TradeCraft: AN ARENA FOR LONG-TERM STRATEGIC SOCIAL REASONING AND PLANNING

Anonymous authors
Paper under double-blind review

ABSTRACT

Modern large language models (LLMs) demonstrate strong capabilities in planning and social reasoning when evaluated separately. However, solving problems in social environments typically requires the integration of both reasoning and social skills, posing a greater challenge. We present *TradeCraft*, a flexible and extensible multi-agent environment that embeds strict reasoning and planning requirements into socially grounded tasks. *TradeCraft* integrates trading, negotiation, and multi-step item crafting, supporting two rule sets: a Minecraft-inspired system, and a Little Alchemy 2–inspired system, each with about 1000 items and over 1000 formulas. The environment provides both a web-based GUI for human participation and a text-based API (compatible with `gymnasium`) for LLM agents, enabling diverse forms of human–AI and multi-agent interaction. To catalyze further research, we introduce a workflow-based LLM agent that leverages task-specific prompting and ReAct mechanisms for trading and crafting, while exhibiting configurable social preferences ranging from cooperative to competitive. We further conduct a nine-dimensional evaluation through self-play experiments, analyzing cooperation, goal alignment, information utilization, theory of mind, and other aspects across multiple LLMs and strategy-guidance settings. *TradeCraft* is open source at `https://github.com/TradeCraft-team/TradeCraft`.

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., 2023b;a), 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. 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 introduce *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

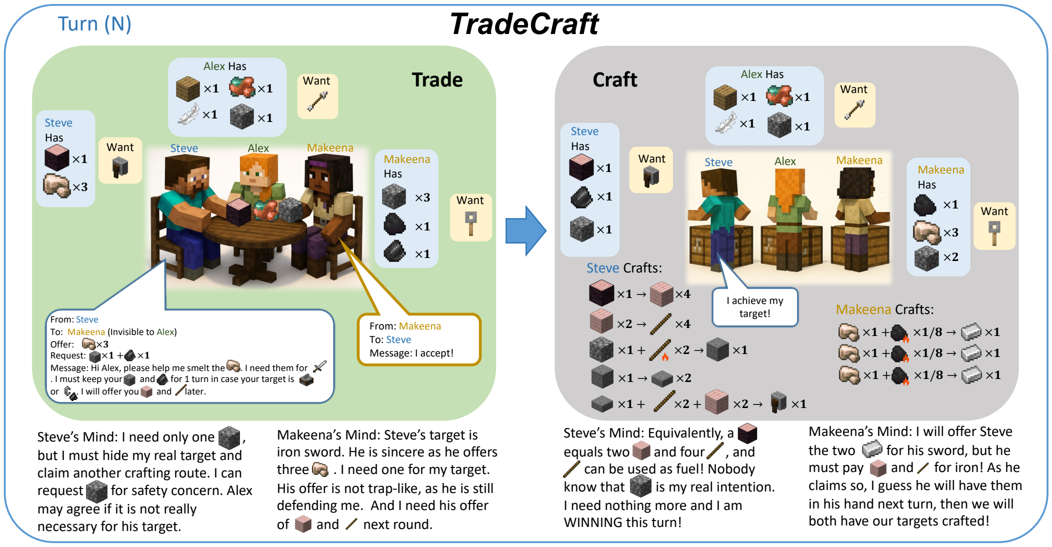


Figure 1: TradeCraft involves social interaction, deep reasoning, long-term planning, and fine-grained control. Players engage in social negotiation and trading, then synthesize target items through long-term planning and precise control. Achieving success requires high-order theory of mind: reasoning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

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, open-world 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.

In summary, our contributions are as follows: (1) *TradeCraft* Environment: We propose *TradeCraft*, a novel open-ended multi-agent environment with a compositional crafting and trading system, which explicitly targets the evaluation of complex social reasoning capabilities together with long-term planning in agents. Our platform supports both AI and human players, facilitating rigorous human-AI comparison studies.

(2) A basic evaluation paradigm: We define a suite of benchmark tasks within *TradeCraft* that involve both social and planning abilities. We demonstrate how these abilities in different fields are evaluated and see whether ability integration brings new challenge to AI agents.

(3) Benchmark Results of LLMs: We release the initial evaluation protocols and baseline results, laying the groundwork for the community to develop and benchmark more agents on *TradeCraft*.

2 The *TradeCraft* Environment

2.1 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

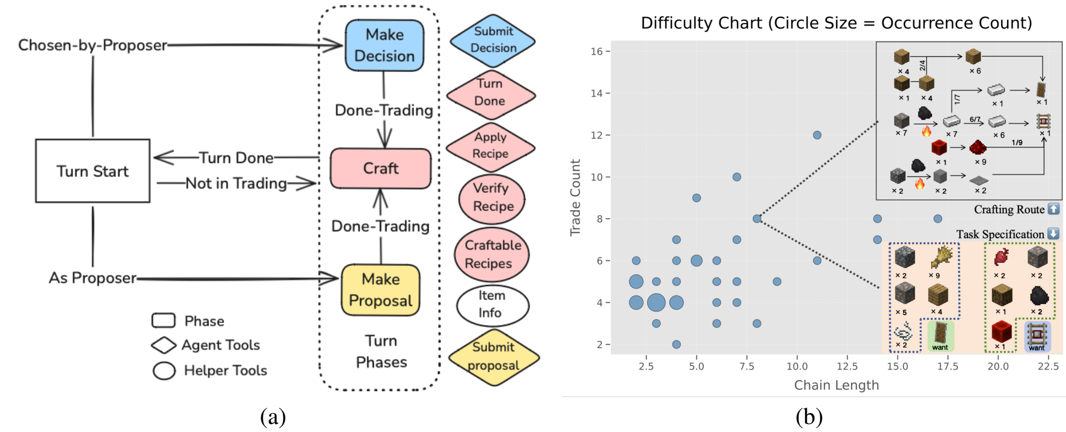


Figure 2: (a) Pipeline of the *TradeCraft* game. (b) Distribution of task difficulty in our game set, defined by the combined complexity of craft chain length and required trade interactions, together with a case study of game initial state and the crafting route.

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 B). 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.

2.2 Game Implementation and Interfaces

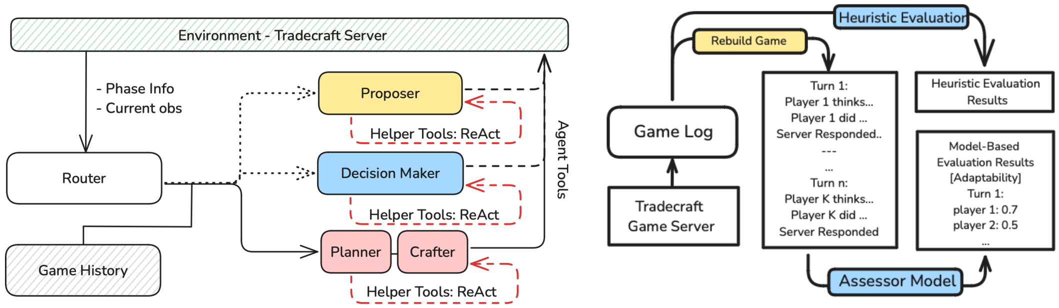
The *TradeCraft* implementation consists of a server and two types of user-interfaces.

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, shown in Figure. 2(a), while expansion with customized tools are supported.

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 in `langchain` tool format, accepting



(a) (b)

Figure 3: **Methods.** Left: the multi-role agent design, where roles *Router*, *Proposer*, *Decision Maker*, *Planner* and *Crafter* are integrated in a whole pipeline. Right: The evaluation pipeline for various dimensions we designed.

an argument dictionary. The end-of-phase tools affects the game state, containing submit proposal, submit decision, make craft, etc., while within-phase tools are for querying information, such as item info, game history, available crafts, etc. Using a tool leads to a new observation about change of game state or the response to a query.

2.3 Initial Game States

A game state is the hand items and targets of all players at some time. Since all target items must be craftable from the union of all players' hand items, initial states (referred to as *initials*) that are not carefully designed tend to be invalid or trivial. For the Minecraft ruleset, we provide 40 predefined initial setups for the 1-vs-1 game mode, covering a range of difficulty levels (Figure 2(b)). The initials 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 B). The difficulty of an initial setup is assessed along two dimensions: the length of the crafting chain and the minimum number of trading steps required. In designing these setups, 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 promote strategic competition and planning, initial hand items are not of exact amount to craft all goals. Redundant items helps introduce alternative crafting routes or provides distracts or deceiving targets. This design enhances the complexity of the game and allows for better evaluation of the model's intelligence.

3 Methods

3.1 Construction of *TradeCraft* Agent

Our agent is built on the `gymnasium` API and operates by selecting actions based on the observations made during each loop cycle. The agent follows a multi-role architecture, with each role adhering to the ReAct framework, maintaining an individual context, and performing specific tasks. To align with the game's different phases, we have designed distinct roles: the *Router*, which activates other roles, and the active roles of *Proposer*, *Decision Maker*, and *Crafter*, each corresponding to specific phases of the game. Additionally, a *Planner* is responsible for generating actionable plans that guide the Crafter's actions, as illustrated in Figure 3(a). Furthermore, a "game history" is maintained as a list of logs, which informs the decision-making process of each role. This history is refined at each turn to prevent exceeding the context window.

The ReAct workflow within each role generates rich, intermediate thoughts, which are effectively utilized by tools either during the working steps or as the final output. We group these logs for further evaluation and analysis.

Table 1: Three groups of evaluation metrics derived from *TradeCraft* gameplay logs. Group 1 consists of outcome-oriented heuristic indicators directly measurable from logs; group 2 captures higher-level social and strategic qualities, scored by LLM; group 3 specifies to detect signals of ToM in various orders, scored by LLM.

Group	Metrics and Description
Heuristic, outcome-oriented signals	**Win–Loss Record**: success in achieving the designated target item. **Invalid Behavior per Turn**: frequency of protocol-violating actions (e.g., infeasible trades, empty proposals). **Average Game Turns**: mean number of turns to complete a game (lower indicates higher efficiency). **Proposal Rejection Rate**: proportion of trade proposals rejected by the opponent.
Model-based social and strategic qualities	**Goal Alignment**: consistency of actions and proposals with the designated target. **Cooperation**: pursuit of mutual benefit (e.g., equitable trades, shared progress). **Adaptability**: flexibility in response to dynamic states and opponent behavior. **Intention Concealment**: deliberate obfuscation of goals via selective disclosure or misdirection. **Strategic Planning**: evidence of long-horizon reasoning, resource management, and contingency planning. **Self-Interested Behavior**: prioritization of individual payoff relative to collective benefit. **Information Utilization**: effectiveness in leveraging signals such as opponent actions and resource states. **Persuasion**: ability to influence opponents through communication or structured proposals.
Model-based ToM of various orders	**1st order**: Reasoning about the opponent's current beliefs or goals. *E.g., "I think they want X."* **2nd order**: Reasoning about what the opponent believes about oneself. *E.g., "I think she thinks that I want more resources."* **3rd or higher**: Deeper recursive belief reasoning across three or more levels. *E.g., "I guess she thinks that I know she will reject the proposal."*

3.2 Evaluations

The gameplay logs generated by agents in *TradeCraft* comprise both *behaviors* (i.e., observable actions) and *thoughts* (i.e., internal reasoning traces such as chain-of-thought logs). To systematically analyze these materials, we consider two complementary groups of indicators, *heuristic* and *model-based*, each reflecting different aspects of strategy and outcome.

As summarized in Table 1, the first two groups of metrics are complementary while the third is extracted from the second as a special part. The heuristic outcome-oriented signals provide direct, quantitative evidence of task performance in terms of success, efficiency, and rule adherence. In contrast, the model-based social and strategic qualities capture more nuanced aspects of behavior, such as cooperation, persuasion, and long-term planning, which are not directly measurable from raw outcomes but are critical for understanding strategic competence in social interaction settings. As ToM is multi-dimensionally structured and plays an important role in capturing recursive belief reasoning central to negotiation, they are taken out from group 2 and form a single group.

In sum, these metrics allow us to evaluate LLMs in a holistic manner—linking basic performance outcomes, social-cognitive reasoning, and role-driven behavioral traits—thus offering a comprehensive basis for subsequent analysis.

4 Experiments

We evaluate our agents in *TradeCraft* using the defined metrics, focusing on a controlled **two-player** setting though our system allows more. This choice reduces design complexity and highlights ToM signals, which become more probable to be trivial in larger groups where higher-order ToM on two agents induces self-reflection and thus more variable social behaviors (Wang et al., 2024).

As described in Section 2.3, our benchmark includes five initial game states of varying difficulty. For each state, we perform **cross-play** between agents driven by pairs of LLMs instantiated with different social preferences: cooperative, competitive, and unprimed. Gameplay data, including both behaviors and reasoning traces, are exported from MongoDB in JSON format.

For evaluation, we employ **Gemini-1.5-Pro** as the assessor. As shown in Figure 3(b), records are reconstructed into per-turn utterances, actions, proposals, and server responses. The assessor assigns per-turn scores in $[0, 1]$ for metrics in groups 2, and identifies ToM levels using dedicated prompts. Player-level scores are obtained by averaging across turns and across repeated games or seeds. We report both per-case results and macro-averages, further stratified by persona and role to analyze how style conditions influence strategy, social behavior, ToM, and ultimately win/loss outcomes.

4.1 Model-Level Behavioral Comparison

We conducted pairwise matchups between agnets driven by GPT-4o, Claude-3.7-Sonnet, and Gemini-1.5-Pro to systematically analyze the performance variance among the models. As an agent's behavior is not independent of its opponent's behavioral pattern, Figures 4(a)-(c) present radar plots based on specific pairwise interactions, rather than aggregating scores across all games for a single agent.

From the overall results (Figure 2(a)), we observe that the Claude-3.7-Sonnet–based agent demonstrates a clear advantage over GPT-4o and Gemini-1.5-Pro across multiple model-based evaluation dimensions. Claude also shows relatively uniform scores across metrics, suggesting a stable and balanced decision pattern. In contrast, GPT-4o reveals a distinct bias—scoring lower in *Cooperation* but higher in *Intention Concealment*—indicating a strategy that prioritizes goal preservation and strategic ambiguity over collaboration. Gemini-1.5-Pro, by comparison, performs weakest in *Persuasion* and *Intention Concealment*, making it the most transparent and least deceptive of the three.

Pairwise contexts reveal additional dynamics. Gemini tends to behave more cooperatively against GPT-4o, yet its *Cooperation* score drops notably when facing Claude. Conversely, GPT-4o conceals intentions strongly when paired with Gemini, but this tendency diminishes against Claude. These findings suggest that model behavior is highly context-dependent, shaped not only by internal policies but also by the opponent's behavioral profile or **social orientation**.

Heuristic metrics complement these observations. GPT-4o records the highest rate of invalid actions and rejects nearly 90% of trade proposals, reinforcing its uncooperative tendencies. Gemini frequently accepts proposals, aligning with its cooperative orientation. Meanwhile, the *Average Game Turn* metric highlights strategic differences: GPT-4o tends to finish games quickly, reflecting aggressive goal pursuit, whereas Claude's longer interactions suggest a more cautious, deliberative style. Together, these divergences underscore the value of studying LLMs' social orientations in interactive environments.

Figure 5(a) summarizes outcomes across all pairwise matchups. GPT-4o and Claude-3.7-Sonnet show strategic parity (4:4 win-lose), while GPT-4o dominates Gemini-1.5-Pro (7:3) with no mutual wins or losses, suggesting strong exploitation of weaker strategies. Claude also outperforms Gemini (4:2), but the presence of four mutual-loss games points to potential instability or risk-prone decisions. Gemini-1.5-Pro, by contrast, fails to secure dominance in any pairing.

Those analysis above suggests that GPT-4o might prefer a more competitive and aggressive strategy in *TradeCraft*, whereas Gemini-1.5-Pro tends to exhibit more cooperative behavior. Although Claude-3.7 demonstrates the most well-rounded behavioral profile according to our model-based evaluation, its inability to consistently secure wins against less adversarial opponents highlights a potential limitation in effectively capitalizing on cooperative dynamics.

4.2 ToM in TradeCraft

And regarding **Theory of Mind**, we also evaluated the pairwise interaction logs across different foundation models. Our updated results show that none of the three models exhibits any ToM behavior beyond the first order:

(a) GPT-4o vs. Claude (b) Gemini vs. GPT-4o (c) Gemini vs. Claude

(d) competitive vs. competitive (e) cooperative vs. cooperative (f) unprimed vs unprimed

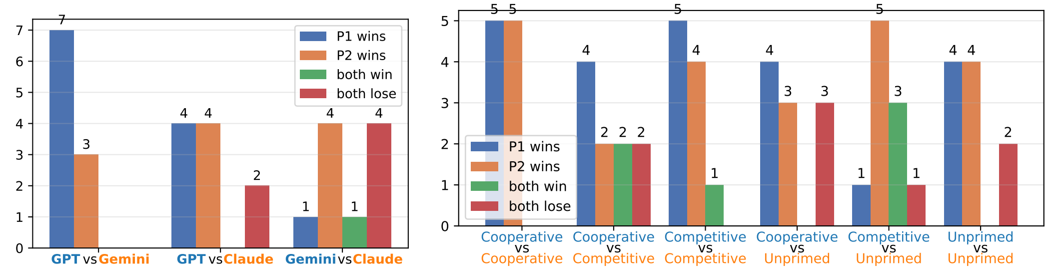
(g) competitive vs. cooperative (h) unprimed vs. cooperative (i) competitive vs. unprimed

(j) Different Models (k) Different Social Preference

Figure 4: Evaluation results across all agent settings. (a-c) : Pairwise comparisons between different foundation models (GPT-4o, Claude-3.7-Sonnet, and Gemini-1.5-Pro). (d-f) : Matchups between agents with the same social preference (*competitive*, *cooperative*, and *unprimed*), revealing internally consistent behavioral patterns. (g-i): Interactions between agents with differing social preferences, highlighting behavioral asymmetries such as intention concealment and cooperation. (j-k): Heuristic evaluation metrics aggregated by model type and social preference, respectively.

Overall, all three models demonstrate relatively strong **first-order ToM** ability—being able to infer or speculate about the opponent's immediate goals and intentions. However, we find **no evidence of second-order or higher-order ToM reasoning**, i.e., recursive mental-state attribution such as "I think she thinks that I want more resources."

These observations suggest that in *TradeCraft*-style game-based tasks, LLMs *may not spontaneously* display Theory-of-Mind behaviors without specific prompts to elicit them, even though such reasoning is central to human-like negotiation, deception detection, and perspective-taking. Importantly, *TradeCraft* provides a systematic setting to explore this gap.



(a) model comparisons

(b) social preference comparisons

Figure 5: **Outcome head-to-head comparisons.** (a) results of model comparisons; (b) results of social preference comparisons. Bar of same color with name represents winning counts, green for both win and red for both lose.

Model	**1st-Order ToM**	**2nd-Order ToM**	**3rd-or-Higher ToM**
Claude-3.7-Sonnet	0.867	0.0	0.0
GPT-4o	0.736	0.0	0.0
Gemini-1.5-Pro	0.840	0.0	0.0

Table 2: Observed Theory-of-Mind levels across three LLMs in *TradeCraft*.

We will release our detailed ToM evaluation methodology together with the raw logs in our repository upon approval, and update the corresponding sections in the revised manuscript.

4.3 Impact of Social Preferences on Game Outcome

Having established how different models behave under a shared task, we next examine how intentional prompt design—specifically manipulating an agent's social preference—shapes behavior and outcomes. To this end, we modify the agent prompts to induce distinct social orientations. Alongside the baseline **Gemini-Unprimed**, which follows only the gameplay objective of crafting its assigned target item, we define two variants: **Gemini-Cooperative**, which emphasizes collaboration and mutual benefit, and **Gemini-Competitive**, which promotes adversarial behavior and goal obstruction. Table 3 summarizes the detailed prompts for each agent type.

Table 3: Agent assigned with different social preference (Driven only by Gemini)

Agent Name	**Specified Social Preferences**
Unprimed	You do not need to consider the other team's goal—treat them as neutral trading partners.
Cooperative	Support your opponent's progress through information-sharing and fair trade.
Competitive	Attempt to disrupt or delay your opponent's progress toward their target.

Figure 4(d–i, k) reports the evaluation results for agents under different **social orientations**. Agents sharing the same orientation exhibit relatively consistent behavioral patterns when paired against each other (Figure 4(d–f)). By contrast, subplots (g–i), which show cross-orientation matchups, reveal clear asymmetries: **Competitive** agents tend to conceal intentions more deliberately and display stronger *Self-Interested Behavior*, whereas **Cooperative** agents consistently achieve higher scores in *Cooperation*.

Interestingly, the Cooperative orientation enhances more than just collaboration. Compared to other types, Cooperative agents also achieve higher scores in *Goal Alignment*, *Strategic Planning*, *Adaptability*, and *Information Utilization*. Heuristic metrics reinforce this pattern: Cooperative agents commit fewer invalid actions and accept nearly all incoming trade proposals. We hypothesize that this stems from an underlying "helper-agent" bias in LLMs, which are often trained to assist and align with user intent by default.

Figure 5(b) further summarizes head-to-head outcomes across the three orientations. Diagonal entries (i.e., matches with the same social orientation) show near-even win rates, reflecting the fairness and stability of our evaluation setup. Across cross-orientation matchups, **Cooperative agents achieve**

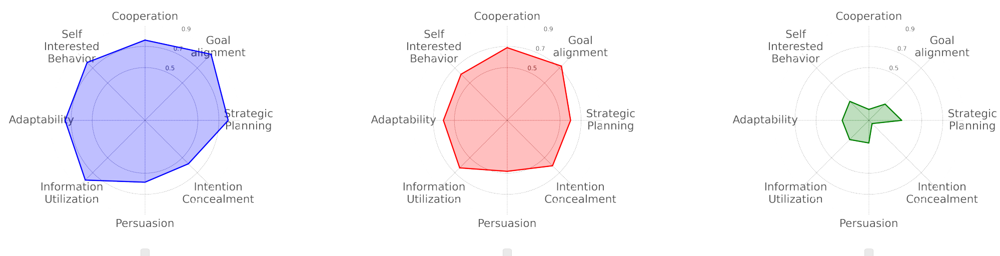
the highest overall win rates. In particular, both Cooperative and Unprimed agents outperform Competitive ones (with win ratios of 5:1 and 4:2, respectively).

These findings not only demonstrate the multifaceted influence of different social orientations on LLM behavior, but also highlight *TradeCraft* as a benchmark environment capable of systematically revealing such orientation-driven dynamics.

4.4 Correlating Evaluation Dimensions with Task Success

Building on the finding that social orientations shape agent behavior and outcomes, we next examine which specific behavioral dimensions are most predictive of successful task completion.

More concretely, we ask: *Which of the eight model-based evaluation dimensions best predict whether an agent ultimately succeeds in crafting its target item?* We aggregated all 60 game records from the previous experiments, identified winners and losers in each game, and compared their average model-based evaluation scores. The results are shown in Figure 6.



(a) Winners (b) Losers (c) Delta

Figure 6: The model-based evaluation results for (a) all winners (b) all losers and (c) the delta.

From the Delta values ($\Delta = \frac{|\text{Winner's Score} - \text{Loser's Score}|}{\text{Loser's Score}}$) in Figure 6(c), we find little difference between winners and losers in behavioral dimensions such as *Cooperation* and *Intention Concealment*. By contrast, winners score notably higher in strategy-oriented dimensions, including *Goal Alignment*, *Strategic Planning*, and *Information Utilization*.

This suggests that in *TradeCraft*, social orientation influences outcomes only indirectly, whereas task-focused abilities—planning and execution toward assigned goals—are the more direct determinants of success. Still, cooperative prompting may enhance adaptability and planning, indirectly boosting win rates.

Importantly, these results highlight *TradeCraft* as a benchmark that not only reveals outcome differences, but also disentangles the strategic and social factors underlying LLM performance.

5 Conclusion

TradeCraft provides a rigorous benchmark with structured tasks and dynamic interactions for evaluating social intelligence through strategic reasoning, planning, and Theory of Mind. For a detailed discussion of related work, see Appendix A. Evaluations of large language models highlight current limitations and guide future progress toward socially intelligent agents.

References

Kelsey Allen, Franziska Brändle, Matthew Botvinick, Judith E Fan, Samuel J Gershman, Alison Gopnik, Thomas L Griffiths, Joshua K Hartshorne, Tobias U Hauser, Mark K Ho, et al. Using games to understand the mind. *Nature human behaviour*, 8(6):1035–1043, 2024.

Bowen Baker, Ingmar Kanitscheider, Todor Markov, Yi Wu, Glenn Powell, Bob McGrew, and Igor Mordatch. Emergent tool use from multi-agent autocurricula. *arXiv preprint arXiv:1909.07528*, 2019.

Chris Baker, Rebecca Saxe, and Joshua B Tenenbaum. 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.

Anton Bakhtin, Noam Brown, Shuohui Chien, Yi Chu, Douwe Kiela, Shibl Mourad, Timo Schick, Arthur Szlam, Emily Dinan, Dhruv Batra, et al. Human-level play in the game of diplomacy by combining language models with strategic reasoning. *Science*, 378(6624):1067–1074, 2022.

Anton Bakhtin, David J Wu, Adam Lerer, Jonathan Gray, Athul Paul Jacob, Gabriele Farina, Alexander H Miller, and Noam Brown. 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`.

Nolan Bard, Jakob N Foerster, Sarath Chandar, Neil Burch, Marc Lanctot, H Francis Song, Emilio Parisotto, Vincent Dumoulin, Shibl Mourad, Brooks Larson, et al. The hanabi challenge: A new frontier for ai research. *Artificial Intelligence*, 280:103216, 2020.

Franziska Brändle, Lena J Stocks, Joshua B Tenenbaum, Samuel J Gershman, and Eric Schulz. Empowerment contributes to exploration behaviour in a creative video game. *Nature Human Behaviour*, 7(9):1481–1489, 2023.

Madeline Carroll, Rohin Shah, Mark K Ho, Thomas L Griffiths, Pieter Abbeel, and Anca D Dragan. On the utility of learning about humans for human-ai coordination. In *Advances in Neural Information Processing Systems*, volume 32, 2019.

Karl Cobbe, Vineet Kosaraju, Mohammad Bavarian, Mark Chen, Heewoo Jun, Lukasz Kaiser, Matthias Plappert, Jerry Tworek, Jacob Hilton, Reiichiro Nakano, et al. Training verifiers to solve math word problems. *arXiv preprint arXiv:2110.14168*, 2021.

Linxi Fan, Guanzhi Wang, Yunfan Jiang, Ajay Mandlekar, Yuncong Yang, Haoyi Zhu, Andrew Tang, De-An Huang, Yuke Zhu, and Anima Anandkumar. 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`.

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

Muzhi Han, Yifeng Zhu, Song-Chun Zhu, Ying Nian Wu, and Yuke Zhu. 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 He, Jordan Boyd-Graber, Hal Daumé III, and Kevin Kwok. Opponent modeling in deep reinforcement learning. In *International Conference on Machine Learning*, pp. 1804–1813. PMLR, 2016.

Yuzhen Lei, Hongbin Xie, Jiaxing Zhao, Shuangxue Liu, and Xuan Song. Mscore: A benchmark for multi-stage collaborative reasoning in llm agents, 2025. URL `https://arxiv.org/abs/2509.17628`.

Joel Z Leibo, Chelsea Hughes, Marc Lanctot, Jean-Baptiste Lespiau, Vinicius Zambaldi, Evan Lockhart, Jeff Clune, and Thore Graepel. Scalable evaluation of multi-agent reinforcement learning with melting pot. In *International Conference on Machine Learning*, pp. 6187–6199. PMLR, 2021.

Joon Sung Park, Joseph O'Brien, Carrie J. Cai, Meredith Ringel Morris, Percy Liang, and Michael S. Bernstein. Generative agents: Interactive simulacra of human behavior. *arXiv preprint arXiv:2304.03442*, 2023.

Siyuan Qi, Shuo Chen, Yexin Li, Xiangyu Kong, Junqi Wang, Bangcheng Yang, Pring Wong, Yifan Zhong, Xiaoyuan Zhang, Zhaowei Zhang, Nian Liu, Wei Wang, Yaodong Yang, and Song-Chun Zhu. 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`.

Neil C Rabinowitz, Frank Perbet, H. Francis Song, Chiyuan Zhang, S. M. Ali Eslami, and Matthew Botvinick. Machine theory of mind. In *International Conference on Machine Learning*, pp. 4218–4227. PMLR, 2018.

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

Nadav Sclar, Moshe Anson, Panos Achlioptas, Avinatan Hassidim, Tammie Halperin, Tal Yahav, Amos Korman, and Amir Feder. Symmtom: A symmetric theory of mind benchmark for multiagent communication. In *Advances in Neural Information Processing Systems*, volume 35, pp. 28402–28415, 2022.

James W. A. Strachan, Dalila Albergo, Giulia Borghini, Oriana Pansardi, Eugenio Scaliti, Saurabh Gupta, Krati Saxena, Alessandro Rufo, Stefano Panzeri, Guido Manzi, Michael S. A. Graziano, and Cristina Becchio. Testing theory of mind in large language models and humans. *Nature Human Behaviour*, 8(7):1285–1295, 07 2024. ISSN 2397-3374. doi: 10.1038/s41562-024-01882-z. URL https://doi.org/10.1038/s41562-024-01882-z.

Winnie Street, John Siy, Geoff Keeling, Adrien Baranes, Benjamin Barnett, Michael McKibben, Tatenda Kanyere, Alison Lentz, Blaise Arcas, and Robin Dunbar. Llms achieve adult human performance on higher-order theory of mind tasks. 05 2024. doi: 10.48550/arXiv.2405.18870.

Haoxiang Sun, Yingqian Min, Zhipeng Chen, Wayne Xin Zhao, Lei Fang, Zheng Liu, Zhongyuan Wang, and Ji-Rong Wen. Challenging the boundaries of reasoning: An olympiad-level math benchmark for large language models. *arXiv preprint arXiv:2503.21380*, 2025.

Guanzhi Wang, Yuqi Xie, Yunfan Jiang, Ajay Mandlekar, Chaowei Xiao, Yuke Zhu, Linxi Fan, and Anima Anandkumar. Voyager: An open-ended embodied agent with large language models. *arXiv preprint arXiv: Arxiv-2305.16291*, 2023.

Junqi Wang, Chunhui Zhang, Jiapeng Li, Yuxi Ma, Lixing Niu, Jiaheng Han, Yujia Peng, Yixin Zhu, and Lifeng Fan. 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.

Jason Wei, Xuezhi Wang, Dale Schuurmans, Maarten Bosma, brian ichter, Fei Xia, Ed Chi, Quoc V Le, and Denny Zhou. Chain-of-thought prompting elicits reasoning in large language models. In S. Koyejo, S. Mohamed, A. Agarwal, D. Belgrave, K. Cho, and A. Oh (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.

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

Jian Xie, Kai Zhang, Jiangjie Chen, Tinghui Zhu, Renze Lou, Yuandong Tian, Yanghua Xiao, and Yu Su. Travelplanner: a benchmark for real-world planning with language agents. In *Proceedings of the 41st International Conference on Machine Learning*, pp. 54590–54613, 2024.

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

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

Ceyao Zhang, Kaijie Yang, Siyi Hu, Zihao Wang, Guanghe Li, Yihang Sun, Cheng Zhang, Zhaowei Zhang, Anji Liu, Song-Chun Zhu, 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.

Shuyan Zhou, Frank F Xu, Hao Zhu, Xuhui Zhou, Robert Lo, Abishek Sridhar, Xianyi Cheng, Yonatan Bisk, Daniel Fried, Uri Alon, et al. Webarena: A realistic web environment for building autonomous agents. *arXiv preprint arXiv:2307.13854*, 2023.

A Related Work

A.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.

A.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.

A.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 performance 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.

B 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.

B.1 Format of a Crafting Formula

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

Shapeless items: **wooden_button.json**

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

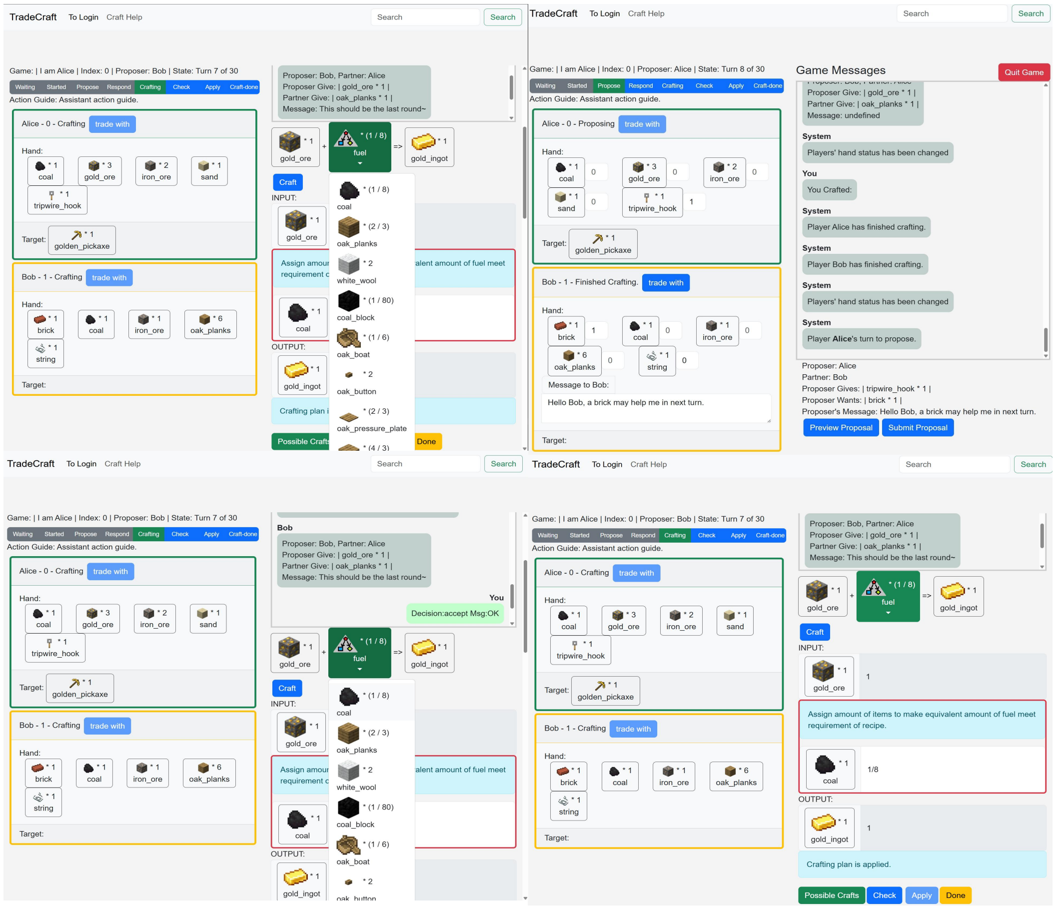


Figure 7: Crafting and trading interface designed based on Minecraft rules. The system restores the original fuel mechanism and incorporates a strict validation process to ensure the correctness of item synthesis. At the end of the crafting phase, the quantities of all items are rounded down to the nearest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

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`.

B.2 Format of an Initial Game State (a problem)

A JSON file with a single problem lookes like:

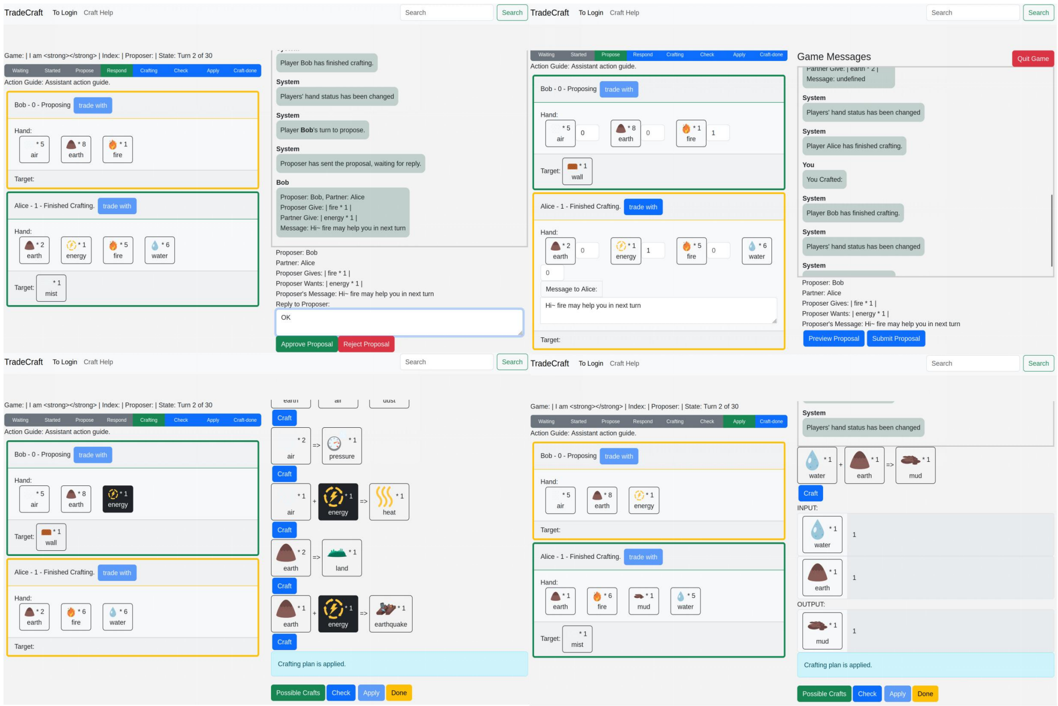


Figure 8: Interface designed based on the rules of Little Alchemy 2. Compared to Minecraft, this environment features more flexible crafting paths and significantly longer synthesis chains, posing greater challenges for agents in terms of long-term planning and situational adaptability.

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 problem set. The above is a two-player game setting, adding one new entry on both “hands” and “targets” will make it a three-player problem. Note that three-player and two-player problems belong to different game modes, their files should be copied to correct paths to avoid exceptions!

B.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/`.

C Details of Model-based Evaluation Prompts

C.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
    }
  ]
}
```


C.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.

C.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*.