LLM-based Multi AI Agents Collaboration in VR Scenarios

Sissy Chen
Student ID: 260601294
School of Computer Science
xi.chen20@mail.mcgill.ca

Ricky Liu
Student ID: 260927798
Faculty of Engineering
li.qi.liu@mail.mcgill.ca

Yili Jin
Student ID: 261148049
School of Computer Science
yili.jin@mail.mcgill.ca

Abstract

This project investigates the potential of using Large Language Model (LLM)-based AI agents to enhance collaboration in Virtual Reality (VR) environments. It aims to develop a framework for collaboration between LLM-based AI agents in VR scenarios. The VR environment is the ThreeDWorld Transport Challenge platform, in which agents work together to transport objects using containers. The AI agent comprises an observation module, an action module, and an LLM-based brain with a Communication Module and a Reasoning Module. The Communication Module generates effective messages using LLMs, while the Reasoning Module leverages LLMs for decision-making and high-level planning. The project incorporates recent work and extends it by introducing a third agent and improving the prompting design. This project serves as an invaluable opportunity for us to enhance our expertise in the field of NLP. The code can be found in *https://github.com/yilijin/COMP550-Final*

1 Introduction

In recent years, the rapid advancements in AI and VR, including the emergence of the Metaverse, have ushered in a new era of possibilities for collaboration and interaction between humans and AI agents. However, achieving effective collaboration between AI agents, as well as between AI agents and humans in VR scenarios, continues to pose significant challenges.

Large Language Models (LLMs) have demonstrated their remarkable capabilities across diverse domains. Recent research has further highlighted the potential of LLMs to function as planners in single-agent tasks, as evidenced by their successful implementation in zero-shot prompting for instruction following tasks (Huang et al., 2022). Additionally, Park et al. (Park et al., 2023) have shown that generative AI agents can interact with each other in a manner reminiscent of human interaction, further illustrating the collaborative potential of LLMs.

This project aims to investigate the potential of leveraging LLM-based AI agents to enhance collaboration and communication in VR environments. The goal is to develop a framework that enables seamless collaboration between LLM-based AI agents in VR scenarios.

The VR environment is based on the ThreeDWorld Transport Challenge platform, which allows agents to collaborate in transporting objects using containers. The AI agent consists of an observation module and an action module, with the focus being on the design of the LLM-based brain. The LLM-based brain consists of a Communication Module and a Reasoning Module. The Communication Module uses LLMs to generate messages for effective communication among agents, considering task instructions, goals, state descriptions, action history, and dialogue history. The Reasoning Module utilizes LLMs to make decisions on actions by retrieving information from the Memory Module, compiling an Action List, and prompting the LLMs to generate high-level plans using current information and the Action List. The chain-of-thought prompting technique is employed to encourage thorough reasoning before providing final answers.

Our work builds upon the research presented in the paper titled *Building Cooperative Embodied Agents Modularly with Large Language Models* (Zhang et al., 2023). This paper, which was recently accepted by Conference on Neural Information Processing Systems (NeurIPS) 2023, serves as the foundation for our own investigation. However, while the original study focused on examining the interaction between two agents, we have expanded upon their work by introducing a third agent into the system. Additionally, we have made modifications and enhancements to the prompting design, further improving upon the existing framework.

In this project, our first objective was to replicate the results presented in a recent paper published at a top conference. Subsequently, we aimed to build upon those findings by expanding the scope and applicability of the research. Throughout this endeavor, we gained valuable insights into the development and implementation of LLM-based AI agents. Overall, this project served as an invaluable opportunity for us to enhance our expertise in the field of NLP.

2 Related Work

The field of LLM-based AI agents has gained significant attention in the past year, with numerous studies focusing on leveraging large language models (LLMs) to achieve human-like intelligence and decision-making capabilities. In this section, we provide an overview of the related work in this field.

Previous research in AI agents primarily focused on training agents with limited knowledge in isolated environments, which deviated from the human learning process and hindered the agents' ability to make human-like decisions (Mnih et al., 2015; Lillicrap et al., 2016). However, the emergence of LLMs, which have demonstrated remarkable potential in achieving human-level intelligence through the acquisition of vast amounts of web knowledge (Touvron et al., 2023a,b), has led to a surge of studies investigating LLM-based agents.

The use of LLMs as central controllers to construct AI agents has been a key direction in this field (Shen et al., 2023; Shinn et al., 2023). Researchers have developed various models and frameworks to equip LLMs with human-like capabilities such as memory and planning, enabling them to effectively complete tasks.

One significant aspect of LLM-based AI agent construction is the design of the agent architecture to leverage LLMs effectively. Previous work has focused on developing modules to bridge the gap between traditional language models and AI agents. (Chen et al., 2019) propose a unified framework consisting of a profiling module, a memory module, a planning module, and an action module. The profiling module identifies the role of the agent, while the memory and planning modules enable the agent to recall past behaviors and plan future actions. The action module translates the agent's decisions into specific outputs.

In addition to agent architecture design, several studies have explored strategies for agent capability acquisition within the LLM-based AI agent framework (Schick et al., 2023). These strategies involve fine-tuning LLMs and enabling agents to acquire specific capabilities for accomplishing tasks effectively. The choice of capability acquisition strategies depends on the specific application scenarios and the desired agent behavior.

In summary, the field of LLM-based AI agents has witnessed significant progress, driven by the capabilities of LLMs and the desire to achieve human-like decision-making. Previous studies have explored agent architecture design, capability acquisition strategies, application domains, and evaluation methods. However, there is still much room for further research and development in this rapidly-evolving field.

3 Methodology

Figure 1 provides an overview of the proposed system framework. All agents are able to communicate within a VR environment, which will be described in Section 3.1. A detailed description of our agent's design will be provided in Section 3.2.

3.1 VR Environment

To facilitate our experiments, we utilize the ThreeDWorld Transport Challenge platform (Gan et al., 2022), which is built on the ThreeDWorld (TDW) platform (Gan et al., 2021), an open-source VR simulator. It involves the task of transporting a small set of objects scattered around a house using containers. Our objective is to achieve collaborative completion of this task by leveraging LLM-based communication and planning.

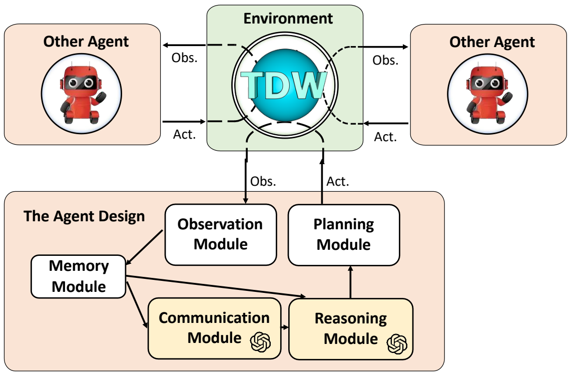


Figure 1: An Overview of the System Framework.

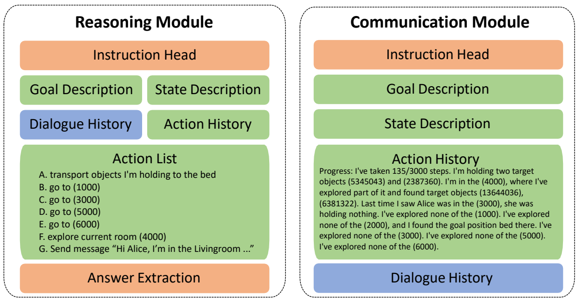


Figure 2: Reasoning and Communication Module.

3.2 AI Agent

The AI Agent consists of an observation module for receiving inputs from the environment and an action module for generating outputs to interact with the environment. While this aspect is not the primary focus of our project, the TDW platform will provide the agent with semantic information based on its observations. The agent can be guided by Python instructions.

The primary objective of our project is centered around the design of the LLM-based brain for our agent. This design is crucial as it governs the agent's ability to plan actions and effectively communicate with other agents. The key considerations include planning the actions to be performed by the agent and facilitating communication with other agents or humans. There are two main modules: the Communication Module and the Reasoning Module (see Figure 2).

3.2.1 Communication Module

It is important for cooperative embodied agents to be able to communicate effectively with others. Effective communication needs to solve two problems: what to send and when to send it. In this module, we address the "what to send" problem by directly using the LLMs as a Message Generator with designed prompts. These prompts are constructed from the components of the Instruction Head, Goal Description, State Description, Action History, and Dialogue History. To better constrain the LLMs' generated messages, we also add a note at the end of the prompt and append two seed messages at the beginning of the Dialogue History to elicit effective communication behavior. The detailed prompt design is shown below:

- **Instruction Head**: This part of the prompts is fixed for an environment, mainly consisting of the task instructions and environmental constraints.
- **Goal Description**: For each task, the goal description is converted from $G = \{g_1, g_2, ..., g_k\}$ using a formal template.
- **State Description**: For each step, the state description is converted from task progress, state of self, state of others, and semantic map retrieved from the Memory Module through a template.
- **Action History**: The concatenation of the last K actions (high-level plans) the agent took.
- **Dialogue History**: The concatenation of the last D dialogues between agents, including the messages sent by the agent itself.

To constrain the message generation done by the LLMs, we add a note at the end of the prompt:

> *Note: The generated message should be accurate, helpful, and brief. Do not generate repetitive messages.*

We also append two seed messages at the beginning of the Dialogue History to elicit effective communication behavior:

> *"Hi, I'll let you know if I find any goal objects, finish any subgoals, and ask for your help when necessary."*

3.2.2 Reasoning Module

The agent requires a strong Reasoning Module to make decisions on which action to take, utilizing all available information gathered and stored so far to maximize cooperation efficiency. We utilize powerful LLMs directly as the Reasoning Module.

This is achieved by first retrieving the related information from the Memory Module and converting it into text descriptions, similar to the Communication Module. We then compile an Action List of all available high-level plans proposed according to the current state and the procedural knowledge stored for the LLMs to make the choice. This formalization makes it easier for the LLMs to focus on the reasoning and create an executable plan without the need for few-shot demonstrations. Finally, we prompt the LLMs with current information and the proposed Action List to generate a high-level plan. We also use the idea of chain-of-thought prompting technique (Wei et al., 2022) (which is the Reading Assignment 4) to encourage the LLMs to perform more reasoning before providing the final answer.

Action List: We compile all available actions regarding the current state into an Action List for the LLMs to select from. This multi-choice formalization makes it easier for the LLM to create an executable plan without any few-shot demonstrations. The available high-level plans for our VR environment include:

- go to room *
- explore current room
- go grasp target object/container *
- put holding objects into the holding container
- transport holding objects to the bed
- send a message: "*"

Answer Extraction: As demonstrated by Wei et al. (Wei et al., 2022), the LLMs' strong reasoning ability can be unlocked by the chain-of-thought prompting technique. Therefore, we employ this prompting technique to motivate the LLMs in engaging in further reasoning before providing the final answer.

4 Experiments

4.1 Setup Details

Six scenes from the TDW-House dataset were selected to constitute the test set. Each scene is composed of 6 to 8 rooms, 10 objects, and a few containers. Two types of tasks and two settings were sampled in each of the scenes, forming a test set of 24 episodes. An episode is terminated if all the target objects have been transported to the goal position or the maximum number of frames has been reached.

The tasks are named 'food task' and 'stuff task'. Containers for the 'food task' can be found in both the kitchen and living room, while containers for the 'stuff task' can be found in the living room and office.

The configuration and distribution of containers vary based on two distinct settings: the 'Enough Container Setting' and the 'Rare Container Setting'. In the 'Enough Container Setting', the ratio of containers to objects stands at $1:2$, and containers associated with a specific task are located in no more than two rooms. On the other hand, in the 'Rare Container Setting', the container-to-object ratio decreases to $1:5$. This distribution differs from the 'Enough Container Setting' as containers in the 'Rare Container Setting' are strictly localized to a single room.

In a multi-agent setting, agents may take different numbers of frames to successfully or unsuccessfully complete a single action. Once an environment step is finished, which is indicated by one of the agents completing their action, the current observation is returned to all agents. All agents are then asked for a new action and agents with ongoing actions will directly switch to the new action if it differs from their current one.

Using Python3, six experiments were run using the testing data. The parameters for each experiment consisted of the following:

- **agents**: The number of agents and the type of each agent (either AI or human)

- **prompt_template_path**: The path to the CSV file specifying the prompt template
- **lm_id**: The LLM (either gpt-3.5-turbo or gpt-4)
- **max_frames**: The number of frames
- **data_prefix**: The path to the data

Each experiment begins by resetting the environment. This is done with the instruction head, which specifies the following:

- a scene $\in \{1a, 2a, 3a, 4a, 5a\}$
- a layout $\in \{0\_0, 0\_1, 1\_0, 1\_1, 2\_0, 2\_1\}$
- a seed
- a task $\in$ {food, stuff}

Each experiment attempts to complete the specified task within the specified maximum number of frames. Each step of each episode is defined by a series of actions and by the frame $f \in N$ at which the step concludes, where $f \leq$ max_frames. Each action is identified by an integer $i \in W$ defining its execution order and by a type variable describing the nature of the action. An experiment runs until either all 24 episodes are terminated or the maximum frames has been reached.

4.2 Demo

The demo of 6 sequentially-run experiments lasts approximately 1 hour. In the demo, we initialize the agents and the setting, and run a specified GPT prompt that initializes the action procedure for the agents in the experiment. The output of the prompt is then translated into instructions for the agents, at each frame.

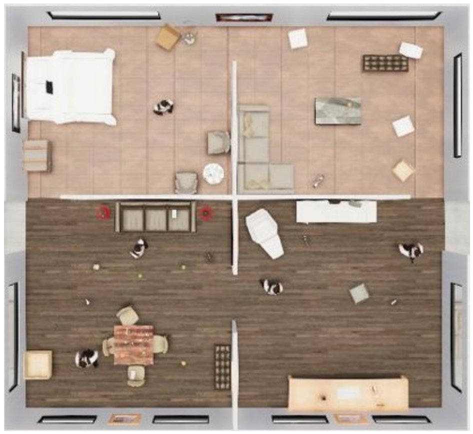
During the demo, the experiments run in a Unity simulation. Every frame is saved for documentation purposes, and a few are displayed below. Motivated by the underlying GPT prompts, the agents change positions frame-by-frame to complete their task. The communication between the agents are logged into a file.

Below is an example of prompt, which initializes the action procedure for the experiment's agents:

```
{prompts: "I'm Alice. I want to transport as
    many target objects as possible to the bed
    with the help of containers within 3000
    steps. I can hold two things at a time, and
    they can be objects or containers. I can
    grasp containers and put objects into them
    to hold more objects at a time. Given my
    goal, progress, and previous actions,
    please help me choose the best available
    action to achieve the goal as soon as
    possible. Note that a container can contain
    three objects, and will be lost once
    transported to the bed. I can only put
    objects into the container I hold after
    grasping it. All objects are denoted as
    <name> (id), such as <table> (712). Actions
    cost several steps. It may be costly to go
    to another room or transport to the bed,
    use these actions sparingly.\nGoal:
    Transport 2 oranges, 3 apples, 1 banana, 3
    breads, 1 burger to the bed.\nProgress:
    I've taken 0/3000 steps. I haven't found
    the goal position bed. I'm holding nothing.
    I'm in the <Office> (3000), where I've
    explored part of it. I've explored none of
    the <Livingroom> (1000). I've explored none
    of the <Bedroom> (2000). I've explored none
    of the <Livingroom> (4000). I've explored
    none of the <Kitchen> (5000). I've explored
    none of the <Livingroom> (6000). \nPrevious
    actions: go to <Office> (3000) at initial
    step\nAvailable actions: (You can only
    choose the action in the list)\nA. go to
    <Livingroom> (1000)\nB. go to <Bedroom>
    (2000)\nC. go to <Livingroom> (4000)\nD. go
    to <Kitchen> (5000)\nE. go to <Livingroom>
    (6000)\nF. explore current room <Office>
    (3000)\n\nAnswer: Let's think step by
    step.", outputs: ['F. explore current room
    <Office> (3000)']}
```

Each demo is run for 30 or so frames. We observe that it is not enough for the agents to complete their tasks, but from frame-to-frame, the agents move.

(a) Frame 0



(b) Frame 15

Figure 3: Frame 0 and Frame 15 within an experiment, from a top-down view. Note the people (agents) moving.

4.3 Numerical Results

We conducted experiments to evaluate the performance of different numbers of agents using various LLMs. In order to provide supporting evidence, we compare our results with those from a previous study (Zhang et al., 2023). The metric is *Transport Rate (TR)*: The fraction

	Two Agents (Zhang et al., 2023)		Two Agents (Ours)		Three Agents (Ours)		
	GPT4 * 2	LLAMA2 * 2	GPT4 * 2	GPT3.5 * 2	GPT4 * 3	GPT3.5 * 3	GPT4 * 2 + GPT3.5
Food	0.82	0.57	0.84	0.73	0.88	0.79	0.85
Stuff	0.61	0.48	0.61	0.52	0.67	0.60	0.66
Total	0.71	0.53	0.72	0.62	0.78	0.74	0.76

Table 1: The transportation rate of various numbers of agents with different LLMs was evaluated, and the results from (Zhang et al., 2023) are provided as supporting evidence.

of the target objects successfully transported to the goal position. Table 1 presents the results.

Examining the results, we observe that for both the two-agent and three-agent scenarios, our approach with GPT4 consistently outperforms the previous study's approach using GPT4. In the two-agent scenario, our GPT4 achieves a transportation rate of 0.84 for food and 0.61 for stuff, compared to 0.82 and 0.57 respectively in the previous study. Similarly, in the three-agent scenario, our GPT4 achieves a transportation rate of 0.88 for food and 0.67 for stuff, while the previous study reported rates of 0.79 and 0.60 respectively for GPT3.5.

When comparing results using different LLMs, we find that GPT4 consistently outperforms GPT3.5. For instance, in the two-agent scenario, GPT4 achieves a transportation rate of 0.84 for food, whereas GPT3.5 achieves a slightly lower rate of 0.73. Similarly, in the three-agent scenario, GPT4 achieves a higher transportation rate of 0.88 for food compared to 0.79 for GPT3.5.

Overall, our experiments demonstrate that our approach with GPT4 yields improved transportation rates compared to the previous study's approach using GPT4, as well as outperforming GPT3.5 in both scenarios. These results showcase the effectiveness of our approach in enhancing the capabilities of multiple agents.

5 Discussion

While our work demonstrates promising results in enhancing collaboration between multiple LLM-based agents, there are still several limitations that need to be addressed in future work:

5.1 Task and Environment Complexity

The VR environment and tasks used in our experiments were relatively simple, involving transporting objects between rooms. In reality, real-world collaborative tasks tend to be more complex, with dynamic environments and uncertain outcomes. Future work should explore more complex scenarios to evaluate the scalability and robustness of the approach.

5.2 Model and Prompt Design

The current prompt design and reasoning module focus on high-level planning and communication. Finer-grained behaviors and interactive skills need to be developed. The models can also be improved by incorporating capabilities like physical intuition and emotional intelligence. More sophisticated prompting techniques may help unlock the full potential of LLMs.

5.3 Generalization to New Tasks

The models are specialized for a fixed set of tasks in confined environments. Their ability to generalize and adapt to completely new tasks and environments remains untested. Future work should evaluate zero-shot and few-shot learning abilities of the agents. Continual learning methods may help improve generalization.

5.4 Human Evaluation

The evaluation is currently based on objective metrics like success rates. A user study involving human subjects should be conducted to assess key qualities like naturalness, safety and trustworthiness from a human perspective. This provides more realistic and comprehensive evaluation.

5.5 Scaling to Larger Teams

The current experiments only involve up to several agents. Real-world scenarios may require coordination between tens or hundreds of individuals. The approach needs to be scaled and its efficiency evaluated with increasing team size.

6 Conclusion

This project investigated the potential of using LLM-based AI agents for collaboration in virtual reality environments. An LLM-driven framework was proposed consisting of communication and reasoning modules to enable coordination between agents. Experiments were conducted in a VR transport task involving multiple agents. Results demonstrated that the approach improved task efficiency compared to previous work, and that larger agent teams and more powerful LLMs led to better performance. However, several limitations were also identified, including the need for more complex scenarios, advanced agent modeling, generalization testing, human evaluation, and scaling to larger group sizes. Overall, the work provided insights into leveraging LLMs for multi-agent collaboration.

7 Statement of Contributions

Yili took the lead on coding while Sissy and Ricky conducted several experiments. For the report writing, Yili led the overall effort and Sissy and Ricky wrote sections 4.1 and 4.2 respectively. It should be noted that the division of labor was not rigid; we discussed and collaborated on all aspects of the project throughout.

References

Xinshi Chen, Shuang Li, Hui Li, Shaohua Jiang, Yuan Qi, and Le Song. 2019. Generative adversarial user model for reinforcement learning based recommendation system. In *Proceedings of the 36th International Conference on Machine Learning, ICML 2019, 9-15 June 2019, Long Beach, California, USA*, volume 97 of *Proceedings of Machine Learning Research*, pages 1052–1061. PMLR.

Chuang Gan, Jeremy Schwartz, Seth Alter, Damian Mrowca, Martin Schrimpf, James Traer, Julian De Freitas, Jonas Kubilius, Abhishek Bhandwaldar, Nick Haber, Megumi Sano, Kuno Kim, Elias Wang, Michael Lingelbach, Aidan Curtis, Kevin T. Feigelis, Daniel Bear, Dan Gutfreund, David D. Cox, Antonio Torralba, James J. DiCarlo, Josh Tenenbaum, Josh H. McDermott, and Dan Yamins. 2021. Threedworld: A platform for interactive multi-modal physical simulation. In *Proceedings of the Neural Information Processing Systems Track on Datasets and Benchmarks 1, NeurIPS Datasets and Benchmarks 2021, December 2021, virtual*.

Chuang Gan, Siyuan Zhou, Jeremy Schwartz, Seth Alter, Abhishek Bhandwaldar, Dan Gutfreund, Daniel L. K. Yamins, James J. DiCarlo, Josh H. McDermott, Antonio Torralba, and Joshua B. Tenenbaum. 2022. The threedworld transport challenge: A visually guided task-and-motion planning benchmark towards physically realistic embodied AI. In *2022 International Conference on Robotics and Automation, ICRA 2022, Philadelphia, PA, USA, May 23-27, 2022*, pages 8847–8854. IEEE.

Wenlong Huang, Pieter Abbeel, Deepak Pathak, and Igor Mordatch. 2022. Language models as zero-shot planners: Extracting actionable knowledge for embodied agents. In *International Conference on Machine Learning, ICML 2022, 17-23 July 2022, Baltimore, Maryland, USA*, volume 162 of *Proceedings of Machine Learning Research*, pages 9118–9147. PMLR.

Timothy P. Lillicrap, Jonathan J. Hunt, Alexander Pritzel, Nicolas Heess, Tom Erez, Yuval Tassa, David Silver, and Daan Wierstra. 2016. Continuous control with deep reinforcement learning. In *4th International Conference on Learning Representations, ICLR 2016, San Juan, Puerto Rico, May 2-4, 2016, Conference Track Proceedings*.

Volodymyr Mnih, Koray Kavukcuoglu, David Silver, Andrei A Rusu, Joel Veness, Marc G Bellemare, Alex Graves, Martin Riedmiller, Andreas K Fidjeland, Georg Ostrovski, et al. 2015. Human-level control through deep reinforcement learning. *nature*, 518(7540):529–533.

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

Timo Schick, Jane Dwivedi-Yu, Roberto Dessì, Roberta Raileanu, Maria Lomeli, Luke Zettlemoyer, Nicola Cancedda, and Thomas Scialom. 2023. Toolformer: Language models can teach themselves to use tools. *CoRR*, abs/2302.04761.

Yongliang Shen, Kaitao Song, Xu Tan, Dongsheng Li, Weiming Lu, and Yueting Zhuang. 2023. Hugginggpt: Solving AI tasks with chatgpt and its friends in huggingface. *CoRR*, abs/2303.17580.

Noah Shinn, Federico Cassano, Ashwin Gopinath, Karthik R Narasimhan, and Shunyu Yao. 2023. Reflexion: Language agents with verbal reinforcement learning. In *Thirty-seventh Conference on Neural Information Processing Systems*.

Hugo Touvron, Thibaut Lavril, Gautier Izacard, Xavier Martinet, Marie-Anne Lachaux, Timothée Lacroix, Baptiste Rozière, Naman Goyal, Eric Hambro, Faisal Azhar, Aurélien Rodriguez, Armand Joulin, Edouard Grave, and Guillaume Lample. 2023a. Llama: Open and efficient foundation language models. *CoRR*, abs/2302.13971.

Hugo Touvron, Louis Martin, Kevin Stone, Peter Albert, Amjad Almahairi, Yasmine Babaei, Nikolay Bashlykov, Soumya Batra, Prajjwal Bhargava, Shruti Bhosale, Dan Bikel, Lukas Blecher, Cristian Canton-Ferrer, Moya Chen, Guillem Cucurull, David Esiobu, Jude Fernandes, Jeremy Fu, Wenyin Fu, Brian Fuller, Cynthia Gao, Vedanuj Goswami, Naman Goyal, Anthony Hartshorn, Saghar Hosseini, Rui Hou, Hakan Inan, Marcin Kardas, Viktor Kerkez, Madian Khabsa, Isabel Kloumann, Artem Korenev, Punit Singh Koura, Marie-Anne Lachaux, Thibaut Lavril, Jenya Lee, Diana Liskovich, Yinghai Lu, Yuning Mao, Xavier Martinet, Todor Mihaylov, Pushkar Mishra, Igor Molybog, Yixin Nie, Andrew Poulton, Jeremy Reizenstein, Rashi Rungta, Kalyan Saladi, Alan Schelten, Ruan Silva, Eric Michael Smith, Ranjan Subramanian, Xiaoqing Ellen Tan, Binh Tang, Ross Taylor, Adina Williams, Jian Xiang Kuan, Puxin Xu, Zheng Yan, Iliyan Zarov, Yuchen Zhang, Angela Fan, Melanie Kambadur, Sharan Narang, Aurélien Rodriguez, Robert Stojnic, Sergey Edunov, and Thomas Scialom. 2023b. Llama 2: Open foundation and fine-tuned chat models. *CoRR*, abs/2307.09288.

Jason Wei, Xuezhi Wang, Dale Schuurmans, Maarten Bosma, Brian Ichter, Fei Xia, Ed H. Chi, Quoc V. Le, and Denny Zhou. 2022. Chain-of-thought prompting elicits reasoning in large language models. In *NeurIPS*.

Hongxin Zhang, Weihua Du, Jiaming Shan, Qinhong Zhou, Yilun Du, Joshua B Tenenbaum, Tianmin Shu, and Chuang Gan. 2023. Building cooperative embodied agents modularly with large language models. *arXiv preprint arXiv:2307.02485*.