Implicit Theory of Mind in LLM Agent Decision-Making

Anonymous Authors[1]

Abstract

Theory of Mind (ToM) is often claimed to emerge in large language models, yet it remains unclear whether LLM-based agents implicitly rely on ToM during concrete decision-making rather than explicit verbal reasoning. We study this question in a multi-agent game environment designed to elicit strategic interaction and goal inference. We introduce a turn-based multi-player trade-and-craft game with publicly observable inventories and private crafting goals, where players trade items and synthesize new ones according to Minecraft-inspired rules until any target item is crafted. To probe agents' mental state reasoning, we augment decision processes with explicit Theory of Mind structures of varying orders: reporting their own estimation and the estimation of the opponent's estimation, etc. These reported beliefs are evaluated against opponents' actual beliefs, allowing us to assess the accuracy of inferred psychological values alongside behavioral outcomes under default settings. Across GPT-4o, o3, o4-mini, and GPT-5, we find that agents' behavior and belief expressions are systematically influenced by explicit ToM scaffolding. Notably, increasing ToM order yields two qualitatively opposite trends across models, revealing heterogeneous effects of higher-order mental state reasoning in LLM agents

1. Introduction

Human intelligence is marked by its strength in reasoning, planning, and social cognition. Recent advances show that large language models (LLMs) have begun to approach, and in some cases surpass, human-level performance in these domains when evaluated separately. For multi-step reasoning and planning, benchmarks in mathematics (Cobbe et al., 2021; Sun et al., 2025) and interactive task-planning (Xie et al., 2024; Zhou et al., 2023) are widely used, while general techniques (Wei et al., 2022; Yao et al., 2023a;b), and special methods (Wang et al., 2023; Han et al., 2024), have proven effective in enhancing performance. In terms of social intelligence, (Strachan et al., 2024) reports that GPT-4 exceeds human performance on a variety of Theory of Mind (ToM) tasks, and (Street et al., 2024) provides evidence that LLMs can master higher-order ToM tasks in human-level performance.

Despite these promising results, limitations remain. Real-world applications rarely demand a single, isolated ability; instead, they require a dynamic combination of reasoning, planning, and social intelligence. For instance, (Wang et al., 2024) shows that models excelling in individual subtasks of ToM may underperform when required to solve the full integrated task in logical / geometric contexts. This suggests that ToM competence measured in direct question-answering settings may be insufficient: what ultimately matters is whether an agent's ToM inferences are *actually used to guide decisions*.

This motivates our core question: in a complex social-communicative environment, do LLM agents leverage ToM to make better proposals and accept/reject decisions, and if so, *how* is ToM incorporated into their decision policies? Answering this requires a suitable interactive setting in which ToM-relevant beliefs and actions co-evolve over time.

To better capture these complexities, recent work has turned to richer evaluation environments such as Diplomacy (Bakhtin et al., 2023), MineDojo (Fan et al., 2022), CivRealm (Qi et al., 2024), and MSCoRe (Lei et al., 2025). However, these settings face trade-offs: some are overly simplified with static cooperation or competition, while others (e.g., CivRealm (Qi et al., 2024)) are so complex that the behavioral signals of LLMs become too unstructured to reliably extract and evaluate. Indeed, researchers have noted a persistent gap: there is "a lack of multi-agent benchmarks for open-world environments" (Allen et al., 2024) that would allow diverse, realistic social interactions to unfold.

With the purpose of balancing the trade-off between **complexity** and **diversity** of LLM-Agent's evaluation, we intro-

[1]Anonymous Institution, Anonymous City, Anonymous Region, Anonymous Country.**AUTHORERR: Missing \icmlcorrespondingauthor.**

Preliminary work. Under review by the International Conference on Machine Learning (ICML). Do not distribute.

duce *TradeCraft*, a new multi-agent benchmark environment designed to probe high-order Theory of Mind, social reasoning, and strategic planning in both AI and human agents. Unlike existing platforms, *TradeCraft* offers an open-ended social sandbox where heterogeneous agents must negotiate, trade, and craft to pursue their goals. At its core is a general-purpose compositional crafting system, inspired by open-world games (cf. Minecraft (Fan et al., 2022), Little Alchemy 2 (Brändle et al., 2023)), which supports complex dependency structures and long-horizon objectives.

A distinctive feature of *TradeCraft* is its rule variability: both goals and mechanics can be randomized or customized across matches. This prevents rote memorization, provides a direct measure of adaptability and learning efficiency and enables a full control of task complexity. Social interaction is equally central because no single agent can succeed alone, agents must plan strategically, cooperate or compete, and engage in trade-based exchanges. Trading naturally gives rise to rich behaviors such as negotiation, trust building, deception, and higher-order belief modeling, offering a principled testbed for social reasoning.

By supporting both AI and human participants, *TradeCraft* enables human-in-the-loop evaluation and comparative studies of social intelligence. Through its combination of cooperation, competition, crafting, economic exchange, and configurable scenarios, *TradeCraft* establishes a unified benchmark that fills a long-standing gap in the study of adaptive multi-agent intelligence, providing an excellent testbed for studying ToM in social-communicative settings.

In summary, our contributions are as follows.

(1) **A lightweight social game for probing ToM in decisions.** We propose *TradeCraft*, a lightweight multi-agent trade-and-craft game that combines cooperation and competition in a controlled, interpretable setting. It is designed to probe whether (and how) LLM agents leverage Theory of Mind in concrete decision-making rather than only in explicit verbal reasoning.

(2) **A quantitative ToM measurement protocol.** Building on *TradeCraft*, we introduce an evaluation methodology that elicits structured ToM traces (zero-/first-/second-order) as item-wise value functions, enabling direct quantitative analyses of belief accuracy and their relationship to downstream behavior.

(3) **Empirical findings on model differences and ToM effects.** Using OpenAI models in self-play, we find (i) a sharp shift in trading strategy patterns across model generations, and (ii) systematic effects of explicit ToM elicitation on both proposal behavior and accept/reject decisions, revealing that higher-order ToM can be incorporated into decision policies in qualitatively different ways across models.

2. Related Work

2.1. Theory of Mind and Strategic Social Reasoning

Theory of Mind (ToM)—the ability to infer others' beliefs, intentions, and desires—is a cornerstone of social intelligence. Classic models formalize ToM via Bayesian inference (Baker et al., 2011) or recursive belief modeling in I-POMDPs (Gmytrasiewicz & Doshi, 2005). More recent approaches like ToMnet (Rabinowitz et al., 2018) use meta-learning to predict agent behavior from limited observations. These methods demonstrate success in simple gridworlds but remain limited in generalizability. In multi-agent reinforcement learning, ToM-inspired models have improved coordination and competition via policy modeling (He et al., 2016; Raileanu et al., 2018). SymmToM (Sclar et al., 2022) further explores this in a communication-rich environment, yet still falls short of oracle performance. As agents acquire ToM, complex behaviors such as deception and strategic communication emerge. TradeCraft builds on this foundation by embedding high-order ToM reasoning into a compositional, dynamic benchmark that explicitly tests belief modeling, negotiation, and strategic planning in cooperative-competitive contexts.

2.2. Benchmarks for Social Intelligence and Mixed-Motive Interaction

Benchmarks like Hanabi (Bard et al., 2020), Diplomacy (Bakhtin et al., 2022), and Melting Pot (Leibo et al., 2021) evaluate agents' abilities in belief inference, negotiation, and social generalization. Others, such as Overcooked-AI (Carroll et al., 2019), highlight challenges in human-AI collaboration and ad-hoc teamwork. Hide-and-Seek (Baker et al., 2019) reveals emergent strategies from self-play in competitive settings. However, most existing environments target isolated facets (e.g., implicit communication, collaboration) and assume static rules. In contrast, TradeCraft introduces a unified, grounded environment where agents must engage in long-horizon planning, resource management, and flexible social strategies under dynamic rule changes. Its hybrid-motive design (collaboration + bartering + competition) supports the emergence of context-sensitive cooperation and deception, offering a more comprehensive testbed for evaluating strategic social intelligence.

2.3. LLMs for Multi-Agent Reasoning and Human-AI Interaction

Large language models (LLMs) have shown promise in social reasoning tasks. Generative Agents (Park et al., 2023) simulate social behaviors through LLMs enhanced with memory and reflection. ProAgent (Zhang et al., 2024) and Hypothetical Minds (Wu et al., 2024) integrate modular ToM reasoning with LLM planners, achieving strong perfor-

mance on Melting Pot tasks. Meanwhile, Cicero (Bakhtin et al., 2022) combines LLM dialogue with planning to play Diplomacy at human level. Despite these advances, current evaluations focus on simulated text environments or fixed games. TradeCraft offers a grounded alternative: it evaluates LLMs in embodied multi-agent scenarios with real-time interaction, compositional objectives, and rule variability. Crucially, it supports human-AI interaction, enabling research into ad-hoc collaboration and ToM reasoning against humans—an underexplored frontier in LLM-based multi-agent learning.

3. Method

3.1. The *TradeCraft* Environment

3.1.1. The Game Design

TradeCraft is a turn-based multiplayer online game designed as a testbed for long-term strategic social reasoning and planning, supporting both human and LLM agents. In each game session, players maintain a collection of items through bartering with other participants and crafting based on predefined formulas. Example crafting formulas are illustrated in Figure. 1. The system currently incorporates item sets and rule systems from Minecraft Java-v1.20 and LittleAlchemy2, while allowing straightforward modification, replacement, or extension of game rules and items (see Section A). Each game involves two or more players, each possessing a *hand* of items (with multiplicity) and being assigned a private *target* item to craft. While all players' hands are fully visible to all participants, each player's target item remains private. The objective for each player is to be the first to craft their designated target item. Since initial hands are typically insufficient for direct target item crafting, players **must** acquire necessary components through trading with other players.

The game runs in turns, each turn consists of two phases: the trade phase and the craft phase, see Figure. 1. In the **trade phase**, one player (called the proposer of this turn) chooses another player and makes a proposal for trading, together with a text message; the chosen player decides to accept or to reject the proposal. If a proposal is accepted, then the hands of the two trading players change accordingly. If rejected, the proposal will be invisible to any other players. After one trial of the one-on-one trading, the trade phase ends. The proposer rotates to the next player at the end of the turn. The players act as the proposer in a fixed order. The **craft phase** follows the trade phase, where each player starts to craft items at the same time. It is possible to craft several times in a single craft phase until they choose to finish crafting. The hand changes will not be revealed to others until all players are done with crafts, and during the craft phase, items can be used in a number of rational numbers (fractions), and at the end of the craft phase, all non-integer amounts are rounded down.

The Server. The server hosts all the game state and dynamics, manages the login users and game. multiple games with different player amounts or rules can be hosted on a single server. Server is written in Python with package `flask`, MongoDB database are used to save game state and logs.

Web-GUI The web-based GUI integrates all game functions together with built-in assistance and crafting support, human users can reach through web-browsers (see Appendix Figure. 7.)

Lang-API The language-based API mirrors the functionality of the Web-GUI in text modal, designed to comply with `gymnasium` for agent integration. Observations are provided in language, and actions are executed via `langchain` tools. There is a set of standard tools provided together with the environment which might help both AI and human players craft their final target.

In gymnasium, a "observation-action" loop is maintained. In each cycle, an agent reads observations and chooses an action with arguments. In the API, **observations** are provided in text format. Conducted by game-dynamics module, language interpreter module translates system messages into text generates the observations, both modules are highly extensible and customizable. Observations contain all the facts that web-GUI contains. **Actions** are accepting an argument dictionary. The end-of-phase action affects the game state, containing *submit proposal*, *submit decision*, *finish crafting*, etc., while within-phase tools are for querying information, such as item info, possible crafts from hands, etc.

3.1.2. Initial Game States

A *task instance* specifies the initial hands (inventories) and private targets of all players. Since all target items must be craftable from the union of all players' initial hands, task instances that are not carefully designed tend to be invalid or trivial. For the Minecraft ruleset, we provide 40 predefined task instances for the 1-vs-1 game mode, covering a range of difficulty levels (Figure 2). The instances can be easily maintained by editing JSON files, and new game modes can be introduced by adding corresponding directories and adjusting configuration settings (see Section A). The difficulty of a task instance is assessed along two dimensions: the length of the crafting chain and the minimum number of trading steps required. In designing these instances, we follow the principle that the union of all players' initial items must suffice to craft each player's target individually; however, it is not guaranteed that all targets can be crafted simultaneously (for instance, the total pool may contain only 3 stones, while two players each require 2 stones). To

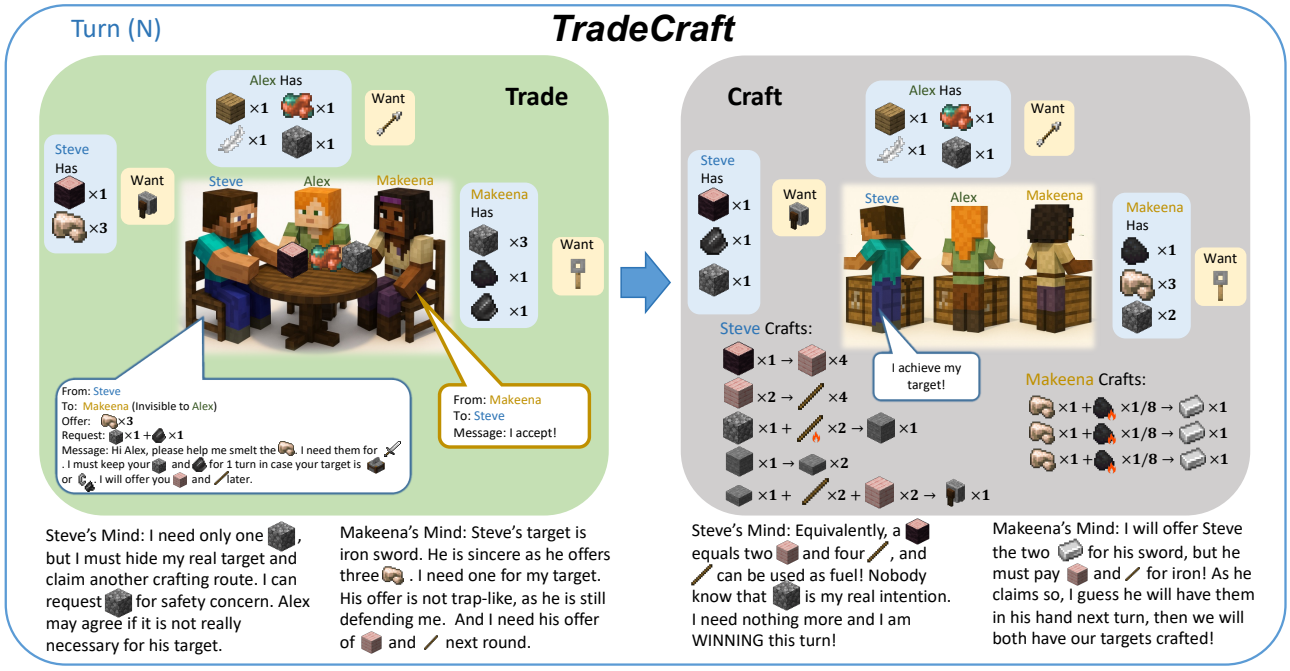

Figure 1. TradeCraft involves social interaction, deep reasoning, long-term planning, and fine-grained control. Players engage in social negotiation and trading, and then synthesize target items through long-term planning and precise control. Success requires higher-order Theory of Mind: reasoning about others' intentions, inventory states, and synthesis strategies in goal-directed contexts.

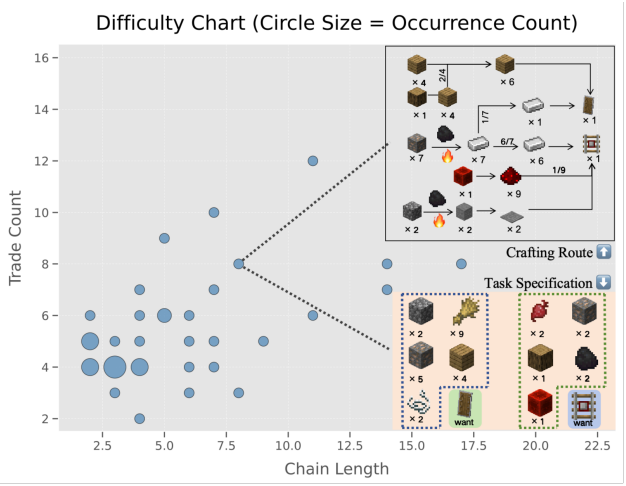

Figure 2. Supported inventories.

promote strategic competition and planning, initial hands are not set to the exact amounts needed to craft all targets. Redundant items can introduce alternative crafting routes or serve as distractors for deception. In later discussions, we use the first half of the full inventories in our experiments.

3.2. Game Implementation and Interfaces

The *TradeCraft* implementation consists of a server and two types of user-interfaces.

3.3. Defining Elicited ToM Reports in *TradeCraft*

We operationalize ToM in *TradeCraft* as *value modeling* over items, and we focus on ToM up to **second order**. In our setting, an agent's ToM is instantiated by a set of item-wise utility scores that reflect (i) its own goal-directed needs, (ii) its beliefs about the opponent's needs, and (iii) its beliefs about how the opponent perceives its needs.

ToM orders. We define three ToM orders: **zero-order ToM (self)** captures the agent's beliefs about item utility for *its own* secret target; **first-order ToM (other)** captures the agent's beliefs about item utility for the *opponent's* secret target; and **second-order ToM (other about self)** captures the agent's beliefs about how the opponent values the agent's items (i.e., the opponent's beliefs about the agent's target).

Item-wise value function. At the beginning of each turn (in the proposal stage), regardless of whether the agent is the proposer or the decision maker, we require it to assign an item-wise utility score on a 0–10 scale, where 0 indicates *useless* and 10 indicates *critical for the secret target*(Figure 3 shows a demo) . We define these values only over items currently present in all players' publicly observable inventories (hands). Let V_0, V_1, and V_2 denote the zero-/first-/second-order ToM value dictionaries, respectively.

Concretely, the agent outputs:

Figure 3. Pipeline of the *TradeCraft* game (left) and our elicited ToM report design (right). Each turn consists of a trade phase (proposal + accept/reject) followed by a simultaneous craft phase. To operationalize ToM in decision making, we elicit structured *ToM reports* at each proposal/decision stage: the agent reports item-wise utility scores on a 0–10 scale over all players' publicly observable inventories (hands). These reports form V_0 (zero-order: the agent's goal-conditioned values), V_1 (first-order: the agent's estimate of the opponent's V_0), and V_2 (second-order: the agent's estimate of the opponent's estimate of the agent's V_0). Importantly, we do not instruct the agent how to use these reports; they serve as a measurement interface for analyzing whether and how ToM information is implicitly incorporated into trade proposals and accept/reject decisions.

- V_0: a dictionary mapping each item in all players' publicly observable inventories (hands) to its estimated utility for achieving *its* target;
- V_1: a dictionary mapping each item in all players' publicly observable inventories (hands) to its estimated utility for achieving the *opponent's* target (based on the agent's current hypothesis of that target);
- V_2: a dictionary mapping each item in all players' publicly observable inventories (hands) to the utility that the agent believes the *opponent* believes the agent assigns to it (i.e., the opponent's estimate of the agent's V_0).

We treat V_0 as the observable trace of zero-order ToM, V_1 as the agent's estimate of the opponent's V_0(first-order ToM), and V_2 as the agent's estimate of the opponent's estimate of the agent's V_0(second-order ToM).

During gameplay, we elicit ToM reports by prompting the agent to output V_0–V_2 at each proposal/decision stage. This design enables quantitative ToM evaluation by comparing these elicited reports against ground-truth information available in the logs (e.g., the opponent's subsequent revealed valuations or targets in controlled settings). Moreover, the elicited ToM reports provide features for analyzing downstream decisions (trades), allowing us to study how higher-order belief modeling shapes strategic behavior.

4. Experiment

In this section, we describe the evaluated agents and our experimental protocol. Each agent maintains a dialogue-context memory across turns and can invoke two environment tools: **item_info**, which retrieves item metadata and required ingredients, and **possible_recipes_from_hands**, which enumerates all craftable items given the current inventory.

We evaluate four models using self-play (e.g., GPT-4o vs. GPT-4o; o3 vs. o3) on the first half of all our task instances(20/40). To isolate the effect of eliciting different ToM report orders, we fix the opponent to be a *same-model* baseline agent that only produces zero-order ToM reports (i.e., it outputs V_0 only).

For each model, we run three ToM configurations against the zero-order baseline: V_0 vs. V_0, (V_0, V_1) vs. V_0, and (V_0, V_1, V_2) vs. V_0. For clarity, we refer to these agents as the V_0 **group** (outputs V_0 only), the V_1 **group** (outputs V_0 and V_1, i.e., zero- and first-order ToM), and the V_2 **group** (outputs V_0, V_1, and V_2, i.e., up to second-order ToM). Each configuration is evaluated once on each of the 20 task instances. **Importantly, our prompts do not provide any additional instruction on *how* these elicited ToM reports should be used for proposing trades or making decisions; thus, any reliance on ToM in the agent's policy is *implicit* and must arise spontaneously (i.e., without-instruction).** We log the per-turn ToM outputs, the trade proposals, and the decision outcomes (accept/reject) for subsequent behavioral analyses.

Our design enables quantitative evaluation of ToM accuracy. As an example metric, we compute the KL divergence between a model's predicted opponent valuation V_1 and the opponent's reported V_0. As shown in Figure 4, we compute and plot the KL divergence between an agent's first-order

ToM estimate of the opponent (i.e., its predicted valuation V_1) and the opponent's reported zero-order valuation (V_0). Lower KL indicates that the agent's belief about the opponent's item values is more consistent with the opponent's own self-assessment.

Overall, weaker models (e.g., GPT-4o) exhibit substantially larger KL divergence, suggesting systematic difficulty in forming accurate beliefs about the opponent's goals and item utilities. In contrast, stronger models achieve lower KL overall, with GPT-5 performing best; we also observe a decreasing-KL trend over turns, indicating that models can refine their opponent belief estimates from interaction history. The V_2 **group** tends to show slightly improved first-order accuracy compared to the V_1 **group**, suggesting that eliciting second-order ToM reports can indirectly regularize or sharpen first-order belief formation.

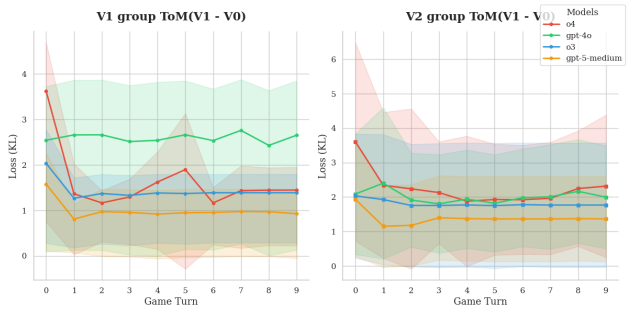

Figure 4. KL divergence of between players' first-order ToM and their opponents' zero-order ToM

4.1. The shift of value distribution in proposal

In this section, we study how eliciting ToM reports affects proposal behavior. For each of the four models and each ToM configuration (zero-/first-/second-order), we collect the set of trade proposals and compute the *self-assessed* values of the proposed items using the agent's reported V_0.

Specifically, for each proposal we compute the total V_0 value of the requested items and the total V_0 value of the offered items, and visualize them in a scatter plot with coordinates $(\text{request},\ \text{offer})$. We then compute the Pearson correlation r between request and offer values to quantify whether the agent tends to balance what it asks for against what it gives.

To further characterize the trading pattern, we fit a linear regression line on the scatter plot. The fitted slope m can be interpreted as an exchange-rate proxy (offer/request): $m \approx 1$ indicates roughly fair trades, $m < 1$ indicates more aggresive amd selfish offers, and $m > 1$ indicates more concessive proposal.

Figure 5 summarizes the fitted slopes across models and ToM orders. We observe two qualitatively different trends. For earlier models (e.g., GPT-4o and o3), increasing ToM order is associated with decreasing correlation r and a smaller slope m, suggesting weaker coupling between offered and requested values and a trend toward more greedy behavior. In contrast, starting from o4-mini, we observe a sharp shift: o4 and GPT-5 (reasoning effort = Medium) show higher offer–request correlation and a larger slope as ToM order increases, indicating that eliciting higher-order ToM reports is more consistently translated into fairer, value-aligned proposals.

4.2. Decision-ToM Utility Fitting

We model the decision maker's accept/reject behavior by fitting a simple utility-based classifier to the observed decisions. The key idea is to treat the decision maker's own ToM-valued item utilities (at chosen ToM orders) as features of a proposal.

Consider a single proposal presented to a decision maker. Let $\mathcal{I}^{\text{recv}}$ and $\mathcal{I}^{\text{give}}$ denote the multisets of items the decision maker would receive and give. For each item i in the proposal, we take the decision maker's reported ToM value at the corresponding orders as its utility. We denote these per-item utilities by $v_i^{(k)}$ for $k \in \{0, 1, 2\}$, corresponding to the decision maker's explicitly reported ToM order.

We then aggregate item-wise utilities into two scalars for each order:

$$G_k = \sum_{i \in \mathcal{I}^{\text{recv}}} v_i^{(k)}, \quad L_k = \sum_{i \in \mathcal{I}^{\text{give}}} v_i^{(k)}, \quad k \in \{0, 1, 2\}.$$

Intuitively, G_k measures the perceived gain and L_k measures the perceived loss under ToM order k. We define a scalar utility for a proposal and progressively incorporate higher-order ToM terms:

$$U^{(0)} = G_0 - \rho\, L_0, \tag{1}$$

$$U^{(0,1)} = U^{(0)} + w_1 \left(G_1 - \gamma\, L_1\right), \tag{2}$$

$$U^{(0,1,2)} = U^{(0,1)} + w_2 \left(G_2 - \kappa\, L_2\right). \tag{3}$$

Logistic regression for accept/reject. We fit a logistic regression model to predict whether the decision maker accepts the proposal (y=1) or rejects it (y=0) using a single feature U:

$$P(y{=}1 \mid U) = \sigma(\beta\, \tilde{U} + b),$$

where $\tilde{U}$ is the standardized utility and σ is the sigmoid function.

We using LBFGS optimizer together with a grid search method to determine the utility parameters (e.g., $\rho, \gamma, \kappa, w, w_1, w_2$), validate the regression model with episode-wise three-fold cross-validation and report macro-F1. Table 1 summarizes the macro-F1 scores under zero-/first-/second-order settings.

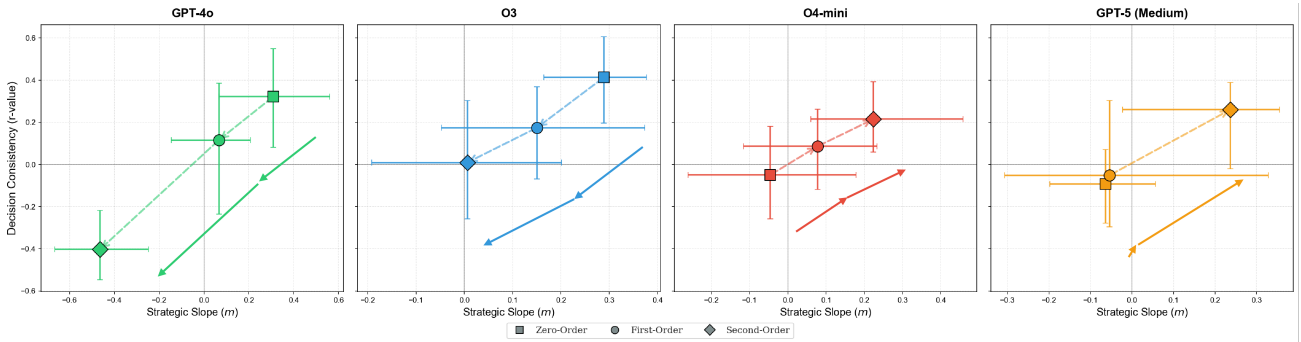

Figure 5. Strategic consistency map (m-r space) with 90% bootstrap CI (B=2000). Each point is a (model, order) regression summary; arrows connect orders. CIs are computed by episode(question)-level bootstrap (folded by Episode), i.e., resampling episode rather than individual proposals

Table 1. Macro-F1 of our utility-fitting model across ToM orders. Values are mean ± standard deviation over episode-wise 3-fold cross-validation splits.

Model	Zero-order	First-order	Second-order
GPT-4o	0.77 ± 0.28	0.94 ± 0.09	1.00 ± 0.00
o3	0.80 ± 0.19	1.00 ± 0.00	1.00 ± 0.00
o4	0.74 ± 0.18	0.80 ± 0.09	0.83 ± 0.04
GPT-5 (medium)	0.74 ± 0.24	0.92 ± 0.06	0.85 ± 0.11

Table 2. Semantics of gain/loss features used in utility fitting. All values are from the *decision maker*'s ToM reports at the corresponding order.

Feature	Meaning
G_0	Value (to the decision maker) of the items the decision maker would *receive* if the proposal is accepted.
L_0	Value (to the decision maker) of the items the decision maker would *give* if the proposal is accepted.
G_1	Value (to the decision maker's estimate of the opponent) of the items the decision maker would *receive*.
L_1	Value (to the decision maker's estimate of the opponent) of the items the decision maker would *give*.
G_2	Value (to the decision maker's estimate of the opponent's estimate of the decision maker) of the items the decision maker would *receive*.
L_2	Value (to the decision maker's estimate of the opponent's estimate of the decision maker) of the items the decision maker would *give*.

Based on the logistic-regression fitting results, the utility model built from ToM-derived values provides a strong approximation of LLM decision-making in *TradeCraft*. In particular, the macro-F1 scores in Table 1 are consistently high (all above 0.74), and the accept/reject behavior becomes especially predictable once agents are prompted to output beyond zero-order ToM.

Since this simple model fits well, analyzing its learned coefficients is informative. Figure6 visualizes the fitted coefficients for agents prompted to output different ToM orders, e report the effective weights of each gain/loss component.

As shown in Figure 6, positive coefficients indicate preference for the corresponding terms, while negative coefficients indicate aversion. This visualization enables a direct comparison of how different models weigh gains and losses under different ToM orders.

We observe that when agents only expose zero-order ToM, decisions are largely driven by gains: the weight on G_0 is typically dominant. At the same time, as models become more capable (from top to bottom in the figure), they exhibit increasingly strong loss aversion, assigning more negative weight to L_0.

When agents are prompted to output first-order ToM, G_0 remains a primary driver and most models still show loss aversion. However, models differ in how they leverage first-order signals. GPT-4o and o4-mini assign positive weights to the first-order terms and typically place larger magnitude on G_1 than on L_1, indicating a tendency to prioritize obtaining items that are predicted to be valuable to the opponent. In contrast, o3 and GPT-5 exhibit a more cooperative pattern: proposals that improve the opponent's predicted utility can become easier to accept, and large opponent-relevant gains may be penalized less aggressively.

When agents are prompted to output second-order ToM, we observe an additional, qualitatively different signal in the fitted coefficients. Across all models, G_2 is rewarded and L_2 is penalized, suggesting a consistent preference for proposals that make the agent appear to be *gaining* in the opponent's eyes, and an aversion to proposals that (the agent believes) make it appear to be *losing*.

Interestingly, weaker models such as GPT-4o can overweight this "image" term: the coefficient on G_2 becomes unusually large, and the model may even assign a negative

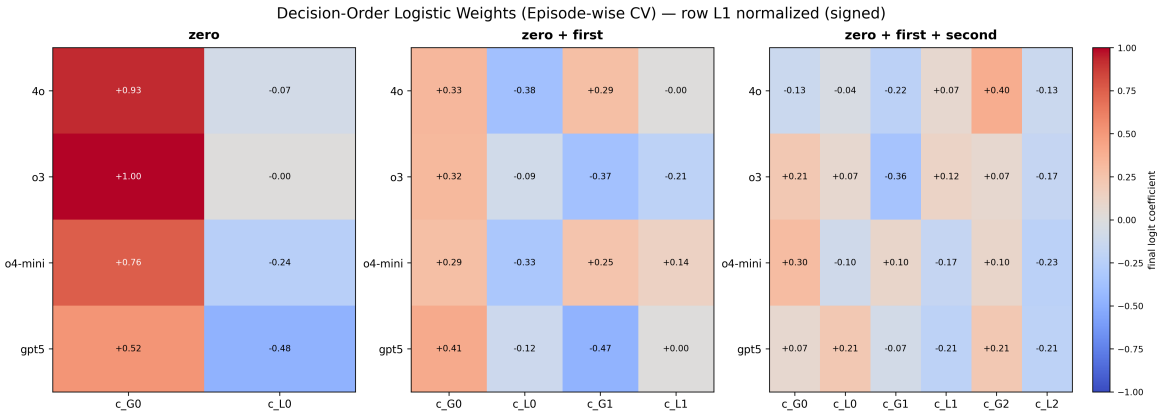

Figure 6. Normalized coefficients c of logistic-regression models.

weight to the self-gain term G_0. This pattern indicates that the model's accept/reject behavior can be driven more by perceived reputation management than by its own directly-valued utility.

We also find that introducing second-order ToM can change how some models treat first-order loss terms. Compared to the first-order setting, o4 and GPT-5 become less willing to accept proposals in which the opponent obtains items that are predicted to be valuable to them (i.e., a more negative weight on L_1). One interpretation is that second-order ToM introduces an explicit strategic layer: the agent not only reasons about what the opponent values, but also considers how the trade reshapes the opponent's beliefs and future bargaining position. As a result, these models become more cautious about empowering the opponent with high-value resources, even when the immediate trade might appear acceptable under self-valued utility alone.

5. Discussion

Our results suggest that elicited ToM reports can serve as a useful "measurement lens" on LLM-agent behavior: the same base policy can exhibit systematically different trading patterns depending on whether and how ToM reports are made salient. In proposal behavior, we observe that models differ qualitatively in how they map ToM signals to exchange behavior, with a sharp strategy transition across model generations. In decision making, the utility-fitting analysis shows that accept/reject choices can be well-approximated by a low-dimensional utility model constructed from ToM-valued gain/loss terms, and that higher-order ToM features can substantially increase predictability for several models.

These findings highlight two broader implications. First, **"having" ToM (being able to report plausible beliefs) and "using" ToM (deploying those beliefs to guide action) are separable**: earlier models may express ToM-like values yet fail to translate them into coherent trading strategies, while stronger models appear to integrate ToM signals more consistently. Second, higher-order ToM can introduce a strategic layer beyond immediate self-utility, potentially reflecting sensitivity to reputation, bargaining position, and the future informational consequences of trades.

6. Limitation and Further Steps

Our study focuses on a lightweight game and on ToM up to second order. We use prompt-based elicitation rather than fully learned belief modules, and we primarily evaluate self-play, which may not capture all dynamics present in human–AI interaction. Future work can expand task diversity (more players, alternative rule sets, richer communication), stress-test generalization under distribution shifts, and connect ToM traces to longer-horizon outcomes such as winning probability and exploitability.

7. Conclusion

We introduced *TradeCraft*, a lightweight social trade-and-craft environment for probing whether and how LLM agents leverage ToM in concrete decision-making. By eliciting structured zero-/first-/second-order ToM reports as item-wise value functions, we developed a quantitative methodology to evaluate belief accuracy and to relate beliefs to proposal and decision behavior. Across OpenAI models, we observe a sharp strategy shift across generations and systematic effects of ToM elicitation on both trading proposals and accept/reject decisions, providing evidence that the role of ToM in decision-making is model-dependent and can change qualitatively with scaling and training.

References

Allen, K., Brändle, F., Botvinick, M., Fan, J. E., Gershman, S. J., Gopnik, A., Griffiths, T. L., Hartshorne, J. K.,

Hauser, T. U., Ho, M. K., et al. Using games to understand the mind. *Nature human behaviour*, 8(6):1035–1043, 2024.

Baker, B., Kanitscheider, I., Markov, T., Wu, Y., Powell, G., McGrew, B., and Mordatch, I. Emergent tool use from multi-agent autocurricula. *arXiv preprint arXiv:1909.07528*, 2019.

Baker, C., Saxe, R., and Tenenbaum, J. B. Bayesian theory of mind: Modeling joint belief–desire attribution. In *Proceedings of the Annual Meeting of the Cognitive Science Society*, volume 33, pp. 2469–2474, 2011.

Bakhtin, A., Brown, N., Chien, S., Chu, Y., Kiela, D., Mourad, S., Schick, T., Szlam, A., Dinan, E., Batra, D., et al. Human-level play in the game of diplomacy by combining language models with strategic reasoning. *Science*, 378(6624):1067–1074, 2022.

Bakhtin, A., Wu, D. J., Lerer, A., Gray, J., Jacob, A. P., Farina, G., Miller, A. H., and Brown, N. Mastering the game of no-press diplomacy via human-regularized reinforcement learning and planning. In *The Eleventh International Conference on Learning Representations*, 2023. URL https://openreview.net/forum?id=F61FwJTZhb.

Bard, N., Foerster, J. N., Chandar, S., Burch, N., Lanctot, M., Song, H. F., Parisotto, E., Dumoulin, V., Mourad, S., Larson, B., et al. The hanabi challenge: A new frontier for ai research. *Artificial Intelligence*, 280:103216, 2020.

Brändle, F., Stocks, L. J., Tenenbaum, J. B., Gershman, S. J., and Schulz, E. Empowerment contributes to exploration behaviour in a creative video game. *Nature Human Behaviour*, 7(9):1481–1489, 2023.

Carroll, M., Shah, R., Ho, M. K., Griffiths, T. L., Abbeel, P., and Dragan, A. D. On the utility of learning about humans for human-ai coordination. In *Advances in Neural Information Processing Systems*, volume 32, 2019.

Cobbe, K., Kosaraju, V., Bavarian, M., Chen, M., Jun, H., Kaiser, L., Plappert, M., Tworek, J., Hilton, J., Nakano, R., et al. Training verifiers to solve math word problems. *arXiv preprint arXiv:2110.14168*, 2021.

Fan, L., Wang, G., Jiang, Y., Mandlekar, A., Yang, Y., Zhu, H., Tang, A., Huang, D.-A., Zhu, Y., and Anandkumar, A. Minedojo: Building open-ended embodied agents with internet-scale knowledge. In *Thirty-sixth Conference on Neural Information Processing Systems Datasets and Benchmarks Track*, 2022. URL https://openreview.net/forum?id=rc8o_j8I8PX.

Gmytrasiewicz, P. J. and Doshi, P. A framework for sequential planning in multi-agent settings. *Journal of Artificial Intelligence Research*, 24:49–79, 2005.

Han, M., Zhu, Y., Zhu, S.-C., Wu, Y. N., and Zhu, Y. Interpret: Interactive predicate learning from language feedback for generalizable task planning. In *Robotics: Science and Systems*, 2024. URL https://doi.org/10.15607/RSS.2024.XX.034.

He, H., Boyd-Graber, J., Daumé III, H., and Kwok, K. Opponent modeling in deep reinforcement learning. In *International Conference on Machine Learning*, pp. 1804–1813. PMLR, 2016.

Lei, Y., Xie, H., Zhao, J., Liu, S., and Song, X. Mscore: A benchmark for multi-stage collaborative reasoning in llm agents, 2025. URL https://arxiv.org/abs/2509.17628.

Leibo, J. Z., Hughes, C., Lanctot, M., Lespiau, J.-B., Zambaldi, V., Lockhart, E., Clune, J., and Graepel, T. Scalable evaluation of multi-agent reinforcement learning with melting pot. In *International Conference on Machine Learning*, pp. 6187–6199. PMLR, 2021.

Park, J. S., O'Brien, J., Cai, C. J., Morris, M. R., Liang, P., and Bernstein, M. S. Generative agents: Interactive simulacra of human behavior. *arXiv preprint arXiv:2304.03442*, 2023.

Qi, S., Chen, S., Li, Y., Kong, X., Wang, J., Yang, B., Wong, P., Zhong, Y., Zhang, X., Zhang, Z., Liu, N., Wang, W., Yang, Y., and Zhu, S.-C. Civrealm: A learning and reasoning odyssey in civilization for decision-making agents. *CoRR*, abs/2401.10568, 2024. URL https://doi.org/10.48550/arXiv.2401.10568.

Rabinowitz, N. C., Perbet, F., Song, H. F., Zhang, C., Eslami, S. M. A., and Botvinick, M. Machine theory of mind. In *International Conference on Machine Learning*, pp. 4218–4227. PMLR, 2018.

Raileanu, R., Denton, E., Szlam, A., and Fergus, R. Modeling others using oneself in multi-agent reinforcement learning. In *International Conference on Machine Learning*, pp. 4257–4266. PMLR, 2018.

Sclar, N., Anson, M., Achlioptas, P., Hassidim, A., Halperin, T., Yahav, T., Korman, A., and Feder, A. Symmtom: A symmetric theory of mind benchmark for multiagent communication. In *Advances in Neural Information Processing Systems*, volume 35, pp. 28402–28415, 2022.

Strachan, J. W. A., Albergo, D., Borghini, G., Pansardi, O., Scaliti, E., Gupta, S., Saxena, K., Rufo, A., Panzeri, S., Manzi, G., Graziano, M. S. A., and Becchio, C. Testing theory of mind in large language models and humans. *Nature Human Behaviour*, 8(7):1285–1295, 07 2024. ISSN 2397-3374. doi: 10.1038/s41562-024-01882-z. URL https://doi.org/10.1038/s41562-024-01882-z.

Street, W., Siy, J., Keeling, G., Baranes, A., Barnett, B., McKibben, M., Kanyere, T., Lentz, A., Arcas, B., and Dunbar, R. Llms achieve adult human performance on higher-order theory of mind tasks. 05 2024. doi: 10.48550/arXiv.2405.18870.

Sun, H., Min, Y., Chen, Z., Zhao, W. X., Fang, L., Liu, Z., Wang, Z., and Wen, J.-R. Challenging the boundaries of reasoning: An olympiad-level math benchmark for large language models. *arXiv preprint arXiv:2503.21380*, 2025.

Wang, G., Xie, Y., Jiang, Y., Mandlekar, A., Xiao, C., Zhu, Y., Fan, L., and Anandkumar, A. Voyager: An open-ended embodied agent with large language models. *arXiv preprint arXiv: Arxiv-2305.16291*, 2023.

Wang, J., Zhang, C., Li, J., Ma, Y., Niu, L., Han, J., Peng, Y., Zhu, Y., and Fan, L. Evaluating and modeling social intelligence: A comparative study of human and ai capabilities. In *Proceedings of the Annual Meeting of the Cognitive Science Society*, volume 46, 2024. URL https://escholarship.org/uc/item/2j53v5nv.

Wei, J., Wang, X., Schuurmans, D., Bosma, M., ichter, b., Xia, F., Chi, E., Le, Q. V., and Zhou, D. Chain-of-thought prompting elicits reasoning in large language models. In Koyejo, S., Mohamed, S., Agarwal, A., Belgrave, D., Cho, K., and Oh, A. (eds.), *Advances in Neural Information Processing Systems*, volume 35, pp. 24824–24837. Curran Associates, Inc., 2022. URL https://proceedings.neurips.cc/paper_files/paper/2022/file/9d5609613524ecf4f15af0f7b31abca4-Paper-Conference.pdf.

Wu, J., Lee, A., Zhang, X., et al. Hypothetical minds: Augmenting llms with theory-of-mind reasoning for social agents. In *International Conference on Learning Representations (ICLR)*, 2024.

Xie, J., Zhang, K., Chen, J., Zhu, T., Lou, R., Tian, Y., Xiao, Y., and Su, Y. Travelplanner: a benchmark for real-world planning with language agents. In *Proceedings of the 41st International Conference on Machine Learning*, pp. 54590–54613, 2024.

Yao, S., Yu, D., Zhao, J., Shafran, I., Griffiths, T., Cao, Y., and Narasimhan, K. Tree of thoughts: Deliberate problem solving with large language models. In *Advances in Neural Information Processing Systems*, volume 36, pp. 11809–11822, 2023a.

Yao, S., Zhao, J., Yu, D., Du, N., Shafran, I., Narasimhan, K., and Cao, Y. React: Synergizing reasoning and acting in language models. In *International Conference on Learning Representations*, 2023b.

Zhang, C., Yang, K., Hu, S., Wang, Z., Li, G., Sun, Y., Zhang, C., Zhang, Z., Liu, A., Zhu, S.-C., et al. Proagent: building proactive cooperative agents with large language models. In *Proceedings of the AAAI Conference on Artificial Intelligence*, volume 38, pp. 17591–17599, 2024.

Zhou, S., Xu, F. F., Zhu, H., Zhou, X., Lo, R., Sridhar, A., Cheng, X., Bisk, Y., Fried, D., Alon, U., et al. Webarena: A realistic web environment for building autonomous agents. *arXiv preprint arXiv:2307.13854*, 2023.

A. TradeCraft Game

We construct a configurable multi-player environment in which agents collaborate or compete to achieve item synthesis objectives through trading and crafting. The environment is designed to support multiple synthesis rule systems, most notably those derived from Minecraft and Little Alchemy 2, enabling researchers to investigate agent behavior under varying levels of combinatorial complexity, structural constraints, and long-horizon planning demands. This multi-rule setting allows for a systematic analysis of agent capabilities. The Minecraft-inspired configuration employs grid-based crafting logic with explicit recipe tables and strict input requirements. Each crafting operation requires precise item combinations and, in many cases, auxiliary fuel resources (e.g., coal for smelting). This setup emphasizes local planning, resource management, and deterministic action validation.

In contrast, the Little Alchemy 2-based system features a significantly more permissive and exploratory synthesis mechanism. Items can be combined in various orders and through long synthesis chains, with minimal structural constraints. This configuration emphasizes long-horizon reasoning, abstraction, and adaptability to open-ended composition paths.

Beyond crafting, the environment incorporates a structured trading phase, wherein agents may exchange items based on their beliefs, needs, or inferred goals. This component enables the evaluation of social reasoning, such as goal inference, negotiation strategies, and basic forms of theory of mind. Agents must not only plan for item synthesis but also engage in cooperative behavior, anticipate their partner's intentions, and adapt their strategy accordingly.

The environment supports seamless switching between rule systems and allows for custom rule definitions, thereby functioning as a general platform for evaluating both synthesis-centric reasoning and socially situated decision-making in multi-agent scenarios.

Each turn consists of a sequence of structured phases, involving trade negotiation, decision-making, and item synthesis. The overall process is as follows:

Initialization. At the beginning of the game, each agent is assigned an initial inventory of items. These items are drawn from a predefined item pool governed by the selected rule system (e.g., Minecraft-style or Little Alchemy 2-style rules). Initialization occurs only once, before the first turn.

Proposal Phase. In each turn, one agent is designated as the proposer and enters the proposal phase. The proposer constructs a trade proposal consisting of:a set of items to offer, a set of items to request, and an optional message conveying intent or context.

Decision Phase. The target agent receives the proposal and evaluates it based on the content of the proposal message, the current items, and its goals. The agent makes a binary decision to either accept or reject the proposal. If accepted, the proposed trade is executed, and both agents' inventories are updated accordingly. If rejected, no exchange occurs.

Craft Phase: After the decision phase, both agents independently attempt to synthesize new items using their current inventories. Crafting actions are validated against the active rule system using the tools. Only combinations that satisfy the system-defined synthesis constraints are permitted. After all crafting operations are complete, the resulting item quantities are floored to the nearest integer.

Once the crafting phase concludes, the environment transitions to the next turn, and a new agent is selected to initiate the proposal phase. The game continues for a predefined number of turns or until specific task objectives are achieved.

A.1. Format of a Crafting Formula

We follow strictly the Minecraft-Java-1.20 crafting recipe settings. Common crafting recipeslooks like:

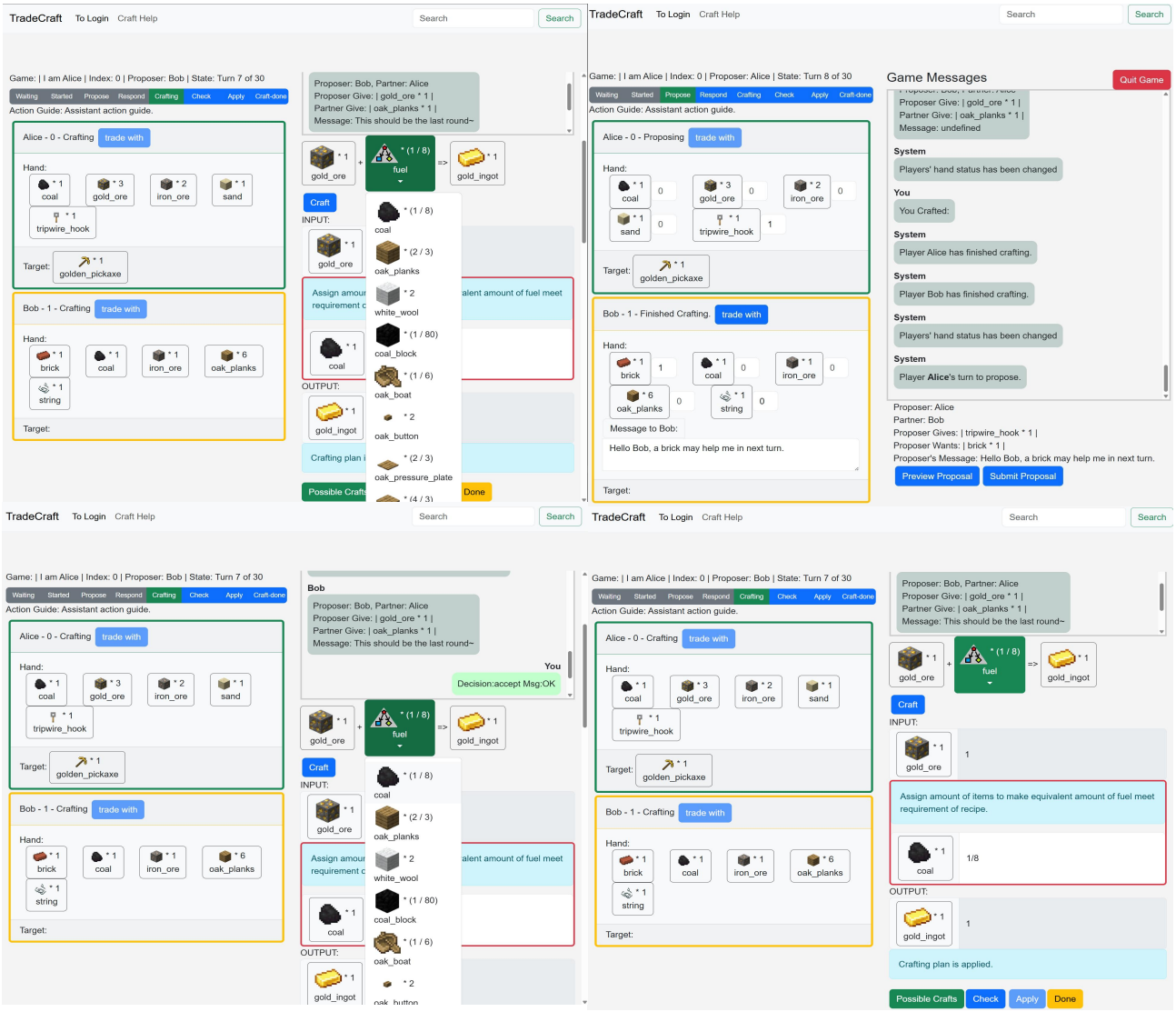

Figure 7. Crafting and trading interface designed based on Minecraft rules. The system restores the original fuel mechanism and incorporates a strict validation process to ensure the correctness of item synthesis. At the end of the crafting phase, the quantities of all items are rounded down to the nearest integer.

Shapeless items: **wooden_button.json**

```
{
  "type": "minecraft:crafting_shapeless",
  "category": "redstone",
  "group": "wooden_button",
  "ingredients": [
    {
      "item": "minecraft:jungle_planks"
    }
  ],
  "result": {
    "item": "minecraft:jungle_button"
  }
}
```

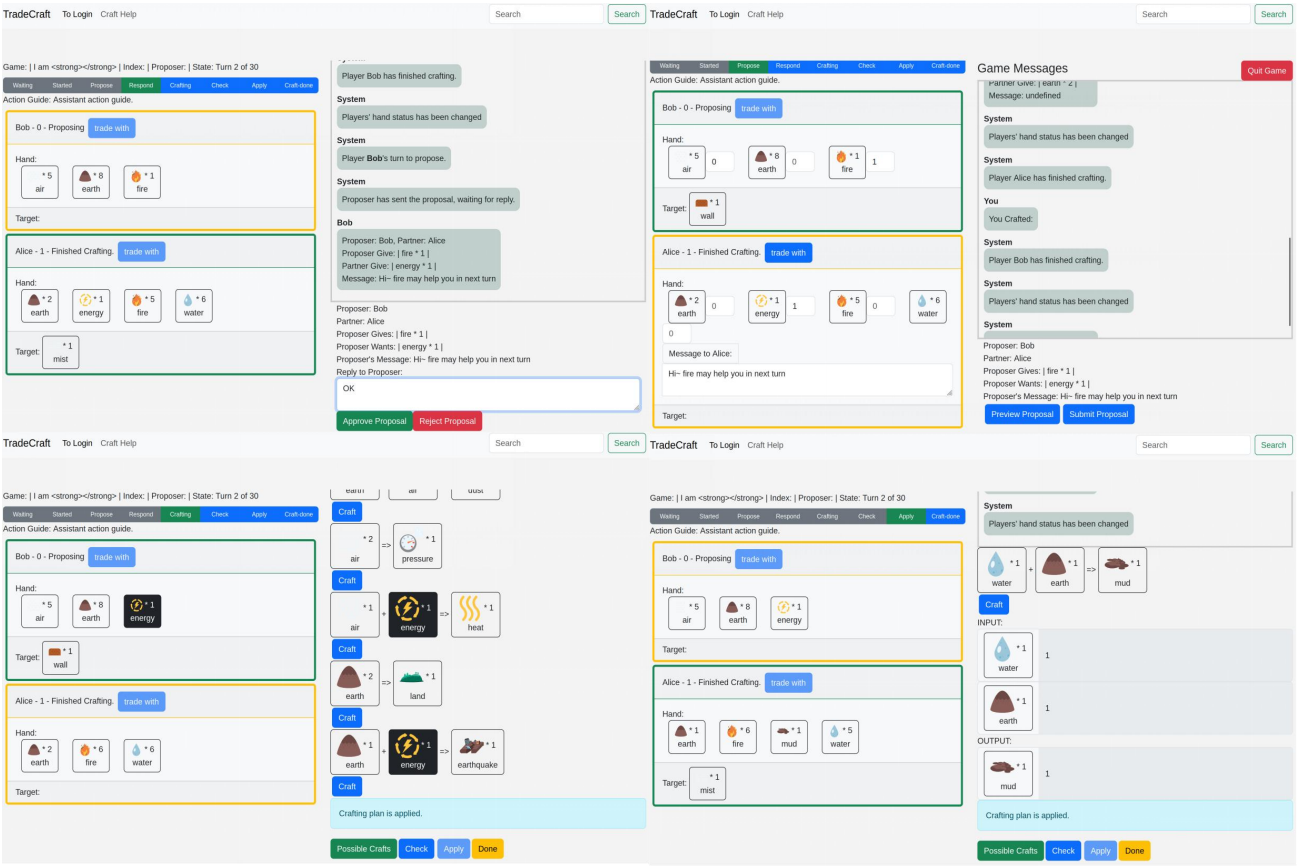

Figure 8. Interface designed based on the rules of Little Alchemy 2. Compared to Minecraft, this environment features more flexible crafting paths and significantly longer synthesis chains, posing greater challenges for agents in terms of long-term planning and situational adaptability.

Shaped items: **diamond_hoe.json**

```
{
  "type": "minecraft:crafting_shaped",
  "category": "equipment",
  "key": {
    "#": {
      "item": "minecraft:stick"
    },
    "X": {
      "item": "minecraft:diamond"
    }
  },
  "pattern": [
    "XX",
    " #",
    " #"
  ],
  "result": {
    "item": "minecraft:diamond_hoe"
  },
  "show_notification": true
}
```

These files locate at `/tradeCraft/src/craft_rules/rule_sets/ruleset/recipes`.

A.2. Format of an Initial Game State (a task instance)

A JSON file with a single task instance looks like:

problem.json


```
[
  {
    "hands": [
        {
            "minecraft:cherry_planks": 1,
            "minecraft:coal": 1,
            "minecraft:iron_ingot": 1,
            "minecraft:raw_copper": 1,
            "minecraft:cobblestone": 1
        },
        {
            "minecraft:oak_planks": 1,
            "minecraft:raw_iron": 5,
            "minecraft:cobblestone": 1,
            "minecraft:raw_copper": 2
        }
    ],
    "targets": [
        {
            "minecraft:shears": 1
        },
        {
            "minecraft:torch": 1
        }
    ]
  }
]
```


where adding new elements in the outer list extends the instance set. The above is a two-player game setting; adding one new entry to both "hands" and "targets" will make it a three-player instance. Note that three-player and two-player instances belong to different game modes, so their files should be copied to the correct paths to avoid exceptions.

A.3. How to add a new ruleset

To add a new ruleset, one may follow the following instructions:

1. Copy the existing ones at `/tradeCraft/src/craft_rules/rule_sets/` and change ruleset name. Copy `recipes`, `tags`, `item_icons` into corresponding folders and remove all temp files.

2. Copy problem sets to `/tradeCraft/src/craft_rules/TC_GAMES/ruleset-name/game-mode`, the detailed structure please refer to the existing ones.

3. Modify configuration file: `/tradeCraft/settings.yaml`, change `craft_rule_choice`, `craft_rule_prefix` (if in your recipe files items have a prefix, such as "minecraft:" in item "minecraft:stick"), and `icon_format` into appropriate ones.

4. Rerun the file `/tradeCraft/run_server.py` in path `/tradeCraft/`.

B. Details of Model-based Evaluation Prompts

B.1. Theory of Mind (ToM) Evaluation

For the evaluation of Theory of Mind (ToM), we designed a structured prompt that instructs the assessor LLM to examine every turn in the game logs and determine whether each player demonstrates first-, second-, or third-order ToM reasoning. The assessment is binary for each dimension: `true` (1) if the behavior is detected, and `false` (0) otherwise. The final score for a given ToM order is computed as the ratio of turns with positive detection to the total number of turns (see Table 3 in the main text).

Below is an excerpt from a real evaluation case, showing how ToM reasoning is detected for a single turn:

Game Log (Turn 8 excerpt)

```
Player 2 THINKS:
"I notice my opponent has stone_bricks,
which might be valuable to them.

Since my goal is to craft a stone shovel,
I could offer raw_copper in exchange.

Since my opponent mentioned they're trying to craft a bucket,
they might need iron."
[First-order ToM]
```

Model Evaluation Output


```
{
  "Turn 8": [
    {
      "user": "player 2",
      "justification": "Player 2 considers what the other
      player needs|first-order ToM.",
      "first_order_tom": true,
      "second_order_tom": false,
      "third_or_higher_tom": false
    },
    {
      "user": "player 1",
      "justification": "Player 1 only evaluates based on their
      own crafting goals|no ToM reasoning.",
      "first_order_tom": false,
      "second_order_tom": false,
      "third_or_higher_tom": false
    }
  ]
}
```


B.2. Other Model-based Dimensions

For the other eight model-based dimensions (e.g., Goal Alignment, Cooperation, Persuasion), we used a similar evaluation pipeline. The assessor LLM receives the complete game log and assigns a score in $[0, 1]$ to each player for each dimension at every turn, with justifications. Final scores are averaged across turns and games. Representative aggregated results are reported in Figures 3(a–i) and Figure 4 of the main text.

Unlike ToM evaluation (binary detection per order), these dimensions are graded continuously, enabling us to capture finer variations in social and strategic behavior.

B.3. Human Validation of Model-based Evaluation

To validate the reliability of our model-based evaluation pipeline, we conducted a small-scale human study. Specifically, we examined three representative dimensions—**Theory of Mind (ToM)**, **Persuasion**, and **Adaptability**—where subjective interpretation could play a critical role. A subset of game logs was sampled, and human raters were asked to perform the same evaluations.

Unlike the model-based evaluation, where the entire game log is processed at once, we presented the records to human annotators on a **turn-by-turn basis**. This design reduced cognitive load and avoided potential fatigue, ensuring that participants could focus on evaluating each player's behavior within a single turn. For each turn, annotators judged (i) the presence of first-/second-/higher-order ToM reasoning (binary), (ii) the strength of persuasion, and (iii) the degree of adaptability (both scored in $[0, 1]$).

The comparison between human annotations and model-based scores is summarized below:

- **ToM judgment consistency**: 86.3% agreement (flattened across ToM levels), indicating strong alignment between human and LLM-based judgments.
- **Persuasion**: Mean Absolute Error (MAE) = 0.236 (score range: 0–1).
- **Adaptability**: MAE = 0.281 (score range: 0–1).

These results suggest that the automated evaluation pipeline is reasonably consistent with human judgments, particularly in ToM detection, where alignment exceeded 85%. For more graded dimensions such as persuasion and adaptability, the moderate MAE values indicate that while the assessor LLM may not perfectly mirror human perception, it nonetheless provides a reliable approximation. This strengthens confidence in the validity of our model-based evaluation framework and supports its use for large-scale, systematic assessment of LLM behaviors in *TradeCraft*.