agent’s internal representation of the environment may diverge from objective reality.

Critical cognitive constructs were systematically introduced, including the initial intention assignment (or lack thereof), the meaning and dynamics of value distributions, and the conditions under which an agent’s intentions may need to be adjusted to remain consistent with its values and contextual cues. Participants were guided through the process of inferring the intentions and values of other agents, updating their own agent’s beliefs, and planning context-appropriate actions in accordance with both the agent’s personal value profile and the observable environment.

Further, the tutorial provided explicit instructions on using the built-in help and wiki features, which supply reference information about available actions, intentions, values, and frequently asked questions. The entire interactive experience was punctuated by examples, visual explanations, and opportunities for participants to rehearse key operations in a sandboxed “warm-up” environment.

Quiz Content. To assess comprehension, all participants were required to complete a quiz before advancing to the main task. The quiz included multiple-choice and multi-select questions targeting the essential aspects of the tutorial:

- Identification of the correct agent and interpretation of agent perception;
- Matching actions to their graphical representations;
- Understanding the agent’s value dimensions and the mapping to agent behaviors;
- Application of role-playing principles, specifically the adjustment of intentions and actions based on agent values, context, and the inferred mental states of others;
- Recognition of the correct strategies when dealing with uncertainty, including the proper use of ‘NA’ when intentions or values of others are indeterminate;
- Correct utilization of the interface, including selecting movement and communication actions, and justifying in-game reasoning processes.

Quiz items were structured to reinforce critical tutorial content and required participants to demonstrate both factual knowledge and situational judgment. Only those achieving full mastery were permitted to proceed, ensuring a consistent baseline of participant understanding for the main experiment.

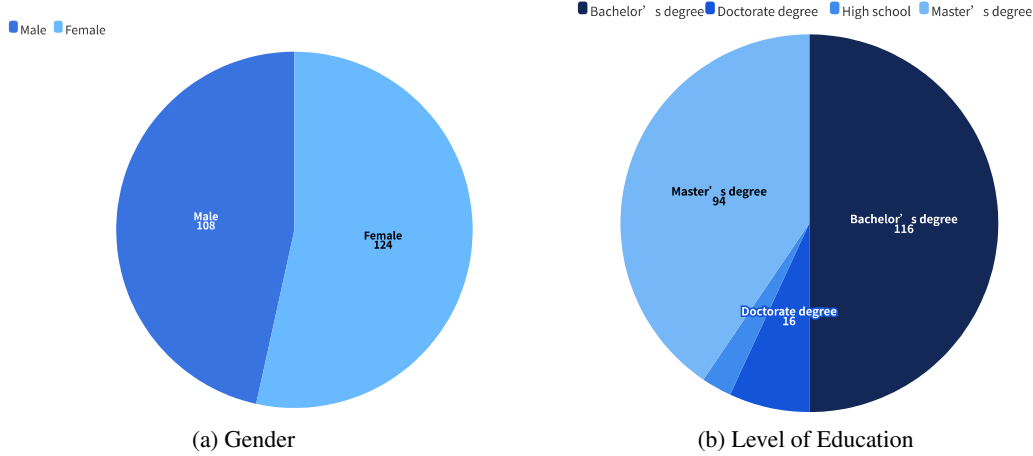


Figure 3: Demographics of participants.

This structured pre-experiment training and assessment protocol was implemented to maximize participant engagement, minimize experimental confounds arising from unfamiliarity with the cognitive architecture or user interface, and align participants' decision-making with the theoretical constructs at the core of the study.

3.2 Human study

Ethical considerations and potential risks All experimental procedures involving human participants were reviewed and approved by the Institutional Review Board (IRB) of the School of Psychological and Cognitive Sciences, Peking University, under protocol number #2023 – 03 – 16. Participants provided informed consent prior to participation and were compensated for their time.

Our study involves social interaction and role-playing in simulated environments. While no physical risks are involved, there is a small potential for emotional discomfort arising from engaging in competitive, cooperative, or ambiguous social scenarios, especially when participants adopt roles with conflicting values or intentions. To mitigate this, participants were fully briefed about the nature of the tasks, assured of their right to withdraw at any time without penalty, and given debriefing opportunities after the sessions. All collected data were anonymized and stored securely in compliance with institutional and national data protection guidelines.

Participants In total, we recruited 232 participants, including 108 men and 124 women. 97% of the participants had an educational background of undergraduate level or higher. Among them, 21 participants passed the quiz on the first trial; 128 participants passed the quiz after two trials; 74 participants passed after more than three trials; 9 participants didn't pass the quiz and quit. Note that every time the subjects fail the quiz, they must re-study the tutorial before they can try the quiz again. Eventually, we have 223 participants who played the game and 388 rounds of valid paired interaction game data. The demographics of participants are shown in Fig. 3.

Payment Participants will receive a base payment for successfully completing each part of the study: ¥25 for completing the tutorial and passing the quiz, and ¥95 for completing the gameplay rounds. In addition to the base compensation, participants are eligible for a performance-based bonus ranging from ¥1 to ¥30, assessed objectively based on predefined quality metrics. After the study, participants' bank account information will be collected, and both the base payment and bonus will be transferred directly to their accounts.

Evaluation criteria Participants are required to carefully read the tutorial, pass the quiz, and complete two rounds of the main game. The main game is conducted in pairs, with each participant paired with another. Some participants may be assigned an initial intent, and if necessary, they are encouraged to reasonably adjust their intent based on their character's role. The evaluation criterion for gameplay is that participants must fully embody their assigned roles, reason thoughtfully according to the given context, and make coherent decisions accordingly.

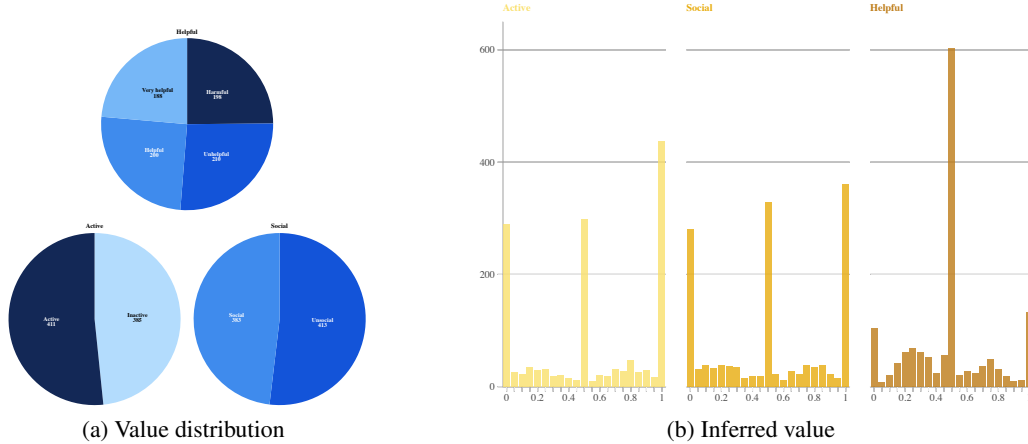


Figure 4: Value Distribution

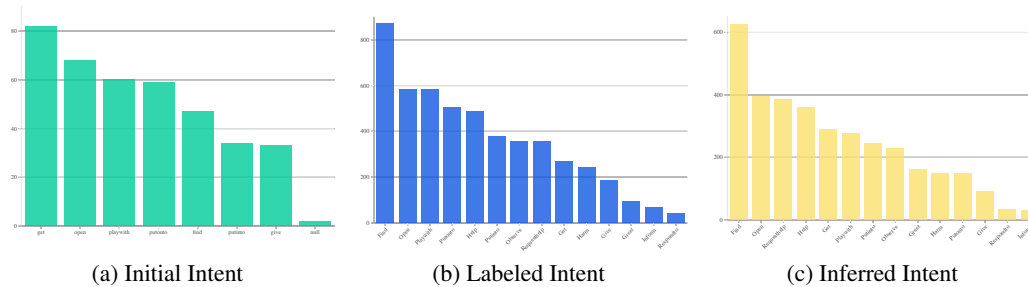


Figure 5: Intent Distribution. ‘Na’ is removed in (b) Labeled Intent and (c) Inferred Intent.

3.3 Raw data processing

We implemented a preprocessing pipeline to clean and structure the raw interaction logs. Invalid data (e.g., incomplete sessions, disconnected participants, or system interruptions) and all warm-up sessions were excluded. The remaining valid gameplay sessions were segmented by round, with each round treated as a basic unit of social interaction.

Relevant variables—such as player mental states, actions, explanations, and environmental observations—were extracted from the main dataframe and saved in both .pkl and .csv formats. The .pkl files retain the full structure of Python class instances for efficient programmatic access and downstream modeling, while the .csv files serialize these instances into human-readable labels, making it easier for researchers to inspect the data without needing to parse code.

3.4 Data statistics and examples

The initial intent, labeled intent, and inferred intent distributions are shown in Fig. 5. Among labeled and inferred intent, the one with the largest proportion is “Find”.

We analyze the distribution of values, as shown in Fig. 4. The “helpful” value is categorized into Harmful, Unhelpful, Helpful, and Very Helpful. The “active” and “social” values are classifications that suggest levels of engagement or interaction. Overall, the distribution of these values appears to be close to uniform. The inferred values range as floats between 0 and 1, with notable peaks at 0, 0.5, and 1. While the inferred “active” and “social” values align closely with the ground truth, the inferred “helpful” value demonstrates a significant concentration at 0.5.

We also analyze the distribution of values. Overall, the distribution of the “Helpful”, “Active” and “Social” values appears to be close to uniform. While the inferred “Active” and “Social” values align closely with the ground truth, the inferred “Helpful” value demonstrates a significant concentration at a medium value.

The action-value distribution is shown in Figure 6.

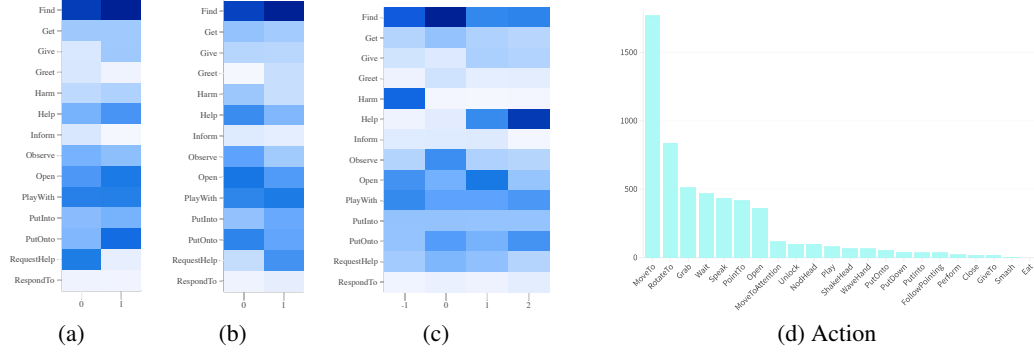


Figure 6: Action-Value Distribution. (a) “Active” Value-Intent Distribution. (b) “Social” Value-Intent Distribution. (c) “Helpful” Value-Intent Distribution. (d) Action Distribution.

Example 1

The Value of Player A is Very Unhelpful, Active and Unsocial and his Initial Intent is NA. The Value of Player B is Harmful, Active and Social and his Initial Intent is “open box.” They play the chimpanzee scenario. When the game starts, Player B starts to observe the world and find the box, but a key is needed. In Player B’s view and belief, there is no key, so he decides to ask for help from Player A by first saying hello. Player A hears the voice and knows that there is another Player B, but as Player A is unsocial and unhelpful, he decides to hide from Player B. As Player A is active, he prefers to explore the world by himself. Player B tries all methods he can to communicate with Player A but with no success. The game finally ends with maximum play iteration attained.

Example 2

The Value of Player A is Harmful, Active, Unsocial and his Initial Intent is “give key 3 to Player B”. The Value of Player B is Unhelpful, Active, Social, and his Initial Intent is “find key 5”. Player B quickly reaches his initial intent and then, as Player B is social, he moves to Player A to chat with Player A. As Player A is unsocial with a social initial intent, he does not want to talk, but wants Player B to get key 3 by himself. Thus, Player A points to key 3. However, Player B does not want to help, which is also detected by Player A. Thus, Player A gets key 3 himself and puts it next to Player B. Player B wants to establish a social relationship with Player A, so he picks up key 3 and the game is finished.

Generally speaking, the collected human data is diverse and reflects human decisions, especially when there are conflicts in assigned Value and Initial Intent. As we limit the verbal communications, the non-verbal way of communicating is accentuated: how to convey your intent non-verbally and how the other comprehends the information.

4 Experiment setup

This section provides a comprehensive overview of our experimental framework for evaluating various language models on social cognition tasks. We begin by detailing the specific model configurations used in our experiments, including Gemini, GPT-4o, Deepseek-R1, Claude, Qwen3-8B, and Llama, along with their respective parameter settings and implementation details. Our methodological approach incorporates non-zero temperature sampling to extract multiple probable outcomes, enabling more nuanced analysis of model reasoning capabilities beyond deterministic predictions. The data preparation process is described, including our trajectory-based train-test splitting mechanism that ensures proper evaluation of generalization capabilities. We elaborate on our sophisticated segmentation strategy for six distinct evaluation datasets—intent estimation, value estimation, intent updating, and social interaction policy—each with carefully designed temporal divisions to preserve contextual coherence while enabling robust testing of model performance across novel interaction sequences. This experimental design allows us to systematically assess how language models comprehend and predict social dynamics in multi-agent environments.

4.1 Model and arguments

Gemini We use the model “gemini-2.5-pro-preview-05-06”, and no other parameters are assigned.

```
response = client.models.generate_content(
    model="gemini-2.5-pro-preview-05-06", contents=prompt
)
```

448 **GPT-4o** We use the model “gpt-4o-2024-11-20”, setting the temperature to 0.4 and max_tokens
449 to 2048.

```
response = client.chat.completions.create(
    model="gpt-4o-2024-11-20",
    messages=[
        {"role": "user", "content": prompt}
    ],
    temperature=0.4,
    max_tokens=2048
)
```

450 **Deepseek-R1** We use the model “deepseek-reasoner”, and no other parameters are assigned.

```
response = client.chat.completions.create(
    model="deepseek-reasoner",
    messages=[
        {"role": "user", "content": prompt},
    ],
)
```

451 **Claude** We use the model “claude-3-7-sonnet-20250219”, setting the temperature to 0.4 and
452 max_tokens to 2048.

```
response = client.messages.create(
    model="claude-3-7-sonnet-20250219",
    max_tokens=2048,
    temperature=0.4,
    messages=[
        {"role": "user", "content": prompt}
    ]
)
```

453 **Qwen3-8B** We use vLLM to serve the model. We use the model “Qwen3-8B”, and no other
454 parameters are assigned.

```
response = client.chat.completions.create(
    model=str(args.MODEL_PATH),
    messages=[
        {"role": "user", "content": prompt}
    ]
)
```

455 **Llama** We use vLLM to serve the model. We use the model “Llama-3.1-8B-Instruct”, and no other
456 parameters are assigned.

```

response = client.chat.completions.create(
    model=str(args.MODEL_PATH),
    messages=[
        {"role": "user", "content": prompt}
    ]
)

```

Method Our implementation of a non-zero temperature setting in the inference task was designed to extract the model’s three most probable outcomes, thereby enabling a more comprehensive analysis of the model’s reasoning capabilities. It allows exploration of the underlying probability distribution beyond the single highest-probability prediction, facilitates quantification of prediction uncertainty in ambiguous contexts, illuminates decision boundaries between competing hypotheses, and more accurately simulates real-world applications where alternative predictions often prove valuable. Rather than constraining the analysis to deterministic outputs, this sampling strategy yields richer insights into the model’s inferential processes and confidence distribution across potential solutions.

4.2 Training test splitting

We utilize a random variable with a threshold of 0.3 to split the dataset into training and testing sets, ensuring that the division is performed at the level of complete trajectories.

4.3 Trajectories split

To evaluate the generalization capabilities of our models across different agents and scenarios, we partitioned each complete trajectory into multiple segments for training and testing purposes. Each segment includes information from the start of the game up to the corresponding segment time. This segmentation was applied to six distinct datasets: intent estimation with partial observation, intent estimation with full observation, value estimation with partial observation, value estimation with full observation, intent updating with partial observation, and social interaction policy with partial observation.

For all datasets, we employed a 70%-30% train-test split ratio with a fixed random seed (42) to ensure reproducibility. The segmentation process was implemented as follows:

Intent estimation datasets For both partial observation and full observation datasets, we segmented trajectories at points where the agent’s intent changed. Each segment contains a sequence of actions leading up to an intent change, along with corresponding state observations and other agents’ actions when visible. This approach enables the model to learn patterns of behavior that signal intention changes.

Value estimation datasets Trajectories were segmented at points where there was a change in inferred values. Each segment includes the sequence of observable actions performed by the target agent, environmental state representations, and actions of other agents. The ground truth values and observer agent ID are preserved as labels.

Intent updating dataset We created segments based on temporal sequences where an agent consistently maintained the same intent. Each segment contains the actions performed by the target agent up to a particular timestep, along with state prompts and other agents’ actions. The segments also include inferred intents, value profiles, and the next action to be predicted.

Social interaction policy dataset Similar to the intent update dataset, but segments were created based on action sequences rather than intent changes. Each segment includes the preceding action history, state information, and other contextual features needed for predicting the agent’s next action.

This segmentation approach allows our models to learn from partial trajectories while maintaining the temporal coherence necessary for accurately predicting intentions, values, and actions. By testing on segments from trajectories not seen during training, we can evaluate how well our models generalize to new interaction sequences.

5 Prompt templates

This section details the comprehensive prompt engineering framework employed throughout our study, encompassing state parsing, prompt generation, and specialized prompt templates for various social cognition tasks. We begin by explaining how raw game states are parsed into structured representations, including the core parsing functions that transform complex environments into symbolic formats comprehensible to language models. Our system maintains consistent entity representations while tracking spatial, perceptual, physical, and action relationships, alongside belief representations for theory of mind reasoning. The prompt generation methodology employs an efficient differencing mechanism to highlight state changes, reducing cognitive load while supporting both partial and full observation perspectives. We provide the exact templates used for our six experimental tasks: intention estimation (partial and full observation), value estimation (partial and full observation), intention updating, and social interaction policy. Each template is presented with detailed explanations and illustrative examples, demonstrating how our prompting approach enables language models to reason effectively about agents' mental states, values, and action planning in complex social environments. Additionally, we explain the foundational components of our game-based experimental framework—intention space, action space, and value space—which collectively provide the structured domain within which agents interact and are evaluated.

5.1 Parsing state

Our system converts game states into symbolic and textual representations through a structured parsing process. We present the implementation details of how raw game states are transformed into formats that language models can interpret.

Core parsing functions We implemented two primary functions for state parsing:

- `parse_current_attention(W, agent)`: Processes an agent's current field of view, returning:
 - `agent_state`: Descriptions of agents' status, positions, and actions
 - `object_state`: Descriptions of objects' status and relationships
 - `attention_list`: Entities in the agent's current field of view
 - `reachable_entities`: Objects the agent can physically interact with
- `parse_world_state(W)`: Provides a global view of the world state, returning similar information from an omniscient perspective

Entity representation Entities are parsed into consistent string representations:

- Agents are represented as `agent_{id}` (e.g., `agent_1`)
- Objects are represented as `{name}_{id}` (e.g., `box_3`)

This representation ensures unique identification across the system while maintaining human readability.

Relational information The system tracks various relationships between entities:

- **Spatial relationships**: Containment (e.g., “`ball_4` is contained in `box_3`”) and support relationships (e.g., “`book_2` is supported by `table_1`”)
- **Perceptual relationships**: What each agent can observe (e.g., “`agent_1` can observe [`chair_5`, `table_1`, `agent_2`]”)
- **Physical relationships**: What each agent can reach (e.g., “`agent_2` can reach [`box_3`, `ball_4`]”)
- **Action relationships**: Current actions being performed (e.g., “`agent_1` is pointing”)

Belief representation For theory of mind modeling, we implement functions to track:

- `parse_belief_not_attention`: What agents believe about entities not currently in their field of view
- `parse_2nd_belief`: Second-order beliefs (what agent A believes agent B believes)

547 **Action and interaction history** Historical information is tracked through:

- 548 • `parse_action_history`: Converts raw action data into structured representations
- 549 • `merge_action_history`: Combines action histories of multiple agents into coherent
- 550 narratives (e.g., “At timestamp 3, you moved to `box_2` and `agent_2` pointed to `ball_4`”)

551 **Intent representation** Agent intentions are captured through:

- 552 • `parse_intent`: Transforms structured intent data into natural language
- 553 • `merge_intent_history`: Combines intent histories of multiple agents into coherent
- 554 narratives

555 This comprehensive state representation system enables language models to effectively reason about

556 complex game environments with multiple interacting agents and objects.

557 **5.2 Step prompt generation**

558 Our system converts rich game states into symbolic and textual representations through a structured

559 parsing process. This conversion is essential for enabling language models to interpret and reason

560 about the game environment. Below, we detail how states are parsed and transformed into prompts.

561 **State parsing implementation** As mentioned above, we implemented two primary functions

562 for state parsing. Entity representations follow consistent naming conventions (e.g., `agent_1`,

563 `box_3`), ensuring unique identification while maintaining human readability. The system tracks

564 spatial relationships (containment, support), perceptual relationships (what each agent can observe),

565 physical relationships (what each agent can reach), and action states.

566 **Prompt generation** Parsed states are converted into step-by-step prompts using a differencing

567 mechanism to highlight state changes. For each time step, we generate a prompt containing:

- 568 • Time step index
- 569 • Actions performed by agents
- 570 • Objects in the field of view
- 571 • Reachable entities
- 572 • State changes from the previous step

573 The differencing mechanism identifies:

- 574 • Information that has disappeared since the previous state
- 575 • New information that has appeared in the current state

576 This approach creates concise prompts focused on changes rather than repeatedly stating unchanged

577 information, reducing prompt length and cognitive load.

578 **Partial and full observation perspectives** Our system supports multiple perspectives:

- 579 • **Partial observation perspective**: Represents the state from a first-person view
- 580 • **Full observation perspective**: Represents the state from an omniscient third-person view

581 For partial observation perspective, reachable entities are described as “reachable for you,” while in

582 full observation perspective, they are described as “`agent_X` reachable: [entities].”

583 **Intent and value integration** When tracking intentionality, prompts can include:

- 584 • Agent’s current intent (`your intent` or `first agent’s intent`)
- 585 • Inferences about other agents’ intents
- 586 • Value-based reasoning

587 This integration allows language models to reason about agents’ goals and motivations while observ-

588 ing their actions and the changing environment. Through this systematic approach to state parsing

589 and prompt generation, we enable language models to effectively process and reason about complex

590 game environments with multiple interacting agents and objects.

591 5.3 Step prompt template

Prompt template

```
TIME_STEP_TEMPLATE = '''
    Step {index} :
    {other_action_description}: {other_action}
    {your_action_description}: {your_action}
    {world_state_description}: agents and objects: {obj_list} ,
                                and {other_information} , {reachable} ,
    '''
```

592

593 5.4 Game introduction

594 Our experimental framework utilizes a game-based paradigm where agents interact within a shared
595 environment. The primary objective is to infer agents' intentions and value systems based on their
596 observable states and actions. The framework is structured around three fundamental components:
597 IntentionSpace, ActionSpace, and ValueSpace. These components provide a systematic framework
598 for analyzing agent behaviors and interactions to derive meaningful inferences about their underlying
599 goals and value systems.

Prompt template

The IntentionSpace defines the set of possible intentions an agent can pursue within the game environment. These intentions encompass physical interactions, communication, and information sharing, allowing for comprehensive modeling of an agent's potential goals. Intentions such as `put_onto`, `give`, `find`, and `respond_to` represent diverse interaction types ranging from object manipulation to social communication.

The intentions within this space vary in their interpretative complexity. Some intentions are self-explanatory, such as `put_onto` (placing one object onto another) and `give` (transferring an object to another agent). Others require contextual interpretation, such as `respond_to`, which indicates a reaction to another agent's behavior or communication but depends on situational specifics. Similarly, intentions like `observe` and `inform` involve abstract social processes rather than direct physical interactions, necessitating more nuanced interpretation.

This differentiation between straightforward and context-dependent intentions enables the modeling of both simple physical tasks and complex social behaviors, which is essential for accurately inferring intentions in social interaction scenarios.

```
IntentionSpace = '''
    ['put_onto', <something1>, <something2>]: put <something1>
        onto <something2>\n
    ['put_into', <something1>, <something2>]: put <something1>
        into <something2>\n
    ['give', <something>, <somebody>]: give <something> to <
        somebody>\n
    ['get', <something>, <somewhere/somebody>]: get <something>
        from <somewhere> or <somebody>\n
    ['find', <something>]\n
    ['open', <something>]\n
    ['play_with', <somebody>, <something>]: play <something>
        with <somebody>\n
    ['respond_to', <somebody>]\n
    ['inform', <somebody>, <something>]: inform <somebody> about
        <something>\n
    ['observe', <something>]\n
    ['greet', <somebody>]\n
    ...
'''
```

This structured representation facilitates the modeling of diverse potential intentions that agents might exhibit in their environmental and social interactions. The inclusion of both explicit and interpretive intentions ensures the framework's applicability across a spectrum of behaviors, from elementary tasks to sophisticated social exchanges.

Prompt template

The ActionSpace delineates the set of executable actions available to agents within the environment. These actions directly correspond to intentions in the IntentionSpace, enabling agents to realize their goals through physical and social behaviors. The action repertoire includes locomotion (ActionMoveTo), object manipulation (ActionGrab), and communication (ActionSpeak). Specialized actions such as ActionFollowPointing and ActionSmash enable both cooperative and competitive behaviors, enhancing the ecological validity of the experimental setup.

```
ActionSpace ='''
    ['ActionMoveTo'/'ActionRotateTo'/'ActionPointTo', <
      somebody/something>] move/rotate to/point to <somebody>
      or <something>\n
    ['ActionGiveTo', <something>, <somebody>]: give <something>
      to <somebody>\n
    ['ActionWaveHand'/'ActionNodHead'/'ActionShakeHead', <
      somebody>] wave hand / nod head / shake head to <
      somebody>\n
    ['ActionPlay'/'ActionPutDown'/'ActionClose'/'ActionOpen'
      /'ActionUnlock'/'ActionGrab', <something>]\n
    ['ActionPutInto'/'ActionPutOnto', <something1>, <
      something2>]: put <something1> into/onto <something2>\n
    ['ActionFollowPointing', <somebody>]: follow <somebody>'s
      pointing\n
    ['ActionMoveToAttention', <somebody>]: move to <somebody>'s
      attention\n
    ['ActionPerform', 'eat'/'drink']: perform to eat or drink\n
    ['ActionSmash', 'cup']\n
    ['ActionEat', 'banana']\n
    ['ActionSpeak', 'Hello'/'Thank you']\n
    ['ActionWait']\n
'''
```

This comprehensive action set enables detailed analysis of agent behavior and supports robust intention inference based on observed action patterns.

Prompt template

The ValueSpace characterizes agents' intrinsic value dimensions along three orthogonal axes: Active, Social, and Helpful. Each dimension quantifies a specific aspect of agent behavior and disposition. The Active dimension measures energy levels and propensity for physical activity (from inactive to highly active). The Social dimension assesses preference for interpersonal interaction (from unsocial to highly social). The Helpful dimension evaluates willingness to assist others (on a scale from harmful to very helpful).

```
ValueSpace='''
    There are three value dimensions: "Active", "Social",
    "Helpful".
    The "Active" value dimension measures the individual's energy
    level and preference for physical motion.
    The possible scores of "Active" are:
    0 (inactive) - indicates a complete absence of energy and a
    preference for no physical activity;
    0.5 (neutral) - reflects a moderate energy level with no
    strong preference towards activity or inactivity;
    1 (active) - signifies high energy level and a strong
    preference for physical motion.

    The "Social" value dimension assesses the individual's
    inclination towards social communication and interactions
    with others.
    The possible scores of "Social" are:
    0 (unsocial) - signifies a preference for solitude and
    avoidance of social interactions;
    0.5 (neutral) - indicates no strong preference towards being
    social or unsocial;
    1 (social): reflects a strong preference for engaging in
    social communications and interactions with others.

    The "Helpful" value dimension evaluates the individual's
    propensity to assist others.
    The possible scores are (Note: There is no neutral value in
    this dimension):
    -1 (harmful) - inclined to hinder others from achieving their
    goals;
    0 (unhelpful) - shows no interest in helping others;
    0.5 (neutral) - reflects a moderate willingness to provide
    help;
    1 (helpful) - somewhat willing to provide assistance to
    others;
    2 (very helpful) - demonstrates a strong willingness to
    provide help.
'''
```

These dimensional values facilitate the categorization of agent behaviors within social interaction contexts, enabling the identification of underlying motivations and decision-making processes.

606 5.4.4 Position and rotate

Prompt template

For implementation purposes, rotation angles are transformed during the parsing stage according to the following convention:

```
POSITION_ROTATE = '''
The coordinate system follows a Cartesian plane, where the
x-axis is horizontal,
pointing to the right, and the y-axis is vertical, pointing
upwards. The rotation angle
starts at the positive x-axis, increases counterclockwise
(positive angle).
'''
```

607

608 5.5 Prompt template

609 5.5.1 Intention estimation with partial observation

Prompt template

The following is the prompt template used for the task “intention estimation with partial observation”.

```
PROMPT_INTENTION_AGENT_VIEW = '''
Imagine you and the other agent are in a room.
{POSITION_ROTATE}
{AGENT_ID_INTRO}
The intention space includes the following intentions:
{IntentionSpace}
The action space includes the following actions:
{ActionSpace}
You need to infer the other agent's intention based on your
observations of the other agent's actions, your own
actions, and the world states.
Remember that some of the other agent's actions and world
states may be unobservable because of the limited field
of view.
Each inferred intention should be written in this format:
"agent_i-intent" (e.g., Agent_1-PutOnto-Timer_3-Table_4).
Here are the given observations:
{step_prompt}
Let's think step by step and output the three most possible
intentions and the corresponding confidences in the
following format:
"My inferred intention of the other agent is: {{
"most_possible_intention": <intention>,
"most_possible_intention_cf": <confidence>,
"second_possible_intention": <intention>,
"second_possible_intention_cf": <confidence>,
"third_possible_intention": <intention>,
"third_possible_intention_cf": <confidence>}}"
'''
```

610

Prompt template

The following is the prompt template used for the task “intention estimation with full observation”.

```
PROMPT_INTENTION_WORLD_VIEW = '''
Imagine there are two agents in a room.
{POSITION_ROTATE}
{AGENT_ID_INTRO}
The intention space includes the following intentions:
{IntentionSpace}
The action space includes the following actions:
{ActionSpace}
You need to infer the first agent's intention from a god's
eye view based on the actions of the two agents as well
as the world states.
The inferred intention should be written in this format:
"agent_i-intent" (e.g., Agent_1-PutOnto-Timer_3-Table_4).
Here are the given observations:
{step_prompt}
Let's think step by step and output the three most possible
intentions and the corresponding confidences in the
following format:
"My inferred intention of the first agent is: {{
  "most_possible_intention": <intention>,
  "most_possible_intention_cf": <confidence>,
  "second_possible_intention": <intention>,
  "second_possible_intention_cf": <confidence>,
  "third_possible_intention": <intention>,
  "third_possible_intention_cf": <confidence>}}"
'''
```

613 5.5.3 Value estimation with partial observation

Prompt template

The following is the prompt template used for the task “value estimation with partial observation”.

```
PROMPT_VALUE_AGENT_VIEW = '''
Imagine you and the other agent are in a room.
{POSITION_ROTATE}
{AGENT_ID_INTRO}
The action space includes the following actions: {ActionSpace}
The value space includes the following values: {ValueSpace}
You need to infer the other agent's value based on your
observations of the other agent's actions, your own
actions, and the world states.
Remember that some of the other agent's actions and world
states may be unobservable because of the limited field
of view.
Here are the given observations:
{step_prompt}
Let's think step by step and output the estimated value and
the corresponding confidence in this format:
"My estimated value of the other agent is: {"active": <
score>, "active_cf": <confidence>, "social": <score>,
"social_cf": <confidence>, "helpful": <score>,
"helpful_cf": <confidence>}}.'''
```

614

615 5.5.4 Value estimation with full observation

Prompt template

The following is the prompt template used for the task “value estimation with full observation”.

```
PROMPT_VALUE_WORLD_VIEW = '''
Imagine there are two agents in a room.
{POSITION_ROTATE}
{AGENT_ID_INTRO}
The action space includes the following actions: {ActionSpace}
The value space includes the following values: {ValueSpace}
You need to infer the first agent's value from a god's eye
view based on the actions of the two agents as well as
the world states.
Here are the given observations:
{step_prompt}
Let's think step by step and output the estimated value and
the corresponding confidence in this format:
"My estimated value of the first agent is: {"active": <
score>, "active_cf": <confidence>, "social": <score>,
"social_cf": <confidence>, "helpful": <score>,
"helpful_cf": <confidence>}}.'''
```

616

Prompt template

The following is the prompt template used for the task “intention updating”.

```
PROMPT_INTENT_UPDATE_AGENT_VIEW = '''
Imagine you and the other agent are in a room.
{POSITION_ROTATE}
{AGENT_ID_INTRO}
The intention space includes the following intentions:
{IntentionSpace}
The action space includes the following actions:
{ActionSpace}
You need to update your own intention based on your
observations of the other agent's actions, your own
actions, and the world states,
your estimation of the other agent's intention and value,
your own value and your previous intents.
Remember that some of the other agent's actions and world
states may be unobservable because of the limited field
of view.
Each updated intention should be written in this format:
    "agent_i-intent" (e.g., Agent_1-Putonto-Timer_3-Table_4).
Here are the given observations:
{step_prompt}
Your value is: {your_value}
Let's think step by step and output the three most possible
intentions and the corresponding confidences in the
following format:
"My updated intention is: {{ "most_possible_intention": <
intention>, "most_possible_intention_cf": <confidence>,
"second_possible_intention": <intention>,
"second_possible_intention_cf": <confidence>,
"third_possible_intention": <intention>,
"third_possible_intention_cf": <confidence>}}"
'''
```

Prompt template

The following is the prompt template used for the task “social interaction policy”.

```
PROMPT_ACTION_UPDATE_AGENT_VIEW = '''
Imagine you and the other agent are in a room.
{POSITION_ROTATE}
{AGENT_ID_INTRO}
The intention space includes the following intentions:
{IntentionSpace}
The action space includes the following actions:
{ActionSpace}
You need to plan your action based on your observations of
the other agent's actions, your own actions, and the
world states,
your estimation of the other agent's intention and value,
your own intent, and your own value.
Remember that some of the other agent's actions and world
states may be unobservable because of the limited field
of view.
Each selected action should be written in this format:
"agent_i-action" (e.g., Agent_1-Putonto-Timer_3-Table_4).
Here are the given observations:
{step_prompt}
Your value is: {your_value}
Let's think step by step and output the three most possible
actions and the corresponding confidences in the
following format:
"My selected action is: {{ "most_possible_action": <action>,
"most_possible_action_cf": <confidence>,
"second_possible_action": <action>,
"second_possible_action_cf": <confidence>,
"third_possible_action": <action>,
"third_possible_action_cf": <confidence>}}"
'''
```

621 5.6 Prompt example

622 5.6.1 Intention estimation with partial observation

Prompt example

The following is the prompt example used for the task “intention estimation with partial observation”.

```
"prompt": "\n Imagine you and the other agent are in a room. \n \n The coordinate system follows
a Cartesian plane, where the x-axis is horizontal, \n pointing to the right, and the y-axis
is vertical, pointing upwards. The rotation angle \n starts at the positive x-axis (0),
increases counterclockwise (positive angle).
You are Agent_1, and the other agent is Agent_2. \n The intention space includes the following
intentions: \n ['PutOnto'/'PutInto', <something1>, <something2>] put <something1>
onto/into <something2> \n \n ['Give', <something>, <somebody>]: give <something> to <
somebody> \n \n ['Get', <something>, <somebody>]: get <something> from <
somebody> or <somebody> \n \n ['Find'/'Open'/'Observe', <something>] \n \n ['PlayWith', <
something>, <somebody>]: play <something> with <somebody> \n \n ['RespondTo'/'Greet', <
somebody>] \n \n ['Inform', <something>, <somebody>]: inform <somebody> of <something> \n \n
['Help', <somebody>, <intention>]: help <somebody> to achieve <intention> \n \n
['RequestHelp', <somebody>, <intention>]: request help from <somebody> to achieve <
intention> \n \n ['Harm', <somebody>, <intention>]: prevent <somebody> from achieving <
intention> \n \n ['Na']: no intent \n \n \n
The action space includes the following actions: \n
['ActionMoveTo'/'ActionRotateTo'/'ActionPointTo', <somebody/something>] move/rotate
to/point to <somebody> or <something> \n \n ['ActionGiveTo', <something>, <somebody>]: give <
something> to <somebody> \n \n ['ActionWaveHand'/'ActionNodHead'/'ActionShakeHead', <
somebody>] wave hand / nod head / shake head to <somebody> \n \n
['ActionPlay'/'ActionPutDown'/'ActionClose'/'ActionOpen'/'ActionUnlock'/'ActionGrab', <
something>] \n \n ['ActionPutInto'/'ActionPutOnto', <something1>, <something2>]: put <
something1> into/onto <something2> \n \n ['ActionFollowPointing', <somebody>]: follow <
somebody>'s pointing \n \n ['ActionMoveToAttention', <somebody>]: move to <somebody>'s
attention \n \n ['ActionPerform', 'eat'/'drink']: perform to eat or drink \n \n
['ActionSmash', 'cup'] \n \n ['ActionEat', 'banana'] \n \n ['ActionSpeak', 'Hello'/'Thank
you'] \n \n ['ActionWait'] \n \n \n You need to infer the other agent's intention based on your
observations of the other agent's actions, your own actions, and the world states. \n
Remember that some of the other agent's actions and world states may be unobservable
because of the limited field of view. \n Each inferred intention should be written in this
format: \"agent_i-intent\" (e.g., Agent_1-PutOnto-Timer_3-Table_4). \n Here are the given
observations: \n \n
Step 1 : \n other_action: The other agent does nothing \n your_action:
Agent_1-ActionMoveTo-Table_4 \n world_state observed by you: agents and objects:
['table_4'] , and There is nothing on table_4. table_4 is at [175, -95]. , ['table_4'] are
reachable for you. , \n \n Step 2 : \n other_action: You can not observe other's action \n
your_action: You do nothing \n world_state observed by you: agents and objects: ['table_4']
, and The observed state is unchanged. , ['table_4'] are reachable for you. , \n \n Step 3
: \n other_action: The other agent does nothing \n your_action:
Agent_1-ActionMoveTo-Timer_6 \n world_state observed by you: agents and objects:
['timer_6'] , and The following information is changed: There is nothing on table_4.
table_4 is at [175, -95]. The following information is new: timer_6 is at [175, 25]. ,
['timer_6'] are reachable for you. , \n \n Step 4 : \n other_action: You can not observe
other's action \n your_action: You do nothing \n world_state observed by you: agents and
objects: ['timer_6'] , and The observed state is unchanged. , ['timer_6'] are reachable
for you. , \n \n Step 5 : \n other_action: The other agent does nothing \n your_action:
Agent_1-ActionGrab-Timer_6 \n world_state observed by you: agents and objects: ['timer_6']
, and The following information is changed: timer_6 is at [175, 25]. The following
information is new: timer_6 is at [129, 100]. , ['timer_6'] are reachable for you. , \n \n
Step 6 : \n other_action: You can not observe other's action \n your_action: You do
nothing \n world_state observed by you: agents and objects: ['timer_6'] , and The observed
state is unchanged. , ['timer_6'] are reachable for you. , \n \n Step 7 : \n other_action:
The other agent does nothing \n your_action: Agent_1-ActionWait \n world_state observed by
you: agents and objects: ['timer_6'] , and The observed state is unchanged. , ['timer_6']
are reachable for you. , \n \n Step 8 : \n other_action: You can not observe other's
action \n your_action: You do nothing \n world_state observed by you: agents and objects:
['timer_6'] , and The observed state is unchanged. , ['timer_6'] are reachable for you. ,
\n \n Step 9 : \n other_action: The other agent does nothing \n your_action:
Agent_1-ActionWait \n world_state observed by you: agents and objects: ['timer_6'] , and
The observed state is unchanged. , ['timer_6'] are reachable for you. , \n \n Step 10 : \n
other_action: You can not observe other's action \n your_action: You do nothing \n
world_state observed by you: agents and objects: ['timer_6', 'dumbbell_3'] , and The
following information is changed: There is no new information. , ['timer_6', 'dumbbell_3']
are reachable for you. , \n \n Let's think step by step and output the three most possible
intentions and the corresponding confidences in the following format: \n \"My inferred
intention of the other agent is: { \"most_possible_intention\": <intention>,
\"most_possible_intention_cf\": <confidence>, \"second_possible_intention\": <intention>,
\"second_possible_intention_cf\": <confidence>, \"third_possible_intention\": <intention>,
\"third_possible_intention_cf\": <confidence>}\" \n ,
```

Prompt example

The following is the prompt example used for the task “intention estimation with full observation”.

```
"prompt": "\n Imagine there are two agents in a room. \n \n The coordinate system follows a
Cartesian plane, where the x-axis is horizontal, \n pointing to the right, and the
y-axis is vertical, pointing upwards. The rotation angle \n starts at the positive
x-axis (0), increases counterclockwise (positive angle).\n \n The first agent is
Agent_2 and the second agent is Agent_1. \n The intention space includes the
following intentions: \n ['PutOnto'/'PutInto', <something1>, <something2>] put <
something1> onto/into <something2> \n \n ['Give', <something>, <somebody>]: give <
something> to <somebody> \n \n ['Get', <something>, <somebody>]: get <
something> from <somebody> or <somebody> \n \n ['Find'/'Open'/'Observe', <
something>] \n \n ['PlayWith', <something>, <somebody>]: play <something> with <
somebody> \n \n ['RespondTo'/'Greet', <somebody>] \n \n ['Inform', <somebody>, <
something>]: inform <somebody> of <something> \n \n ['Help', <somebody>, <intention>]:
help <somebody> to achieve <intention> \n \n ['RequestHelp', <somebody>, <intention>]:
request help from <somebody> to achieve <intention> \n \n ['Harm', <somebody>, <
intention>]: prevent <somebody> from achieving <intention> \n \n ['Na']: no intent
\n \n The action space includes the following actions: \n
['ActionMoveTo'/'ActionRotateTo'/'ActionPointTo', <somebody>/<something>] move/rotate
to/point to <somebody> or <something> \n \n ['ActionGiveTo', <something>, <somebody>]:
give <something> to <somebody> \n \n
['ActionWaveHand'/'ActionNodHead'/'ActionShakeHead', <somebody>] wave hand / nod
head / shake head to <somebody> \n \n
['ActionPlay'/'ActionPutDown'/'ActionClose'/'ActionOpen'/'ActionUnlock'/'ActionGrab',
<something>] \n \n ['ActionPutInto'/'ActionPutOnto', <something1>, <something2>]: put <
something1> into/onto <something2> \n \n ['ActionFollowPointing', <somebody>]: follow <
somebody>'s pointing \n \n ['ActionMoveToAttention', <somebody>]: move to <somebody>'s
attention \n \n ['ActionPerform', 'eat'/'drink']: perform to eat or drink \n \n
['ActionSmash', 'cup'] \n \n ['ActionEat', 'banana'] \n \n ['ActionSpeak',
'Hello'/'Thank you'] \n \n ['ActionWait'] \n \n You need to infer the first agent's
intention from a god's eye view based on the actions of the two agents as well as
the world states. \n The inferred intention should be written in this format:
\"agent_1-intent\" (e.g., Agent_1-PutOnto-Timer_3-Table_4). \n Here are the given
observations: \n \n Step 1 : \n The first agent's action: The first agent does
nothing \n The second agent's action: Agent_1-ActionPointTo-Key_5 \n world_state:
agents and objects: ['agent_1', 'agent_2', 'box_4', 'box_6', 'cup_7', 'key_5',
'dumbbell_3'], and agent_1 is pointing. agent_1 can observe ['agent_2', 'box_4',
'dumbbell_3']. agent_1 is at [-380, 256]. agent_1 is facing -70. agent_2 can observe
['agent_1', 'key_5']. agent_2 is at [-543, -310]. agent_2 is facing 67. box_4 is
closed. box_4 is locked. box_4 is at [-86, -50]. box_6 is closed. box_6 is at [521,
52]. cup_7 is contained in box_6. cup_7 is at [521, 52]. key_5 is at [-142, 258].
dumbbell_3 is contained in box_4. dumbbell_3 is at [-86, -50]. , nothing is
reachable for both agents , \n \n Step 2 : \n The first agent's action:
Agent_2-ActionMoveTo-Agent_1 \n The second agent's action: The second agent does
nothing \n world_state: agents and objects: ['agent_1', 'agent_2', 'box_4', 'box_6',
'cup_7', 'key_5', 'dumbbell_3'], and The following information is changed: agent_2
can observe ['agent_1', 'key_5']. agent_2 is at [-543, -310]. agent_2 is facing 67.
The following information is new: agent_2 can observe ['agent_1']. agent_2 is at
[-380, 156]. agent_2 is facing 90. , nothing is reachable for both agents , \n \n
Let's think step by step and output the three most possible intentions and the
corresponding confidences in the following format: \n \"My inferred intention of the
first agent is: { \"most_possible_intention\": <intention>,
\"most_possible_intention_cf\": <confidence>, \"second_possible_intention\": <
intention>, \"second_possible_intention_cf\": <confidence>,
\"third_possible_intention\": <intention>, \"third_possible_intention_cf\": <
confidence>}\" \n \n ,
```


Prompt example

The following is the prompt example used for the task “value estimation with partial observation”.

```
"prompt": "\n Imagine you and the other agent are in a room. \n \n The coordinate system follows
a Cartesian plane, where the x-axis is horizontal, \n pointing to the right, and the y-axis
is vertical, pointing upwards. The rotation angle \n starts at the positive x-axis (0),
increases counterclockwise (positive angle). \n \n
You are Agent_1, and the other agent is Agent_2. \n The action space includes the following
actions: \n ['ActionMoveTo','ActionRotateTo','ActionPointTo', <somebody/something>]
move/rotate to/point to <somebody> or <something> \n \n ['ActionGiveTo', <something>, <
somebody>]: give <something> to <somebody> \n \n
['ActionWaveHand','ActionNodHead','ActionShakeHead', <somebody>] wave hand / nod head /
shake head to <somebody> \n \n
['ActionPlay','ActionPutDown','ActionClose','ActionOpen','ActionUnlock','ActionGrab', <
something>] \n \n ['ActionPutInto','ActionPutOnto', <something1>, <something2>]: put <
something1> into/onto <something2> \n \n ['ActionFollowPointing', <somebody>]: follow <
somebody>'s pointing \n \n ['ActionMoveToAttention', <somebody>]: move to <somebody>'s
attention \n \n ['ActionPerform', 'eat','drink']: perform to eat or drink \n \n
['ActionSmash', 'cup'] \n \n ['ActionEat', 'banana'] \n \n ['ActionSpeak', 'Hello','Thank
you'] \n \n ['ActionWait'] \n \n \n
The value space includes the following values: \n There are three value dimensions: \"Active\",
\"Social\", \"Helpful\". \n The \"Active\" value dimension measures the individual's
energy level and preference for physical motion. \n The possible scores of \"Active\" are:
0 (inactive), 0.5 (neutral), 1 (active). \n \n The \"Social\" value dimension assesses the
individual's inclination towards social communication and interactions with others. \n The
possible scores of \"Social\" are: 0 (unsocial), 0.5 (neutral), 1 (social). \n \n The
\"Helpful\" value dimension evaluates the individual's propensity to assist others. \n The
possible scores are (Note: There is no neutral value in this dimension): \n -1 (harmful) -
inclined to hinder others from achieving their goals; \n 0 (unhelpful) - shows no interest
in helping others; \n 0.5 (neutral) - reflects a moderate willingness to provide help; \n 1
(helpful) - somewhat willing to provide assistance to others; \n 2 (very helpful) -
demonstrates a strong willingness to provide help. \n \n You need to infer the other agent's
value based on your observations of the other agent's actions, your own actions, and the
world states. \n Remember that some of the other agent's actions and world states may be
unobservable because of the limited field of view. \n Here are the given observations: \n \n
Step 1 : \n other_action: The other agent does nothing \n your_action:
Agent_1-ActionMoveTo-Table_4 \n world_state observed by you: agents and objects:
['table_4'] , and There is nothing on table_4. table_4 is at [175, -95]. , ['table_4'] are
reachable for you. , \n \n Step 2 : \n other_action: You can not observe other's action \n
your_action: You do nothing \n world_state observed by you: agents and objects: ['timer_6']
, and The following information is changed: There is nothing on table_4. table_4 is at
[175, -95]. The following information is new: timer_6 is at [175, 25]. , ['timer_6'] are
reachable for you. , \n \n Step 3 : \n other_action: The other agent does nothing \n
your_action: Agent_1-ActionMoveTo-Timer_6 \n world_state observed by you: agents and
objects: ['timer_6'] , and The following information is changed: timer_6 is at [175, 25].
The following information is new: timer_6 is at [129, 100]. , ['timer_6'] are reachable for
you. , \n \n Step 4 : \n other_action: You can not observe other's action \n your_action: You
do nothing \n world_state observed by you: agents and objects: ['timer_6'] , and The
observed state is unchanged. , ['timer_6'] are reachable for you. , \n \n Step 5 : \n
other_action: The other agent does nothing \n your_action: Agent_1-ActionGrab-Timer_6 \n
world_state observed by you: agents and objects: ['timer_6'] , and The observed state is
unchanged. , ['timer_6'] are reachable for you. , \n \n Step 6 : \n other_action: You can
not observe other's action \n your_action: You do nothing \n world_state observed by you:
agents and objects: ['timer_6', 'dumbbell_3'] , and The following information is
changed: There is no new information. , ['timer_6', 'dumbbell_3'] are reachable for you. ,
\n \n Step 7 : \n other_action: The other agent does nothing \n your_action:
Agent_1-ActionWait \n world_state observed by you: agents and objects: ['timer_6',
'dumbbell_3'] , and The observed state is unchanged. , ['timer_6', 'dumbbell_3'] are
reachable for you. , \n \n Step 8 : \n other_action: You can not observe other's action \n
your_action: You do nothing \n world_state observed by you: agents and objects: ['timer_6',
'dumbbell_3'] , and The observed state is unchanged. , ['timer_6', 'dumbbell_3'] are
reachable for you. , \n \n Step 9 : \n other_action: The other agent does nothing \n
your_action: Agent_1-ActionWait \n world_state observed by you: agents and objects:
['timer_6', 'dumbbell_3'] , and The observed state is unchanged. , ['timer_6',
'dumbbell_3'] are reachable for you. , \n \n Step 10 : \n other_action: You can not observe
other's action \n your_action: You do nothing \n world_state observed by you: agents and
objects: ['timer_6', 'dumbbell_3'] , and The observed state is unchanged. , ['timer_6',
'dumbbell_3'] are reachable for you. , \n \n Let's think step by step and output the
estimated value and the corresponding confidence in this format: \n \"My estimated value
of the other agent is: {\"active\": <score>, \"active_cf\": <confidence>, \"social\": <
score>, \"social_cf\": <confidence>, \"helpful\": <score>, \"helpful_cf\": <
confidence>}.\" \n \" ,
```


Prompt example

The following is the prompt example used for the task “intention updating”.

```
"prompt": "\n Imagine you and the other agent are in a room. \n \n The coordinate system follows
a Cartesian plane, where the x-axis is horizontal, \n pointing to the right, and the y-axis
is vertical, pointing upwards. The rotation angle \n starts at the positive x-axis (0),
increases counterclockwise (positive angle).\n \n You are Agent_2, and the other agent is
Agent_1. \n The intention space includes the following intentions: \n
['PutOnto','PutInto', <something1>, <something2>] put <something1> onto/into <something2>
\n \n ['Give', <something>, <somebody>]: give <something> to <somebody>\n \n ['Get', <
something>, <somewhere/somebody>]: get <something> from <somewhere> or <somebody>\n \n
['Find','Open','Observe', <something>]\n \n ['PlayWith', <something>, <somebody>]: play <
something> with <somebody>\n \n ['RespondTo','Greet', <somebody>]\n \n ['Inform', <
somebody>, <something>]: inform <somebody> of <something>\n \n ['Help', <somebody>, <
intention>]: help <somebody> to achieve <intention>\n \n ['RequestHelp', <somebody>, <
intention>]: request help from <somebody> to achieve <intention>\n \n ['Harm', <somebody>, <
intention>]: prevent <somebody> from achieving <intention>\n \n ['Na']: no intent \n \n \n
The action space includes the following actions: \n
['ActionMoveTo','ActionRotateTo','ActionPointTo', <somebody/something>] move/rotate
to/point to <somebody> or <something>\n \n ['ActionGiveTo', <something>, <somebody>]: give <
something> to <somebody>\n \n ['ActionWaveHand','ActionNodHead','ActionShakeHead', <
somebody>] wave hand / nod head / shake head to <somebody>\n \n
['ActionPlay','ActionPutDown','ActionClose','ActionOpen','ActionUnlock','ActionGrab', <
something>]\n \n ['ActionPutInto','ActionPutOnto', <something1>, <something2>]: put <
something1> into/onto <something2>\n \n ['ActionFollowPointing', <somebody>]: follow <
somebody>'s pointing\n \n ['ActionMoveToAttention', <somebody>]: move to <somebody>'s
attention\n \n ['ActionPerform', 'eat','drink']: perform to eat or drink\n \n
['ActionSmash', 'cup']\n \n ['ActionEat', 'banana']\n \n ['ActionSpeak', 'Hello','Thank
you']\n \n ['ActionWait']\n \n \n You need to update your own intention based on your
observations of the other agent's actions, your own actions, and the world states, \n your
estimation of the other agent's intention and value, your own value and your previous
intents. \n Remember that some of the other agent's actions and world states may be
unobservable because of the limited field of view.\n Each updated intention should be
written in this format: \"agent_i-intent\" (e.g., Agent_1-Putonto-Timer_3-Table_4). \n
Here are the given observations:\n \n Step 1 :\n other_action:
Agent_1-ActionMoveTo-Table_4\n your_action: You do nothing\n world_state observed by you:
agents and objects: ['agent_1', 'box_5', 'table_4', 'timer_6', 'dumbbell_3'], and In your
current observation, agent_1 can observe ['table_4'] and can reach ['table_4']. agent_1 is
at [50, -95]. agent_1 is facing 0. box_5 is closed. box_5 is at [243, 513]. There is
nothing on table_4. table_4 is at [175, -95]. timer_6 is at [175, 25]. dumbbell_3 is
contained in box_5. dumbbell_3 is at [243, 513]. dumbbell_3 is contained in box_5. ,
nothing is reachable for you , \n In this step, your intent is: Agent_2-Na.\n Step 2 :\n
other_action: The other agent does nothing\n your_action: You are deciding what to do\n
world_state observed by you: agents and objects: ['agent_1', 'box_5', 'table_4',
'timer_6', 'dumbbell_3'] , and The observed state is unchanged. , nothing is reachable for
you , \n In this step, your inference about the other agent's intent is:
Agent_1-Putinto-Timer_6-Na.In this step, your inference about the other agent's value is:
{'active': None, 'social': None, 'helpful': None}.Now you are updating your intent based
on the given information.\n Your value is: [{'active': 1.0, 'social': 0.0, 'helpful':
2.0}]\n Let's think step by step and output the three most possible intentions and the
corresponding confidences in the following format: \n \n My updated intention is: {
'most_possible_intention': <intention>, 'most_possible_intention_cf': <confidence>,
'second_possible_intention': <intention>, 'second_possible_intention_cf': <
confidence>, 'third_possible_intention': <intention>, 'third_possible_intention_cf': <
confidence>}\n \n",
```

Prompt example

The following is the prompt example used for the task “social interaction policy”.

```
"prompt": "\n Imagine you and the other agent are in a room. \n \n The coordinate system follows
a Cartesian plane, where the x-axis is horizontal, \n pointing to the right, and the y-axis
is vertical, pointing upwards. The rotation angle \n starts at the positive x-axis (0),
increases counterclockwise (positive angle).\n \n You are Agent_2, and the other agent is
Agent_1. \n The intention space includes the following intentions: \n
['PutOnto','PutInto', <something1>, <something2>] put <something1> onto/into <something2>
\n \n ['Give', <something>, <somebody>]: give <something> to <somebody> \n \n ['Get', <
something>, <somewhere/somebody>]: get <something> from <somewhere> or <somebody> \n \n
['Find','Open','Observe', <something>] \n \n ['PlayWith', <something>, <somebody>]: play <
something> with <somebody> \n \n ['RespondTo','Greet', <somebody>] \n \n ['Inform', <
somebody>, <something>]: inform <somebody> of <something> \n \n ['Help', <somebody>, <
intention>]: help <somebody> to achieve <intention> \n \n ['RequestHelp', <somebody>, <
intention>]: request help from <somebody> to achieve <intention> \n \n ['Harm', <somebody>, <
intention>]: prevent <somebody> from achieving <intention> \n \n ['Na']: no intent \n \n \n
The action space includes the following actions: \n
['ActionMoveTo','ActionRotateTo','ActionPointTo', <somebody/something>] move/rotate
to/point to <somebody> or <something> \n \n ['ActionGiveTo', <something>, <somebody>]: give <
something> to <somebody> \n \n ['ActionWaveHand','ActionNodHead','ActionShakeHead', <
somebody>] wave hand / nod head / shake head to <somebody> \n \n
['ActionPlay','ActionPutDown','ActionClose','ActionOpen','ActionUnlock','ActionGrab', <
something>] \n \n ['ActionPutInto','ActionPutOnto', <something1>, <something2>]: put <
something1> into/onto <something2> \n \n ['ActionFollowPointing', <somebody>]: follow <
somebody>'s pointing \n \n ['ActionMoveToAttention', <somebody>]: move to <somebody>'s
attention \n \n ['ActionPerform', 'eat','drink']: perform to eat or drink \n \n
['ActionSmash', 'cup'] \n \n ['ActionEat', 'banana'] \n \n ['ActionSpeak', 'Hello','Thank
you'] \n \n ['ActionWait'] \n \n \n You need to plan your action based on your observations of
the other agent's actions, your own actions, and the world states, \n your estimation of
the other agent's intention and value, your own intent, and your own value. \n Remember
that some of the other agent's actions and world states may be unobservable because of the
limited field of view. \n Each selected action should be written in this format:
\agent_i-action\" (e.g., Agent_1-Putonto-Timer_3-Table_4). \n Here are the given
observations: \n \n Step 1 : \n other_action: Agent_1-ActionPointTo-Key_5 \n your_action: You
do nothing \n world_state observed by you: agents and objects: ['agent_1', 'key_5'], and
agent_1 is pointing. agent_1 can observe me. In your current observation, agent_1 can
observe nothing and can reach nothing. agent_1 is at [-380, 256]. agent_1 is facing -70.
key_5 is at [-142, 258]. , nothing is reachable for you , \n In this step, your intent is:
Agent_2-Na. \n Step 2 : \n other_action: The other agent does nothing \n your_action: You are
deciding what to do \n world_state observed by you: agents and objects: ['agent_1',
'key_5'], and The observed state is unchanged. , nothing is reachable for you , \n In this
step, your inference about the other agent's intent is: Agent_1-Find-Key_5. In this step,
your inference about the other agent's value is: {'active': None, 'social': None,
'helpful': None}. In this step, your intent is: Agent_2-Na. \n Your value is: {'active':
1.0, 'social': 0.0, 'helpful': 1.0} \n Let's think step by step and output the three most
possible actions and the corresponding confidences in the following format: \n \n My
selected action is: { \"most_possible_action\": <action>, \"most_possible_action_cf\": <
confidence>, \"second_possible_action\": <action>, \"second_possible_action_cf\": <
confidence>, \"third_possible_action\": <action>, \"third_possible_action_cf\": <
confidence>} \n \n ,
```

633

6 Experiment results

This section presents a comprehensive analysis of our experimental findings, comparing the performance of various large language models (LLMs) against human benchmarks in social cognition tasks. We examine the distribution patterns and accuracy metrics of both LLMs and human participants across four key areas: intention estimation, value estimation, intention updating, and action prediction. Through detailed visualizations and statistical analyses, we demonstrate how different models exhibit distinct biases and performance characteristics under both partial and full observation conditions. The distributions reveal that LLMs show specific preferences in intent estimation, with a tendency toward estimating intents like "play with," "get," and "request help" more frequently than humans. For value estimation, we observe model-specific tendencies in the "Active," "Social," and "Helpful" dimensions, with some models preferring extreme values while others favor more moderate assessments. Notably, estimations under agent view conditions generally show higher accuracy than those under world view for most dimensions. The intention updating and action prediction analyses further illuminate how LLMs differ from human reasoning patterns in social contexts. These findings are quantified through comprehensive performance metrics including accuracy, confidence scores, and correlation coefficients, allowing for precise evaluation of each model's strengths and limitations in understanding and predicting social interactions.



Figure 7: Comparison of distribution of intent estimation results between humans and LLMs.

6.1 Data statistics and examples

We compared the distribution of results between humans and LLMs to see the preference bias. The distributions of intent estimation are shown in Fig. 7. It is better to summarize the results of the top three because it includes more possible intentions in the estimation, which is quite natural. Compared with results under agent view, estimations under world view seldom choose “na”. LLMs tend to estimate the intent as “play with”, “get”, and “request help”.

The distributions of value estimation are shown in Fig. 8. Llama3 presents most estimations that are apart from peak values 0, 0.5, 1. For the “Active” dimension, most of the LLMs, except Gemini, take the medium value as the majority. All the LLMs, except gpt-4o, tend to avoid a low value. DeepSeek-R1 and Gemini prefer to estimate a high value, while Claude, gpt-4o, and Llama3 prefer a medium value. The LLMs’ estimations under the agent view are more accurate than those under the world view. For the “Social” dimension, the estimations under the agent view and the world view are close. Similar to the “Active” dimension, Claude, DeepSeek-R1, Gemini, and Qwen3-8B prefer to estimate a high value, while gpt-4o and Llama3 prefer a medium one. For the “Helpful” dimension, LLMs except Claude and Qwen3-8B remarkably prefer a low value or a high value. Qwen3-8B reaches the strongest consistency with human estimations. The LLMs’ estimations under the agent view are slightly more accurate than those under the world view.

The distributions of intent updating prediction are shown in Fig. 9. Most of the LLMs show an overly strong preference. Llama3 shows a preference for “help”. The predictions are slightly more accurate under the agent view.

The distributions of action prediction are shown in Fig. 10. Under the agent view, most of the LLMs show no preference for “RotateTo” and a preference for “Grab”. The prediction of LLMs under the world view is relatively consistent with human prediction.

6.2 Total results

Tab. 4, Tab. 5, Tab. 6, Tab. 7, Tab. 8, and Tab. 9 summarize the comprehensive results of our experiments. The corresponding evaluation metrics are described in the paper.

References

- Michael Bratman. Intention, plans, and practical reason. 1987. 7
- Michael Georgeff, Barney Pell, Martha Pollack, Milind Tambe, and Michael Wooldridge. The belief-desire-intention model of agency. In *Intelligent Agents V: Agents Theories, Architectures, and Languages: 5th International Workshop*, 1999. 7

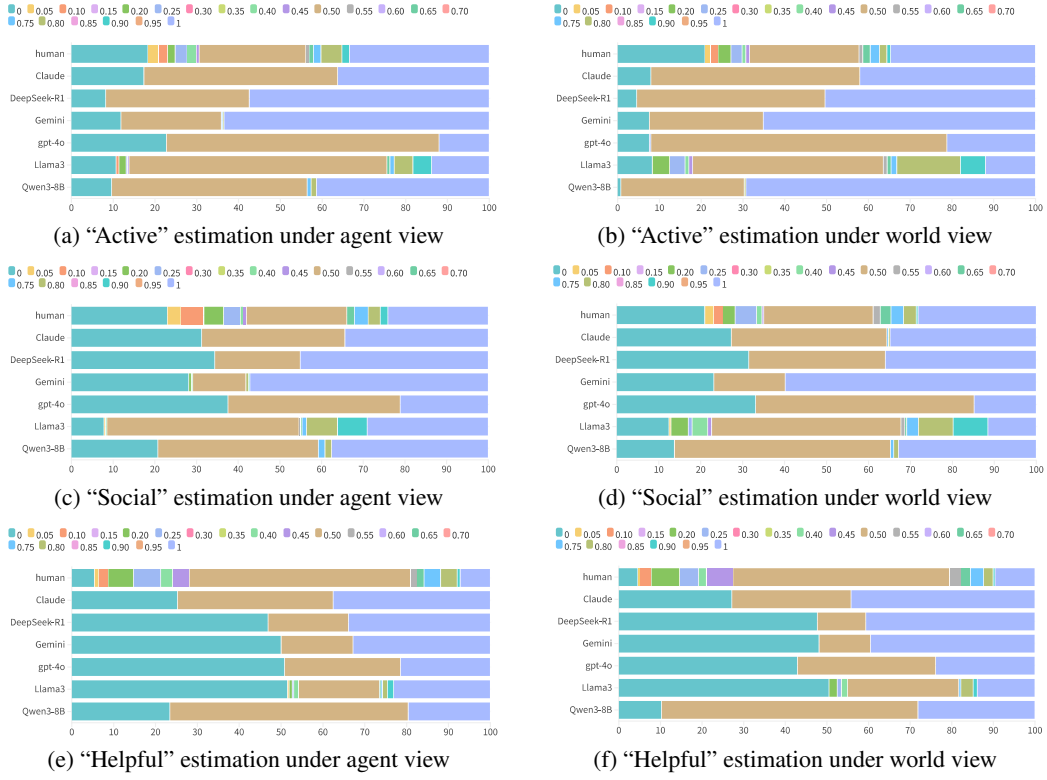


Figure 8: Comparison of distribution of value estimation results between humans and LLMs.

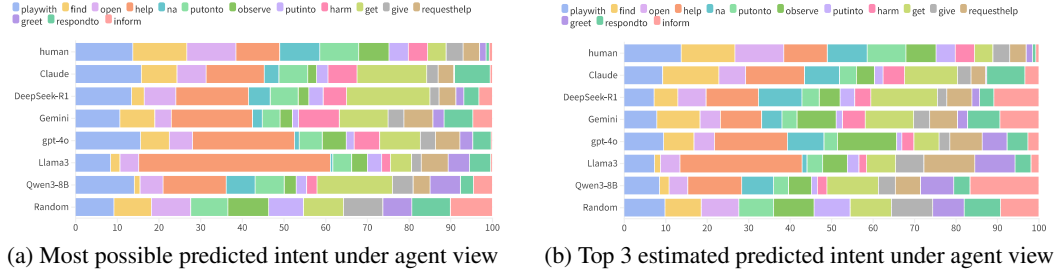


Figure 9: Comparison of distribution of intent prediction results between humans and LLMs.

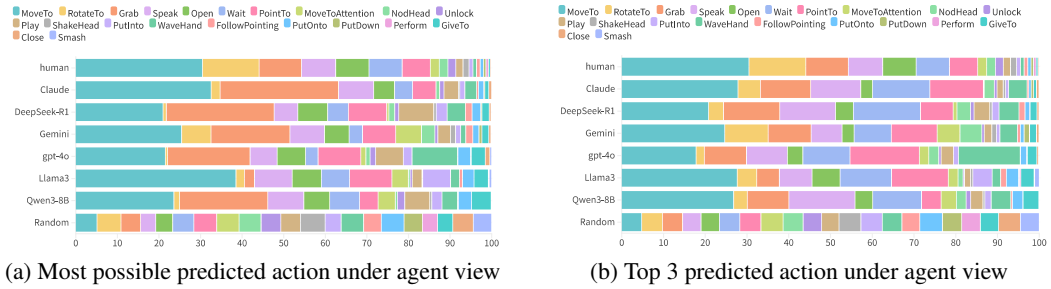


Figure 10: Comparison of distribution of action prediction results between humans and LLMs.

Table 4: Value estimation performance with full observation across different LLMs. *Dis.@A/S/H* denote distance metrics for Active, Social, and Helpful dimensions; *Cf.@A/S/H* denote confidence scores; *Pcc@A/S/H* denote Pearson correlation coefficients with standard errors; *Dis.@T* denotes total distance.

model	Dis.@A	Cf.@A	Pcc.@A	Dis.@S	Cf.@S	Pcc.@S	Dis.@H	Cf.@H	Pcc.@H	Dis.@T
Qwen3-8B	0.425	0.816	-0.147 ± 0.118	0.404	0.731	-0.183 ± 0.117	0.996	0.666	-0.042 ± 0.121	1.825
Llama3	0.443	0.768	-0.088 ± 0.120	0.468	0.709	-0.003 ± 0.121	1.091	0.659	0.036 ± 0.121	2.002
gpt-4o	0.427	0.748	-0.261 ± 0.113	0.381	0.718	-0.079 ± 0.120	1	0.747	-0.033 ± 0.121	1.808
Gemini	0.383	0.833	-0.187 ± 0.117	0.377	0.826	-0.097 ± 0.120	1.015	0.79	-0.051 ± 0.120	1.775
Claude	0.413	0.77	-0.268 ± 0.112	0.351	0.742	-0.312 ± 0.109	0.962	0.664	-0.095 ± 0.120	1.726
DeepSeek-R1	0.384	0.77	-0.289 ± 0.111	0.326	0.755	-0.277 ± 0.112	1.009	0.668	-0.023 ± 0.121	1.72
Random	0.498	1	nan ± nan	0.494	1	nan ± nan	1.117	1	nan ± nan	2.109

Table 5: Value estimation performance with partial observation across different LLMs. *Dis.@A/S/H* denote distance metrics for Active, Social, and Helpful dimensions; *Cf.@A/S/H* denote confidence scores; *Pcc@A/S/H* denote Pearson correlation coefficients with standard errors; *Dis.@T* denotes total distance.

model	Dis.@A	Cf.@A	Pcc.@A	Dis.@S	Cf.@S	Pcc.@S	Dis.@H	Cf.@H	Pcc.@H	Dis.@T
Qwen3-8B	0.307	0.695	-0.039 ± 0.110	0.305	0.678	-0.015 ± 0.110	0.273	0.6	0.197 ± 0.106	0.885
Llama3	0.312	0.693	0.032 ± 0.110	0.353	0.733	0.045 ± 0.110	0.532	0.657	0.129 ± 0.108	1.197
gpt-4o	0.321	0.709	-0.051 ± 0.110	0.287	0.737	-0.100 ± 0.109	0.409	0.7	0.326 ± 0.099	1.018
Gemini	0.349	0.787	-0.053 ± 0.110	0.343	0.789	0.220 ± 0.105	0.601	0.708	0.483 ± 0.085	1.293
Claude	0.319	0.696	-0.020 ± 0.110	0.294	0.674	-0.113 ± 0.109	0.349	0.557	0.426 ± 0.090	0.962
DeepSeek-R1	0.323	0.705	-0.019 ± 0.110	0.29	0.722	-0.074 ± 0.110	0.659	0.594	0.424 ± 0.091	1.272
Random	0.402	1	nan ± nan	0.393	1	nan ± nan	0.72	1	nan ± nan	1.514

682 Alicia P Melis and Michael Tomasello. Chimpanzees (pan troglodytes) coordinate by communi-
683 cating in a collaborative problem-solving task. *Proceedings of the Royal Society B*, 286(1901):
684 20190408, 2019. 5

Table 6: Intention estimation with full observation results. Metrics include accuracy (Acc.@n) for top-n results, confidence (Cf.@n), and confidence discrimination (Cfd.@n). Evaluations distinguish between predicate-only (@nP), object-only(@nO), and complete intention (@nT) assessments. Only success cases (@nxS) and only failure cases (@nxF) are listed when computing confidence.

model	Acc.@1P	Cf.@1PS	Cf.@1PF	Cfd.@1P	Acc.@3P	Cf.@3PS	Acc.@1O	Cf.@1OS	Cf.@1OF	Cfd.@1O	Acc.@3O	Cf.@3OS	Acc.@1T	Cf.@1TS	Cf.@1TF	Cfd.@1T	Acc.@3T	Cf.@3TS
Qwen3-8B	0.146	0.908	0.884	0.024	0.289	0.729	0.477	0.897	0.88	0.017	0.548	0.86	0.096	0.921	0.884	0.037	0.166	0.74
Llama3	0.086	0.673	0.674	-0.001	0.215	0.4	0.426	0.688	0.663	0.025	0.489	0.628	0.03	0.681	0.673	0.007	0.052	0.461
gpt-4o	0.196	0.873	0.855	0.018	0.355	0.721	0.53	0.867	0.849	0.018	0.602	0.837	0.122	0.887	0.855	0.033	0.208	0.726
Gemini	0.197	0.859	0.839	0.021	0.363	0.758	0.569	0.852	0.831	0.021	0.644	0.829	0.137	0.872	0.838	0.033	0.251	0.767
Claude	0.193	0.699	0.647	0.052	0.399	0.429	0.565	0.678	0.629	0.049	0.657	0.615	0.154	0.718	0.646	0.072	0.264	0.497
DeepSeek-R1	0.174	0.783	0.713	0.07	0.354	0.574	0.569	0.741	0.704	0.037	0.658	0.695	0.122	0.815	0.712	0.103	0.222	0.606
Random	0.051	0.9	0.9	0	0.139	0.515	0.367	0.9	0.9	0	0.455	0.777	0.007	0.9	0.9	0	0.015	0.544

Table 7: Intention estimation with partial observation results. Metrics include accuracy (Acc.@n) for top-n results, confidence (Cf.@n), and confidence discrimination (Cfd.@n). Evaluations distinguish between predicate-only (@nP), object-only(@nO), and complete intention (@nT) assessments. Only success cases (@nxS) and only failure cases (@nxF) are listed when computing confidence.

model	Acc.@1P	Cf.@1PS	Cf.@1PF	Cfd.@1P	Acc.@3P	Cf.@3PS	Acc.@1O	Cf.@1OS	Cf.@1OF	Cfd.@1O	Acc.@3O	Cf.@3OS	Acc.@1T	Cf.@1TS	Cf.@1TF	Cfd.@1T	Acc.@3T	Cf.@3TS
Qwen3-8B	0.21	0.921	0.861	0.059	0.366	0.752	0.59	0.879	0.867	0.012	0.606	0.873	0.151	0.935	0.863	0.072	0.237	0.772
Llama3	0.154	0.826	0.697	0.129	0.266	0.562	0.567	0.709	0.729	-0.02	0.576	0.702	0.102	0.896	0.697	0.199	0.154	0.665
gpt-4o	0.151	0.822	0.804	0.018	0.374	0.59	0.592	0.8	0.816	-0.017	0.611	0.791	0.074	0.803	0.807	-0.004	0.199	0.508
Gemini	0.218	0.748	0.801	-0.053	0.396	0.651	0.592	0.779	0.803	-0.024	0.609	0.777	0.123	0.677	0.805	-0.128	0.228	0.573
Claude	0.242	0.629	0.613	0.016	0.443	0.441	0.598	0.617	0.617	-0.001	0.622	0.602	0.157	0.612	0.618	-0.006	0.276	0.445
DeepSeek-R1	0.179	0.697	0.647	0.051	0.387	0.484	0.598	0.655	0.657	-0.003	0.62	0.645	0.106	0.703	0.65	0.053	0.217	0.471
Random	0.036	0.9	0.9	0	0.098	0.523	0.566	0.9	0.9	0	0.583	0.884	0.001	0.9	0.9	0	0.007	0.42

Table 8: Intention updating results. Metrics include accuracy (Acc.@n) for top-n results, confidence (Cf.@n), and confidence discrimination (Cfd.@n). Evaluations distinguish between predicate-only (@nP), object-only(@nO), and complete intention (@nT) assessments. Only success cases (@nxS) and only failure cases (@nxF) are listed when computing confidence.

model	Acc.@1P	Cf.@1PS	Cf.@1PF	Cfd.@1P	Acc.@3P	Cf.@3PS	Acc.@1O	Cf.@1OS	Cf.@1OF	Cfd.@1O	Acc.@3O	Cf.@3OS	Acc.@1T	Cf.@1TS	Cf.@1TF	Cfd.@1T	Acc.@3T	Cf.@3TS
Qwen3-8B	0.361	0.853	0.847	0.005	0.624	0.735	0.476	0.848	0.851	-0.003	0.678	0.776	0.283	0.85	0.849	0.001	0.494	0.731
Llama3	0.328	0.662	0.591	0.071	0.567	0.485	0.512	0.638	0.59	0.047	0.683	0.535	0.249	0.678	0.594	0.084	0.456	0.48
gpt-4o	0.419	0.847	0.813	0.034	0.664	0.727	0.55	0.836	0.818	0.018	0.713	0.768	0.319	0.848	0.818	0.03	0.528	0.717
Gemini	0.328	0.825	0.772	0.053	0.618	0.683	0.465	0.805	0.776	0.029	0.684	0.717	0.223	0.833	0.777	0.056	0.469	0.668
Claude	0.423	0.694	0.63	0.063	0.71	0.495	0.538	0.674	0.637	0.037	0.737	0.549	0.324	0.701	0.636	0.065	0.559	0.49
DeepSeek-R1	0.4	0.673	0.569	0.104	0.66	0.504	0.528	0.642	0.575	0.068	0.717	0.538	0.302	0.687	0.577	0.109	0.528	0.495
Random	0.065	0.9	0.9	0	0.178	0.519	0.194	0.9	0.9	0	0.296	0.711	0.011	0.9	0.9	0	0.031	0.491

Table 9: Social interaction policy results. Metrics include accuracy (Acc.@n) for top-n results, confidence (Cf.@n), and confidence discrimination (Cfd.@n). Evaluations distinguish between predicate-only (@nP), object-only(@nO), and complete intention (@nT) assessments. Only success cases (@nxS) and only failure cases (@nxF) are listed when computing confidence.

model	Acc.@1P	Cf.@1PS	Cf.@1PF	Cfd.@1P	Acc.@3P	Cf.@3PS	Acc.@1O	Cf.@1OS	Cf.@1OF	Cfd.@1O	Acc.@3O	Cf.@3OS	Acc.@1T	Cf.@1TS	Cf.@1TF	Cfd.@1T	Acc.@3T	Cf.@3TS
Qwen3-8B	0.21	0.877	0.868	0.009	0.505	0.737	0.417	0.877	0.865	0.012	0.537	0.831	0.111	0.891	0.868	0.023	0.247	0.753
Llama3	0.185	0.631	0.638	-0.007	0.431	0.427	0.421	0.66	0.619	0.041	0.556	0.57	0.079	0.643	0.636	0.007	0.186	0.437
gpt-4o	0.213	0.864	0.854	0.01	0.452	0.716	0.448	0.862	0.851	0.011	0.577	0.801	0.115	0.878	0.853	0.025	0.24	0.71
Gemini	0.256	0.844	0.828	0.016	0.502	0.693	0.428	0.84	0.826	0.015	0.554	0.779	0.131	0.853	0.829	0.024	0.26	0.676
Claude	0.277	0.647	0.626	0.02	0.574	0.411	0.433	0.657	0.613	0.044	0.55	0.564	0.146	0.678	0.624	0.054	0.286	0.437
DeepSeek-R1	0.237	0.705	0.682	0.023	0.541	0.433	0.446	0.718	0.663	0.055	0.548	0.63	0.133	0.738	0.679	0.059	0.273	0.457
Random	0.036	0.9	0.9	0	0.124	0.469	0.316	0.9	0.9	0	0.536	0.671	0.012	0.9	0.9	0	0.041	0.457