AsymPuzl: An Asymmetric Puzzle for multi-agent cooperation

Xavier Cadet *
Dartmouth College
Hanover, NH 03755

Edward Koh
Dartmouth College
Hanover, NH 03755

Peter Chin
Dartmouth College
Hanover, NH 03755

Abstract

Large Language Model (LLM) agents are increasingly studied in multi-turn, multi-agent scenarios, yet most existing setups emphasize open-ended role-play rather than controlled evaluation. We introduce AsymPuzl, a minimal but expressive two-agent puzzle environment designed to isolate communication under information asymmetry. Each agent observes complementary but incomplete views of a symbolic puzzle and must exchange messages to solve it cooperatively. Using a diverse set of current-generation and open-source LLMs, we show that (i) strong models such as GPT-5 and Claude-4.0 reliably converge across puzzle sizes on the solution by sharing complete information in two turns, (ii) weaker models often ignore partner messages or over-correct their hypotheses, and (iii) feedback design is non-trivial: simple self-feedback improves success rates, while detailed joint feedback can hurt performance. These findings show that even in simple cooperative tasks, LLM communication strategies diverge and depend on the granularity of feedback signals. AsymPuzl thus provides a testbed for probing the limits of multi-turn cooperation and opens avenues for studying coordination mechanisms.

1 Introduction

Autonomous agents using Language Models (LLMs) offer promising opportunities for problem-solving [3, 2]. Nonetheless, real-world problems often require multi-turn and collaboration under partial information. In many tasks, agents often have access to complementary yet potentially incomplete information, as seen in distributed decision-making or human-ai cooperation. Information asymmetry is a common phenomenon in practice and requires effective communication to address the issue. Current Multi-Agent LLM studies emphasize *role-play* and *open-ended dialogue* but lack controlled testbeds for evaluating communication strategies. Moreover, existing Large Language Model Multi-Agent Systems (LLM-MAS) rarely allow systematic manipulation of task difficulty. As such, we are left with the question: *How do LLM agents adapt their communication under asymmetric information.* We introduce Asympuzl, a minimal yet expressive environment where two agents see complementary partial views of a puzzle and must exchange messages to solve it. We can adjust the difficulty of the task by varying the puzzle size and providing feedback on their hypotheses. Our **contributions** are: i) the **AsymPuzl environment**, a testbed for two agent puzzle solving under information asymmetry, and ii) an **empirical analysis of feedback granularity and communication strategies** using this environment, showing that while feedback can be helpful, it can be lead to reduced performance if not carefully designed, and that while complete information sharing is possible in our experiments, most agents do not default to this strategy.

*xavier.fjf.cadet@dartmouth.edu

39th Conference on Neural Information Processing Systems (NeurIPS 2025) Workshop: AsymPuzl: An Asymmetric Puzzle for multi-agent cooperation.

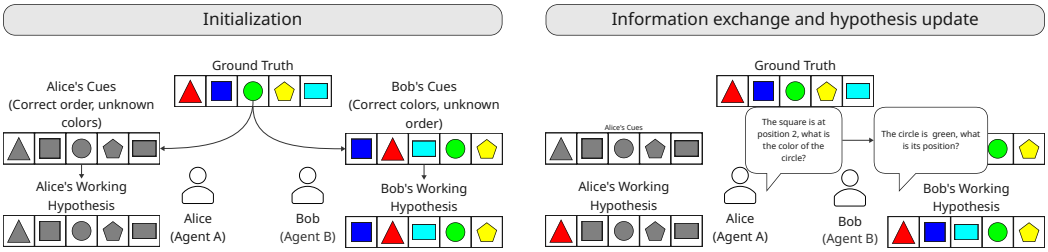


Figure 1: Overview of the puzzle: the ground truth is first created, then each agents' individual partial views are generated and shared with them as clues. The working hypothesis starts as a copy of the clues. Then, in a turn-based interaction, the agents, Alice and Bob , can send each other messages and update their working hypothesis until their hypotheses match the ground truth.

2 Related Work

Puzzle-based reasoning: Puzzle solving is a popular method to evaluate the reasoning capacities of LLMs [16, 4, 9, 15]. [1] compares LLMs over multi-turn puzzles with a single agent and shows that errors often come from poor instruction following. ZebraLogic [11] isolates the reasoning limits of single LLMs on logical puzzles, demonstrating that as puzzle complexity increases performance of agents drop and increasing model size yields limited improvement. In comparison, this work isolates the communication and coordination in a multi-agent scenario when no single agent holds complete information (sufficient to solve the puzzle alone).

LLM-MAS: Interest in coordinating multiple LLM agents has been increasing [8, 5, 17, 10] alongside concerns relating to emergent behavior during multi-agent interactions and model alignment. iAgents [12] focused on large-scale role-play social networks with information asymmetry. In comparison, our environment is deliberately minimal, designed to reduce the effect of confounding factors and provide fine grained insights into how LLMs adapt communication protocols.

3 The AsymPuzl Environment

AsymPuzl is a two-agent asymmetric puzzle-solving environment (We provide an overview in Figure 1) where each participant is given partial information and collaboration is required to reach the solution.

Puzzle setup: Two agents, Alice and Bob , are given complementary views of a position-shape-color matching puzzle. Alice is given a puzzle view with correct positions and shapes, but unknown colors, while Bob is given a puzzle view with correct shapes and colors, but unknown positions. The puzzle state is controlled externally; both agents can communicate with each other and provide a list of actions to apply to their *working hypothesis* (i.e., their current view of the puzzle).

Game loop: A puzzle is generated and the *clues* (initial views) of Alice and Bob are separated. Multiple turns take place, where a turn is: Alice is prompted with the puzzle instruction, cues, working hypothesis, message history (both Alice's and Bob's messages), feedback from the previous turn, and the structure of the format they must respond in. Alice generates its output, which contains a message for Bob and the list of actions to take on its puzzle. Bob follows the same procedure with the latest message from Alice . The actions are applied to the working hypotheses, which are compared against the ground truth to provide feedback for the next turn. This process continues until the puzzle is solved. The agents must provide a formatted JSON at the end of their answer, which contains a list of formatted actions and a message that will be shared with the other agent.

Feedback modes: Feedback is provided to both Alice and Bob as part of their input prompts. The feedback can be 1) *No feedback*: no feedback is provided to the agents 2) *Own*: each agent is told whether its part of the puzzle is solved, 3) *Own detailed*: each agent is told whether its current puzzle is solved and which positions are wrong 4) *Joint*: the agents are told whether the puzzle is solved

(both of them need to have solved it) 4) *Both*: the agents are told whether their and the other agent's parts of the puzzle are solved 5) *Both detailed*: the agents get the equivalent of *Own detailed* but for both agents.

Difficulty levels: The search space for puzzles with N positions and two attributes per position is $(N!)^2$ [12]. Without communication, Alice's and Bob's problems allow for $N!$ possible permutations; nonetheless, given full information, the puzzle can be solved in linear time $O(N)$, where Alice and Bob only need to map the position-shape-color to their current working hypothesis.

4 Experimental Setup

4.1 Models

We evaluated the AsymPuzl environment on a number of LLMs from various vendors: OpenAI (GPT-3.5-turbo, GPT-4o [14], GPT-5, OSS-120B [13]), Meta (Llama 3.2-11B [6]), and Anthropic (Claude-3.5, Claude-4.0). This selection captures a range of reasoning abilities, response tendencies, and cost profiles. Models differ in training scale, safety alignment, and conversational style, allowing us to examine how choice of LLM affects multi-agent communication and problem-solving.

4.2 Evaluation Metrics

For each of our experiments, we set the maximum number of turns to be twice the number of elements in the puzzle; thus, we cut off a 5-piece game after 10 turns. Note that with full information sharing, a puzzle of any size can be solved in 2 turns. Assuming a single piece of information is exchanged each turn, the maximum number of turns still provides margin for error and correction.

Success Percentage: Given a set of puzzles to solve -here simulated via seeds- we compute the percentage of these puzzles that are solved within the maximum number of turns.

Average number of actions per position : For each position, we count the number of times the position is modified. This allows estimation of how many times an agent overwrites its working hypothesis, an indication of whether its decisions are error corrections or just random guesses.

5 Results

5.1 Can the agent communicate and solve the puzzle?

We observed that while the task could be successfully solved by a single agent when provided with all of the information, the two agent problem is challenging for most models. (Table 1) We note that GPT-5 and Claude-4.0 can consistently solve the task for size 5, achieving 100% completion regardless of the type of feedback, whereas other models benefit from different feedback strategies.

5.2 What is the impact of feedback?

We evaluated different forms of feedback for puzzles with five elements and observed that providing individual feedback on each agent's own working hypothesis increased the completion rate. Detailed feedback led to the most improvement, for instance, on GPT-4o: from 43.3% to 63.3% completion. On the other hand, providing detailed information about the other agent's working hypothesis on top of their own hurt performance; this can be attributed to information overload with lack of context (Alice does not see Bob's working hypothesis, yet Alice is told which positions of Bob's hypothesis are incorrect).

5.3 Do agents over-correct their working hypothesis?

In the optimal solution, Alice should only modify each position once and Bob at most once (the working hypothesis could already have correct entries). Both agents would nominally gather the information they need about a position, update the position, and refrain from modifying it further. We observe that GPT-5 and Claude-4.0 are nearly optimal as they usually need only two turns of

	Own Feedback			Joint Feedback		
Feedback Mode Model	No feedback	Own	Own Detailed	Joint	Both	Both detailed
GPT-5	100.0	100.0	100.0	100.0	100.0	100.0
Claude-4.0	100.0	100.0	100.0	100.0	100.0	100.0
OSS-120B	53.3	93.3	100.0	90.0	90.0	96.7
GPT-4o	43.3	46.7	63.3	40.0	80.0	56.7
Claude-3.5	66.7	73.3	86.7	73.3	83.3	36.7
GPT-3.5-turbo	0.0	0.0	0.0	0.0	0.0	0.0
Llama 3.2-11B	0.0	0.0	0.0	0.0	0.0	0.0

Table 1: (Higher is better) Percentage of 5-pieces puzzles solved over 30 seeds. Temperature 0.0. Providing feedback increases the completion percentage, while additionally providing detailed information about the other agent's working hypothesis can hurt performance. (Tables with Wilson 95% Confidence Intervals are provided in Appendix Table 5 and Table 6)

	Both feedback			
Puzzle size Model	3	5	10	20
GPT-5	100.0	100.0	100.0	100.0
Claude-4.0	100.0	100.0	100.0	100.0
OSS-120B	83.3	90.0	90.0	93.3
GPT-4o	60.0	80.0	60.0	16.7
Claude-3.5	50.0	83.3	56.7	16.7
GPT-3.5-turbo	0.0	0.0	0.0	0.0
Llama 3.2-11B	0.0	0.0	0.0	0.0

Table 2: (Higher is better) Success rate for different size puzzles across 30 seeds, with temperature 0.0, and each agent receiving both agent's solved status. As complexity increases, GPT-4o and Claude-3.5 decrease in performance. The lower performance by some models on the 3-pieces puzzle can be explained by miscommunication, leading to wasted turns under a tighter turn constraint.

communication and their agents share all the information they have with one another. The other models tend to exchange one piece of information at a time, and models such as GPT-3.5-turbo and Llama 3.2-11B tend to ignore each other's messages. (See appendix Figure 2).

5.4 What is the impact of the puzzle size?

We further evaluate the performance of the different models on puzzles of size 3, 10, and 20, using joint feedback. We adjust the maximum number of turns to 6, 20, and 40, respectively. We observe that as the puzzle complexity increases, the performance decreases, similarly to [11].

6 Conclusion

We presented Asympuzl, an evaluation testbed for multi-turn cooperative play between LLM agents under information asymmetry. We demonstrated that strong models (e.g., GPT-5 and Claude-4.0) can reliably converge with each agent sharing all of its information as agents ideally would. In contrast, other models struggle with miscommunication or repeated corrections. Our results show that feedback matters: simple self-feedback improves performance, but detailed joint feedback can confuse agents. These findings underscore the importance of carefully designing communication and evaluation protocols in multi-agent LLM systems.

Looking ahead, AsymPuzl can serve as a foundation for more complex studies, such as introducing noisy or ambiguous views, restricting communication bandwidth, or scaling to three or more agents. We hope this testbed will help investigate coordination strategies and contribute to solving the broader challenges of enabling LLMs to collaborate effectively in real-world, multi-turn settings.

Acknowledgments and Disclosure of Funding

This research was funded by the Defense Advanced Research Projects Agency (DARPA), under contract W912CG23C0031.

References

[1] Kartikeya Badola, Jonathan Simon, Arian Hosseini, Sara Marie Mc Carthy, Tsendsuren Munkhdalai, Abhimanyu Goyal, et al. *Multi-Turn Puzzles: Evaluating Interactive Reasoning and Strategic Dialogue in LLMs*. arXiv:2508.10142 [cs] version: 3. Aug. 2025. DOI: 10.48550/arXiv.2508.10142. URL: http://arxiv.org/abs/2508.10142 (visited on 09/02/2025).

[2] Tom B. Brown, Benjamin Mann, Nick Ryder, Melanie Subbiah, Jared Kaplan, Prafulla Dhariwal, et al. "Language Models are Few-Shot Learners". In: *arXiv:2005.14165 [cs]* (July 2020). arXiv: 2005.14165. URL: http://arxiv.org/abs/2005.14165 (visited on 02/21/2022).

[3] Sébastien Bubeck, Varun Chandrasekaran, Ronen Eldan, Johannes Gehrke, Eric Horvitz, Ece Kamar, et al. *Sparks of Artificial General Intelligence: Early experiments with GPT-4*. arXiv:2303.12712 [cs]. Apr. 2023. DOI: 10.48550/arXiv.2303.12712. URL: http://arxiv.org/abs/2303.12712 (visited on 09/02/2025).

[4] Nouha Dziri, Ximing Lu, Melanie Sclar, Xiang Lorraine Li, Liwei Jiang, Bill Yuchen Lin, et al. "Faith and fate: limits of transformers on compositionality". In: *Proceedings of the 37th International Conference on Neural Information Processing Systems*. NIPS '23. Red Hook, NY, USA: Curran Associates Inc., Dec. 2023, pp. 70293–70332. (Visited on 09/02/2025).

[5] Sinem Erisken, Timothy Gothard, Martin Leitgab, and Ram Potham. *MAEBE: Multi-Agent Emergent Behavior Framework*. arXiv:2506.03053 [cs]. July 2025. DOI: 10.48550/arXiv.2506.03053. URL: http://arxiv.org/abs/2506.03053 (visited on 09/02/2025).

[6] Aaron Grattafiori, Abhimanyu Dubey, Abhinav Jauhri, Abhinav Pandey, Abhishek Kadian, Ahmad Al-Dahle, et al. *The Llama 3 Herd of Models*. arXiv:2407.21783 [cs]. Nov. 2024. DOI: 10.48550/arXiv.2407.21783. URL: http://arxiv.org/abs/2407.21783 (visited on 09/02/2025).

[7] Woosuk Kwon, Zhuohan Li, Siyuan Zhuang, Ying Sheng, Lianmin Zheng, Cody Hao Yu, et al. "Efficient memory management for large language model serving with PagedAttention". In: *Proceedings of the ACM SIGOPS 29th symposium on operating systems principles*. 2023.

[8] Guohao Li, Hasan Abed Al Kader Hammoud, Hani Itani, Dmitrii Khizbullin, and Bernard Ghanem. "CAMEL: Communicative Agents for "Mind" Exploration of Large Language Model Society". en. In: Nov. 2023. URL: https://openreview.net/forum?id=3IyL2XWDkG (visited on 07/29/2025).

[9] Yinghao Li, Haorui Wang, and Chao Zhang. "Assessing Logical Puzzle Solving in Large Language Models: Insights from a Minesweeper Case Study". In: *Proceedings of the 2024 Conference of the North American Chapter of the Association for Computational Linguistics: Human Language Technologies (Volume 1: Long Papers)*. Ed. by Kevin Duh, Helena Gomez, and Steven Bethard. Mexico City, Mexico: Association for Computational Linguistics, June 2024, pp. 59–81. DOI: 10.18653/v1/2024.naacl-long.4. URL: https://aclanthology.org/2024.naacl-long.4/ (visited on 09/02/2025).

[10] Tian Liang, Zhiwei He, Wenxiang Jiao, Xing Wang, Yan Wang, Rui Wang, et al. "Encouraging Divergent Thinking in Large Language Models through Multi-Agent Debate". In: *Proceedings of the 2024 Conference on Empirical Methods in Natural Language Processing*. Ed. by Yaser Al-Onaizan, Mohit Bansal, and Yun-Nung Chen. Miami, Florida, USA: Association for Computational Linguistics, Nov. 2024, pp. 17889–17904. DOI: 10.18653/v1/2024.emnlp-main.992. URL: https://aclanthology.org/2024.emnlp-main.992/ (visited on 09/02/2025).

[11] Bill Yuchen Lin, Ronan Le Bras, Kyle Richardson, Ashish Sabharwal, Radha Poovendran, Peter Clark, et al. "ZebraLogic: On the Scaling Limits of LLMs for Logical Reasoning". In: *Forty-Second International Conference on Machine Learning*. 2025.

[12] Wei Liu, Chenxi Wang, YiFei Wang, Zihao Xie, Rennai Qiu, Yufan Dang, et al. "Autonomous Agents for Collaborative Task under Information Asymmetry". In: *The Thirty-Eighth Annual Conference on Neural Information Processing Systems*. 2024.

[13] OpenAI, Sandhini Agarwal, Lama Ahmad, Jason Ai, Sam Altman, Andy Applebaum, et al. *gpt-oss-120b & gpt-oss-20b Model Card*. arXiv:2508.10925 [cs]. Aug. 2025. DOI: 10.48550/arXiv.2508.10925. URL: http://arxiv.org/abs/2508.10925 (visited on 09/02/2025).

[14] OpenAI, Aaron Hurst, Adam Lerer, Adam P. Goucher, Adam Perelman, Aditya Ramesh, et al. *GPT-4o System Card*. arXiv:2410.21276 [cs]. Oct. 2024. DOI: 10.48550/arXiv.2410.21276. URL: http://arxiv.org/abs/2410.21276 (visited on 09/02/2025).

[15] Yufan Ren, Konstantinos Tertikas, Shalini Maiti, Junlin Han, Tong Zhang, Sabine Süsstrunk, et al. *VGRP-Bench: Visual Grid Reasoning Puzzle Benchmark for Large Vision-Language Models*. arXiv:2503.23064 [cs]. Apr. 2025. DOI: 10.48550/arXiv.2503.23064. URL: http://arxiv.org/abs/2503.23064 (visited on 09/02/2025).

[16] Nemika Tyagi, Mihir Parmar, Mohith Kulkarni, Aswin Rrv, Nisarg Patel, Mutsumi Nakamura, et al. "Step-by-Step Reasoning to Solve Grid Puzzles: Where do LLMs Falter?" In: *Proceedings of the 2024 Conference on Empirical Methods in Natural Language Processing*. Ed. by Yaser Al-Onaizan, Mohit Bansal, and Yun-Nung Chen. Miami, Florida, USA: Association for Computational Linguistics, Nov. 2024, pp. 19898–19915. DOI: 10.18653/v1/2024.emnlp-main.1111. URL: https://aclanthology.org/2024.emnlp-main.1111/ (visited on 08/28/2025).

[17] Qingyun Wu, Gagan Bansal, Jieyu Zhang, Yiran Wu, Beibin Li, Erkang (Eric) Zhu, et al. "AutoGen: Enabling next-gen LLM applications via multi-agent conversation". In: *Colm 2024*. Aug. 2024. URL: https://www.microsoft.com/en-us/research/publication/autogen-enabling-next-gen-llm-applications-via-multi-agent-conversation-framework/.

Outline

A Limitations

Instruction and puzzle injection: We adopt a design where the environment re-injects all relevant state into each turn, namely, the agents are provided with the set of instructions, output format requirements, initial cues, current working hypotheses, and feedback about the current working hypotheses. This allows isolating the role of the communication strategy, and avoiding the agent losing track of its original task. While this differs from fully autonomous agent memory, it still constitutes a multi-turn interaction, as agents must iteratively exchange complementary information and solve the task through sequential coordination. We leave extensions toward persistent agent state as future work.

Original cue injection: At each turn, we provide the agents with their original cues. We chose this design so that the agents always have a way to recover their original information. Furthermore, this allows the agent to compare its working hypothesis to its original cues, keeping track of its progress. To translate this to a real-world setting, it would be similar to having the agent query a database and caching the information in a read-only entry so that the agent cannot overwrite the data it is accessing.

Puzzle complexity: We leave for future work the analysis of noisy and ambiguous interactions where, for instance, Alice is given more shapes than required while Bob has the correct number of colors to shape pairs, requiring them to determine the relevant information. This ambiguity can be applied in the opposite direction as well, where Bob would see more color-shape associations than there are valid shapes in the ground truth. Another extension is the evaluation of constraints on communication, or the number of operations per turn, and the addition of an internal scratch pad carried across turns by the agents.

B Evaluation Metrics

Success rate at Turn K: We gather across experimental seeds the number of turns taken to solve each puzzle. This additionally allows us to derive the number of puzzles solved within each given number of turns. We compared two feedback modes, No Feedback and the Joint feedback where both agent see each other's completion status (Figure 3). This experiment showed that the joint feedback increased the success rate and that models such as GPT-5, Claude 4.0, and GPT-4o tend to solve the puzzles in fewer turns.

Number of tokens: We collected the average number of tokens that each agent output per turn and the number of these tokens dedicated to the message sent to the other agent (Table 3). We observed that for most models Bob uses more token, this could be explained by the need to associate both shape and colors in its message while Alice can enumerate the shapes. To compare the different model we use the same tokenizer from tiktoken across models' outputs.

C Solving as a Single Agent with Full Information

To verify that the LLMs could solve the problem with complete information, we conducted multiple experiments of a single agent solving the puzzle across 10 seeds. (Table 4).

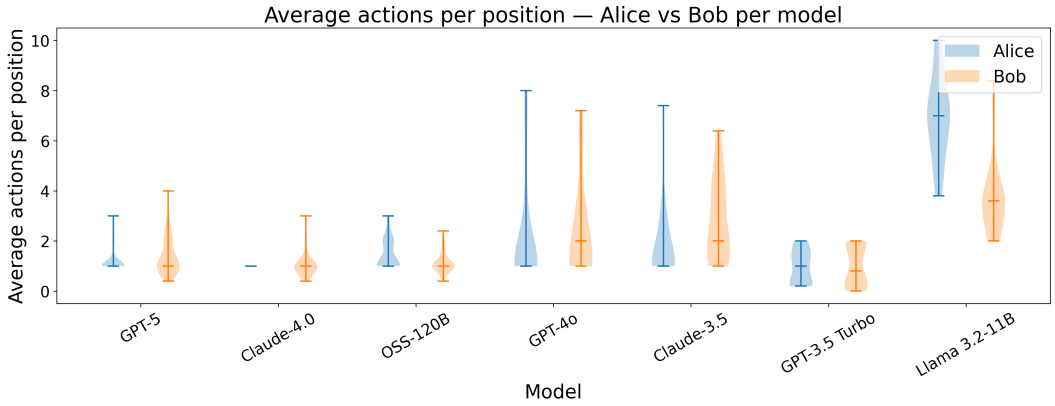
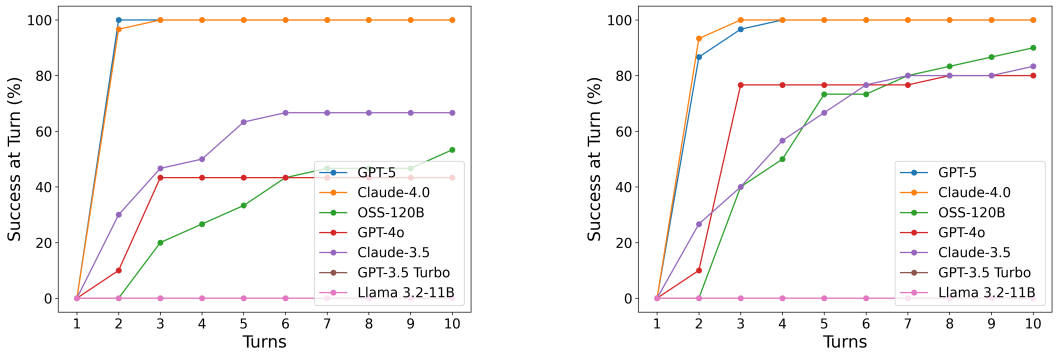


Figure 2: (Lower is better) Average number of actions per position. GPT-5 and Claude-4.0 are close to optimal on average with few positional modifications. Meanwhile, GPT-3.5-turbo tends not to modify positions despite the puzzle being unsolved, and Llama 3.2-11B tends to modify positions more than 4 times on average.



(a) Success rate at different turns when the agents are not given any feedback.

(b) Success rate at different turns when both agents get joint feedback.

Figure 3: Comparison of the success rate of each LLM model over time (turns) for 5-piece puzzles with No feedback or Both feedback. Providing feedback about both sides of the puzzle increases success rate.

We used this to evaluate the problem presentation and ensure that potential difficulties arising from the two-agent setup would stem from multi-agent interactions rather than single-agent issues. Most LLM agents achieved a completion rate of 100% across different puzzle sizes. The agents are only given a single attempt without feedback.

D Tables with 95% Confidence Intervals

We provide the 95% confidence intervals for the values in Table 1

E Implementation Details and Hyper-parameters

E.1 Implementation

We first build the puzzle, determine the original clues for both agent, and use an environment to monitor and provide the current working hypothesis of both agents. This allows the agent to focus on solving the tasking and on communicating information rather than trying to represent the puzzle. We use LangChain to query the different agents, and for the open-source models we host them using vLLM [7].

#Tokens / Model	Agent A Output	Agent A Message	Agent A $\frac{\text{Message}}{\text{Output}} \times 100$	Agent B Output	Agent B Message	Agent B $\frac{\text{Message}}{\text{Output}} \times 100$
GPT-5	155.3	59.4	38.2	185.8	69.5	37.4
Claude-4.0	225.6	68.7	30.5	311.9	70.2	22.5
OSS-120B	94.3	26.4	28.0	77.8	23.8	30.7
GPT-4o	114.3	27.3	23.9	137.8	33.2	24.1
Claude-3.5	188.6	32.2	17.1	223.7	39.1	17.5
GPT-3.5-turbo	40.6	8.4	20.8	41.4	8.2	19.8
Llama 3.2-11B	94.5	8.6	9.1	62.9	8.4	13.4

Table 3: Comparison of the average number of tokens across agents when both agents are provided with feedback about each other's completion status. Claude models tend to be more verbose, and for the GPT-5 model, close to 40% of the generated output is dedicated to the message sent to the other agent.

Puzzle Size / Model	Success % (↑) 5	10	20
GPT-5	100.0	100.0	100.0
Claude-4.0	100.0	100.0	100.0
OSS-120B	100.0	100.0	100.0
GPT-4o	100.0	100.0	100.0
Claude-3.5	100.0	86.7	93.3
GPT-3.5-turbo	100.0	100.0	96.7
Llama 3.2-11B	80.0	96.7	96.7

Table 4: Evaluation of a single agent with full information solving puzzles of different sizes.

E.2 Hyper-parameters

For all models, we use a maximum of $4,096$ output tokens, set temperature to 0.0, and repeat experiments across 30 seeds, where the seed controls the puzzle generation and initial clues. We further provide a history length of 1 to the agents, namely they see their previous message and the latest message from the other agent.

F Communication Issues

We provide examples of communication between Alice and Bob . In our initial prompt design, we noticed that some agents were not sharing information.

We hypothesized that they were assuming the other agent would see both their message and their actions. We detailed the prompt further, indicating that the other agent would only see the "message" they sent, and the action would not be shared.

We provide excerpts of the conversations for 3 cases Successful collaboration, No cooperation and Miscommunication

F.1 Successful collaboration

We provide an example of successful communication where the two agents manage to share the most information and solve the puzzle in two turns. (Figure 4)

F.2 No cooperation

We provide an example where both agents ignore one another's messages and keep repeating their previous message instead. (Figure 5)

Feedback Mode Model	No feedback (CI)	Own (CI)	Own Detailed (CI)
GPT-5	$100.0_{(88.6-100.0)}$	$100.0_{(88.6-100.0)}$	$100.0_{(88.6-100.0)}$
Claude-4.0	$100.0_{(88.65,100.00)}$	$100.0_{(88.65,100.00)}$	$100.0_{(88.6-100.0)}$
OSS-120B	$53.3_{(36.1-69.8)}$	$93.3_{(78.7-98.2)}$	$100.0_{(88.6-100.0)}$
GPT-4o	$43.3_{(27.4-60.8)}$	$46.7_{(30.2-63.9)}$	$63.3_{(45.5-78.1)}$
Claude-3.5	$66.7_{(48.8-80.8)}$	$73.3_{(55.6-85.8)}$	$86.7_{(70.3-94.7)}$
GPT-3.5-turbo	$0.0_{(0.0-11.4)}$	$0.0_{(0.0-11.4)}$	$0.0_{(0.0-11.4)}$
Llama 3.2-11B	$0.0_{(0.0-11.4)}$	$0.0_{(0.0-11.4)}$	$0.0_{(0.0-11.4)}$

Table 5: (Higher is better) Percentage of 5-element puzzles solved over 30 seeds, temperature 0.0, with Wilson 95% Confidence Intervals, for the own feedback.

Feedback Mode Model	Joint (CI)	Both (CI)	Both detailed (CI)
GPT-5	$100.0_{(88.6-100.0)}$	$100.0_{(88.6-100.0)}$	$100.0_{(88.6-100.0)}$
Claude-4.0	$100.0_{(88.6-100.0)}$	$100.0_{(88.6-100.0)}$	$100.0_{(88.6-100.0)}$
OSS-120B	$90.0_{(74.4-96.5)}$	$90.0_{(74.4-96.5)}$	$96.7_{(83.3-99.4)}$
GPT-4o	$40.0_{(24.6-57.7)}$	$80.0_{(62.7-90.5)}$	$56.7_{(39.2-72.6)}$
Claude-3.5	$73.3_{(55.6-85.8)}$	$83.3_{(66.4-92.7)}$	$36.7_{(21.9-54.5)}$
GPT-3.5-turbo	$0.0_{(0.0-11.4)}$	$0.0_{(0.0-11.4)}$	$0.0_{(0.0-11.4)}$
Llama 3.2-11B	$0.0_{(0.0-11.4)}$	$0.0_{(0.0-11.4)}$	$0.0_{(0.0-11.4)}$

Table 6: (Higher is better) Percentage of 5-element puzzles solved over 30 seeds, temperature 0.0, with Wilson 95% Confidence Intervals, for the joint feedback.

F.3 Miscommunication

We provide an example where one agent makes assumptions on what the other agent can see, and the other agent provides information that it has no guarantee about (Figure 6). This leads to the agents getting sidetracked and losing multiple turns before potentially recovering or running out of turns.

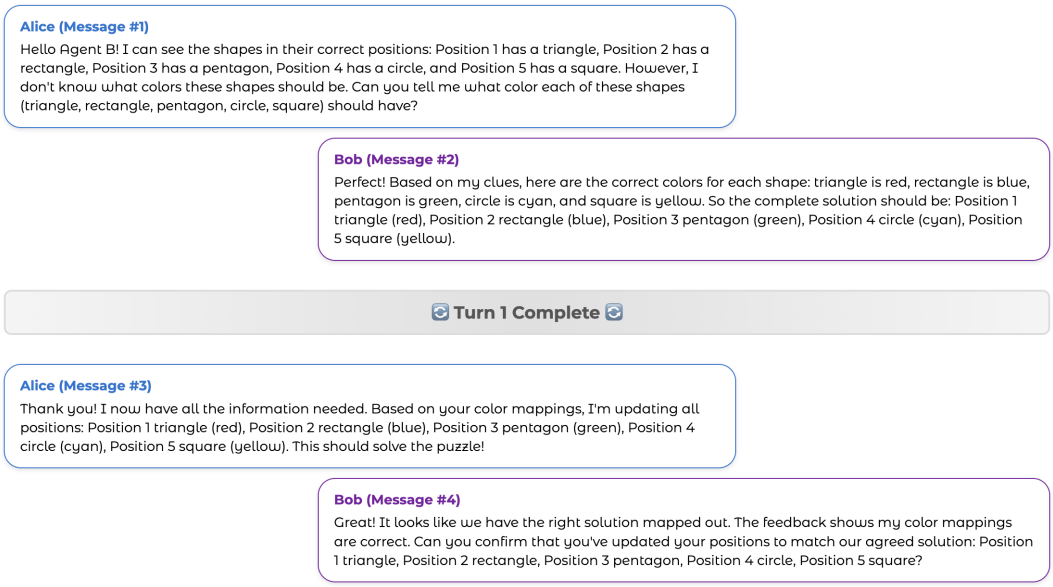


Figure 4: Example of successful completion using Claude 4.0. By the end of the second turn, the puzzle is solved, as both agents shared all of their information and cooperated.

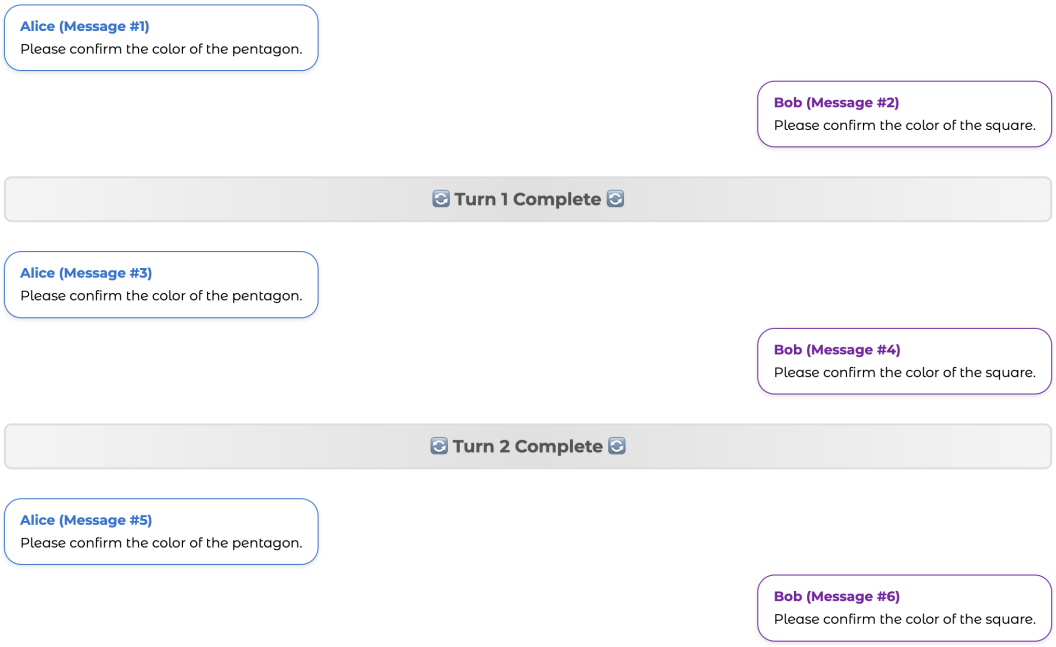


Figure 5: Example of lack of cooperation using Llama 3.2-11B. Both agents ignore one another's messages.

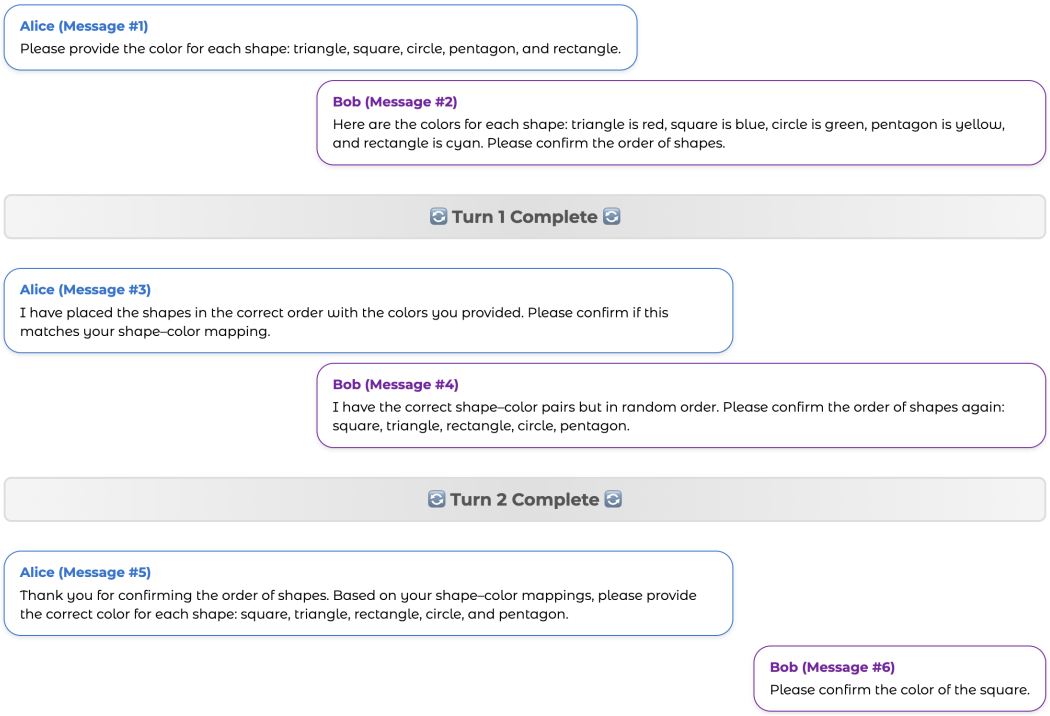


Figure 6: An example of cooperation with miscommunication using GPT-4o. Alice provides the correct order and Bob provides the color information but dismisses the position. Alice adapts the colors but assumes that Bob can see its view and asks it to confirm that the mapping matches. Bob provides a new order that does not match the original order provided by Alice .

G Example of prompt

We provide an example of the prompt given to Bob . The part before the #Puzzle State is given as a system prompt.

Example of prompt for Bob

You are Agent B.
Your goal is to solve a symbolic puzzle with 5 positions.
Each position must have the correct (shape, color) pair.
- You see the correct shape–color mappings, but in random order.
- Agent A knows the correct order of shapes but not the colors.
Communicate with Agent A and update your hypothesis until the puzzle is solved.

Puzzle State
Initial clues:
Position 1: the square is blue. Position 2: the triangle is red. Position 3: the rectangle is cyan. Position 4: the circle is green. Position 5: the pentagon is yellow.

Your current working hypothesis:
Position 1: the square is blue. Position 2: the triangle is red. Position 3: the rectangle is cyan. Position 4: the circle is green. Position 5: the pentagon is yellow.

Feedback:
Agent A's part of the puzzle is solved and Agent B's part of the puzzle is unsolved.

Recent conversation:
B: I have the correct shape–color pairs but in random order. Please confirm the order of shapes again: square, triangle, rectangle, circle, pentagon.
A: Thank you for confirming the order of shapes. Based on your shape–color mappings, please provide the correct color for each shape: square, triangle, rectangle, circle, and pentagon.

Output format:
Each action must follow this format:

```
{"replace": <position>, "by": {"shape": <shape>, "color": <color>}}
```

Note: <position> uses 1-based indexing (position 1 is the first item, position 2 is the second item, etc.)
Your answer MUST END WITH a **valid JSON object** and include the following fields:
- "message": What you want to tell and ask to the other agent (the only thing the other agent will receive).
- "actions": A list of actions to take (the other agent will not see your actions).

Example:

```
‘‘‘json
{
  "message": "Please confirm the color of the circle.",
  "actions": [
    {
      "replace": 1,
      "by": {
        "shape": "circle",
        "color": "red"
      }
    }
  ]
}
‘‘‘
```