LLMs for Robotic Manipulation in AR Environments

Cedric Bone
Golisano College of Computing and Information Sciences
Rochester Institute of Technology
Rochester, U.S.A
cb9017@rit.edu

***Abstract*—This paper presents an approach to controlling a robotic arm in an augmented reality (AR) environment using large language models (LLMs). By using natural language as an input through voice transcription and subsequent interpretation by LLMs, the method simplifies command input for complex robotic tasks. The prototype uses a Unity-based AR scene, Whisper audio transcription, and OpenAI's language models to interpret user instructions and help generate structured commands for a virtual robotic arm to perform pick-and-place operations. In addition, this work evaluates the robustness of the pipeline to different models in order to test the interpreting capabilities of various OpenAI models with hard-to-interpret user commands. Limitations and possible future directions, including the development of an action model are also discussed.**

***Index Terms*—Large Language Models, Augmented Reality, Virtual Reality, Robotic Arm Control, Natural Language Processing, Human-Robot Interaction**

I. Introduction

Recent advancements in model architecture and multimodal learning have transformed artificial intelligence systems from passive tools into dynamic collaborators, allowing for more natural human-robot interaction (HRI) [1]. Traditional robotic control methods typically use complex interfaces or manually programmed scripts [2] which can be challenging for users without domain expertise. LLMs offer a solution by understanding and generating natural language, allowing users to convey high-level tasks to robots without specialized training.

In this study, an LLM-based voice command system is integrated into an AR environment developed in Unity. The AR scene allows users to issue voice commands and see a virtual robot arm's actions in real time. Spoken commands are transcribed using Whisper, and the text is then interpreted by an LLM. The LLM output is then transformed into actionable, structured commands parsed and executed by a robotic arm simulation in Unity using inverse kinematics (IK) and preferred actions. Because the environment is fully virtual, object positions are known simplifying the IK. In a real-world scenario, a vision model would be needed to localize objects.

This LLM parsing approach reduces the complexity of giving instructions. Users can, for example, state: "Move the orange block to the yellow platform," and the system interprets and executes this request without the user requiring knowledge of kinematics, manual controls, or writing a script. This technique not only makes HRI more accessible but also expands the feasibility of deploying robots in human-centered environments. Additionally, this technique enables capabilities beyond handling simple commands, allowing for the interpretation of ambiguous commands.

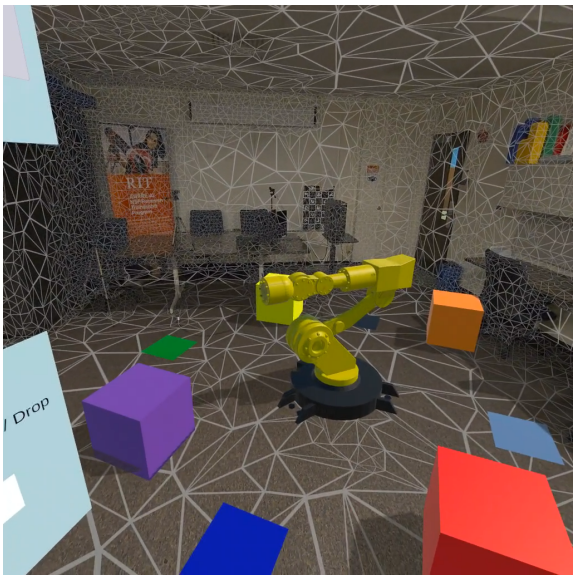

Fig. 1. The simulated robotic arm in its home position at the center of the AR environment.

To further evaluate the interpretive robustness of models in this pipeline a separate evaluation was conducted. Multiple OpenAI models were assessed on their ability to infer intended actions from vague language, an important task when robotics are used in a dynamic human-centered environment. This secondary evaluation acts as a benchmark to identify which OpenAI model is the most performant for this use case.

II. Related Work

The integration of LLMs into robotics enables robots to understand and execute natural language instructions, which allows for more natural and effective HRI [3]. The main contribution of LLMs in robotics is their ability to enable natural language understanding and generation. Traditional robot interfaces often require users to learn specific command formats, but LLMs allow users to issue high-level instructions.

LLMs are neural networks trained on massive text corpora to learn statistical the patterns of language. An LLM models a probability distribution over sequences of tokens. Given input tokens, the model predicts the probability of subsequent tokens. LLMs are generally trained using auto-regressive modeling that can let the models learn relationships implicitly encoded in the data [4]. Given a prompt, the model generates outputs by iteratively sampling the probability of the next token given the previous tokens. This generative capability allows LLMs to produce instructions, explanations, or action plans. In robotics, the LLM's output can be post-processed into structured instructions or symbolic plans that guide a robot's behaviors [5], [6].

An example of LLM integration in robotics is the PaLM-SayCan framework. This system lets robots interpret natural language commands, decompose them into a series of actionable sub-tasks, and execute them. PaLM-SayCan uses an LLM to understand user instructions and determine feasible actions based on the robot's current context and environment. For instance, when a user requests, "Can you help me get an apple?", the LLM parses the instruction into subtasks such as "walk to the kitchen," "open the refrigerator," "retrieve the apple," and "deliver it to the user" [7]. Another example is the LM-Nav framework which focuses on natural language-based navigation. It combines a Vision Navigation Model (VNM), a Vision-Language Model (VLM), and an LLM to achieve effective navigation and control. The VNM constructs a map of the environment, and the VLM identifies objects to guide navigation. By using the strengths of the three components, LM-Nav was shown to autonomously navigate environments using natural language input [8].

Transformers are a key part of LLMs. Unlike traditional recurrent neural networks or long short-term memory, transformers use a self-attention mechanism that can directly model dependencies between any two tokens in a sequence [4]. The transformer's ability to capture long-range dependencies ND TO learn contextual representations has made it useful in robotics for dealing with action commands rather than just tokens. To handle robotic perception, transformers must process not only language tokens but inputs from other modalities such as visual inputs (images, depth maps). Instead of producing just next-word predictions, robotics-oriented transformers generate sequences of actions. Actions can be discretized or represented as parametric tokens.

One of the first attempts to use transforms in robotics was Robotics Transformer 1 (RT-1). RT-1 encoded images and natural language instructions as compact token representations. A recent version, Robotics Transformer X (RT-X) introduced cross-platform learning by training a universal robot control policy that can generalize across different robots and environments [9]. The integration of LLMs and transformers into robotics has advanced the field, allowing for more intuitive HRI and improving robot adaptability. Transformer-based architectures allow for efficient control and generalization across various robotic platforms [3], [10] which has allowed for further exploration of multimodal models, such as Vision-Language-Action (VLA) models, which integrate perception, decision-making, and control [11]. While these systems have shown promise, challenges remain, such as the need for larger multi-modal datasets, improved robustness in real-world scenarios, and more efficient on-device processing.

III. Methodology

The system is implemented in the Unity game engine, using Oculus VR integration with a wearable headset (Meta Quest 3). The AR space includes hand tracking so users could interact with the simulated objects without controllers, two control boards, (one for voice control and one for manual control, a five-degree-of-freedom robotic arm positioned at the center of four colored platforms (green, blue, black, and white) and four colored blocks (purple, yellow, orange, red) as seen in 1.

A user could interact with the simulated robot arm with 2 methods. The user could use a voice command board that allows users to press a button to issue spoken instructions. The user would hold down the virtual button for the length of time they were issuing the command. The command audio would have its waveform recorded by the on-headset microphones and sent to the Whishper API to be transformed into transcribed text. That text would then be sent to the specified OpenAI LLM model with a system prompt to return the intended action, item, and location that the user requested in a standardized format so it could be parsed and executed. The goal with using the LLM instead of simply parsing the given command was because using LLMs allowed for the interpretation of non-standardized ambiguous commands that a human unfamiliar with robotics might give. The LLM could take a command and turn it into a standardized format that the robot's software can interpret. Users also had access to a manual control board, which enabled direct user manipulation of the robot, the manual command board allowed users to press buttons to control each joint of the robot along with a pickup/drop button. These interaction interfaces are shown in 2

After receiving a parsed command, the robotic arm uses IK to orient its base joint toward the specified object or location. The arm then follows a routine to pick up the indicated object and then uses IK to move it to the requested platform following a revered version of the pickup routine to drop the item at the specified platform. Once completed, it returns to a home position. For example, an instruction like "Move the purple block to the green platform" prompts the LLM to produce a structured command identifying the action, the item, and the location. A command parser extracted the structured fields from the LLM response. The identified item and target platform were then mapped to corresponding Unity objects.

An additional evaluation was conducted. Five OpenAI models were tested for their ability to return a correctly structured command identifying the action showing their ability to infer missing details or segment complex instructions correctly. The script prompted each model with each command 4 times totaling 100 command for each model. The accuracy of each

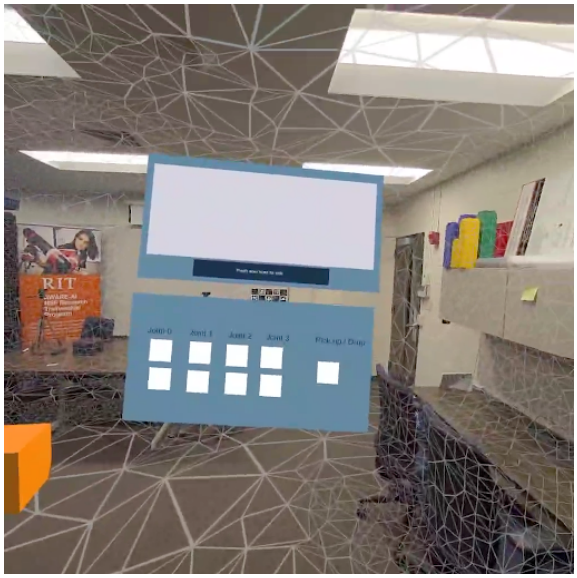

Fig. 2. The handheld command boards used for issuing voice commands or manually controlling the robot's joints.

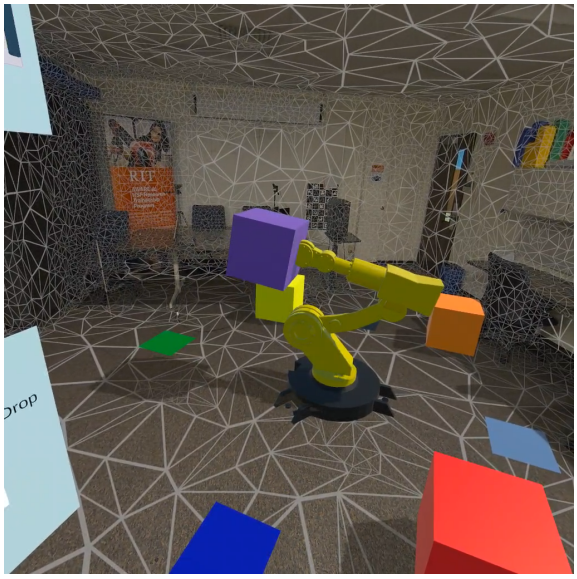

Fig. 3. The robot arm executing a pick-and-place action based on an LLM-parsed voice command.

prompt was recorded as correct or incorrect allowing for an accuracy to be given to each model indicating how accurate they were at paring the given commands.

IV. Results

Testing within the AR environment indicated that the system can reliably interpret and execute user commands. A user could deliver voice commands and the robotic arm typically performed the requested action after a brief processing delay of approximately .5-3 seconds. Also, the system demonstrated resilience to moderate background noise and variations in user speech. Although it was mainly tested in laboratory conditions,

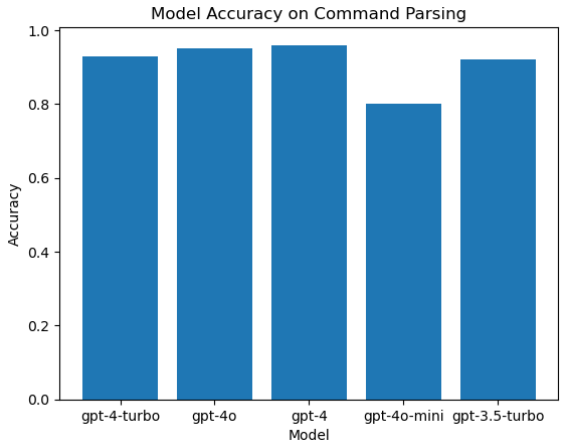

Fig. 4. Accuracy comparison of different OpenAI models on ambiguous commands. GPT-4 achieved the highest accuracy (96%), followed by gpt-4o (95%), gpt-4-turbo (93%), gpt-3.5-turbo (92%), and gpt-4o-mini (80%).

the transcription and interpretation pipeline was robust enough to handle everyday speaking conditions.

The evaluation script tested various models with 25 ambiguous commands. Five OpenAI models were tested, each handling 100 prompts (25 commands × 4 repetitions per command). The best-performing model was GPT-4 which achieved an accuracy score of 96%. Less capable variants such as gpt-4o-mini showed a noticeable drop in performance. These results show the importance of model selection: more advanced LLMs yield more consistent interpretation, thereby improving user experience and reducing the likelihood of errors or misunderstandings. Figure 4 summarizes the accuracy of various models tested.

V. Limitations and Future Work

While the demonstrated system successfully interprets and executes voice instructions via an LLM-based pipeline, it currently relies heavily on cloud-based services for both transcription (Whisper) and interpretation (OpenAI LLMs). This dependency introduces potential delays due to network latency and reduces system responsiveness. Future work should focus on developing compressed or quantized on-device models that can run efficiently on edge hardware.

Another limitation is the assumption of a controlled environment with known object positions. The current pipeline does not incorporate real-time vision modules, and the robotic arm's IK relies on the simulation with knowledge of object locations. Future work will integrate visual perception modules to enable the robotic arm to identify and track objects.

While an LLM-based pipeline effectively interprets ambiguous commands into structured formats, it remains a two-step solution. A promising direction is the integration of language understanding and action inference into a single action model. An action model treats actions as a modality, allowing the model to directly decode a user's natural language commands into actionable robot directives. Future work aims to develop and evaluate a VLA prototype, focusing on language-to-action

inference to enhance system reliability. As the scope of the system increases, additional features may be needed such as introducing clarification requests when commands are overly ambiguous and implementing collision avoidance algorithms in the IK solver. Overall, addressing these limitations would advance the system's effectiveness. Getting local inference, integrating vision models, and merging language and action would let robotic systems be applicable in a wider range of real-world environments.

REFERENCES

[1] Z. Wu, P. Shu, Y. Li, Q. Li, T. Liu, and X. Li, *Robot Control via Natural Instructions Empowered by Large Language Model*. Cham: Springer Nature Switzerland, 2024, p. 437–457. [Online]. Available: https://doi.org/10.1007/978-3-031-66656-8_19

[2] J. Fu, A. Rota, S. Li, J. Zhao, Q. Liu, E. Iovene, G. Ferrigno, and E. De Momi, "Recent advancements in augmented reality for robotic applications: A survey," *Actuators*, vol. 12, no. 8, p. 323, 2023. [Online]. Available: https://www.proquest.com/docview/2856748314/abstract/EA958FC13A224859PQ/1

[3] J. Wang, Z. Wu, Y. Li, H. Jiang, P. Shu, E. Shi, H. Hu, C. Ma, Y. Liu, X. Wang, Y. Yao, X. Liu, H. Zhao, Z. Liu, H. Dai, L. Zhao, B. Ge, X. Li, T. Liu, and S. Zhang, "Large language models for robotics: Opportunities, challenges, and perspectives," no. arXiv:2401.04334, Jan. 2024. [Online]. Available: http://arxiv.org/abs/2401.04334

[4] A. Vaswani, N. Shazeer, N. Parmar, J. Uszkoreit, L. Jones, A. N. Gomez, u. Kaiser, and I. Polosukhin, "Attention is all you need," in *Advances in Neural Information Processing Systems*, vol. 30. Curran Associates, Inc., 2017. [Online]. Available: https://papers.nips.cc/paper_files/paper/2017/hash/3f5ee243547dee91fbd053c1c4a845aa-Abstract.html

[5] F. Zeng, W. Gan, Y. Wang, N. Liu, and P. S. Yu, "Large language models for robotics: A survey," no. arXiv:2311.07226, Nov. 2023, arXiv:2311.07226 [cs]. [Online]. Available: http://arxiv.org/abs/2311.07226

[6] S. Gupta, K. Yao, L. Niederhauser, and A. Billard, "Action contextualization: Adaptive task planning and action tuning using large language models," *IEEE Robotics and Automation Letters*, vol. 9, no. 11, p. 9407–9414, Nov. 2024. [Online]. Available: https://ieeexplore.ieee.org/document/10679904

[7] M. Ahn, A. Brohan, N. Brown, Y. Chebotar, O. Cortes, B. David, C. Finn, C. Fu, K. Gopalakrishnan, K. Hausman, A. Herzog, D. Ho, J. Hsu, J. Ibarz, B. Ichter, A. Irpan, E. Jang, R. J. Ruano, K. Jeffrey, S. Jesmonth, N. J. Joshi, R. Julian, D. Kalashnikov, Y. Kuang, K.-H. Lee, S. Levine, Y. Lu, L. Luu, C. Parada, P. Pastor, J. Quiambao, K. Rao, J. Rettinghouse, D. Reyes, P. Sermanet, N. Sievers, C. Tan, A. Toshev, V. Vanhoucke, F. Xia, T. Xiao, P. Xu, S. Xu, M. Yan, and A. Zeng, "Do as i can, not as i say: Grounding language in robotic affordances," no. arXiv:2204.01691, Aug. 2022, arXiv:2204.01691 [cs]. [Online]. Available: http://arxiv.org/abs/2204.01691

[8] D. Shah, B. Osinski, B. Ichter, and S. Levine, "Lm-nav: Robotic navigation with large pre-trained models of language, vision, and action," no. arXiv:2207.04429, Jul. 2022, arXiv:2207.04429 [cs]. [Online]. Available: http://arxiv.org/abs/2207.04429

[9] A. O'Neill, A. Rehman, A. Maddukuri, A. Gupta, A. Padalkar, A. Lee, A. Pooley, A. Gupta, A. Mandlekar, A. Jain, A. Tung, A. Bewley, A. Herzog, A. Irpan, A. Khazatsky, A. Rai, A. Gupta, A. Wang, A. Singh, A. Garg, A. Kembhavi, A. Xie, A. Brohan, A. Raffin, A. Sharma, A. Yavary, A. Jain, A. Balakrishna, A. Wahid, B. Burgess-Limerick, B. Kim, B. Schölkopf, B. Wulfe, B. Ichter, C. Lu, C. Xu, C. Le, C. Finn, C. Wang, C. Xu, C. Chi, C. Huang, C. Chan, C. Agia, C. Pan, C. Fu, C. Devin, D. Xu, D. Morton, D. Driess, D. Chen, D. Pathak, D. Shah, D. Büchler, D. Jayaraman, D. Kalashnikov, D. Sadigh, E. Johns, E. Foster, F. Liu, F. Ceola, F. Xia, F. Zhao, F. Stulp, G. Zhou, G. S. Sukhatme, G. Salhotra, G. Yan, G. Feng, G. Schiavi, G. Berseth, G. Kahn, G. Wang, H. Su, H.-S. Fang, H. Shi, H. Bao, H. Ben Amor, H. I. Christensen, H. Furuta, H. Walke, H. Fang, H. Ha, I. Mordatch, I. Radosavovic, I. Leal, J. Liang, J. Abou-Chakra, J. Kim, J. Drake, J. Peters, J. Schneider, J. Hsu, J. Bohg, J. Bingham, J. Wu, J. Gao, J. Hu, J. Wu, J. Wu, J. Sun, J. Luo, J. Gu, J. Tan, J. Oh, J. Wu, J. Lu, J. Yang, J. Malik, J. Silvério, J. Hejna, J. Booher, J. Tompson, J. Yang, J. Salvador, J. J. Lim, J. Han, K. Wang, K. Rao, K. Pertsch, K. Hausman, K. Go, K. Gopalakrishnan, K. Goldberg, K. Byrne, K. Oslund, K. Kawaharazuka, K. Black, K. Lin, K. Zhang, K. Ehsani, K. Lekkala, K. Ellis, K. Rana, K. Srinivasan, K. Fang, K. P. Singh, K.-H. Zeng, K. Hatch, K. Hsu, L. Itti, L. Y. Chen, L. Pinto, L. Fei-Fei, L. Tan, L. J. Fan, L. Ott, L. Lee, L. Weihs, M. Chen, M. Lepert, M. Memmel, M. Tomizuka, M. Itkina, M. G. Castro, M. Spero, M. Du, M. Ahn, M. C. Yip, M. Zhang, M. Ding, M. Heo, M. K. Srirama, M. Sharma, M. J. Kim, N. Kanazawa, N. Hansen, N. Heess, N. J. Joshi, N. Suenderhauf, N. Liu, N. Di Palo, N. M. M. Shafiullah, O. Mees, O. Kroemer, O. Bastani, P. R. Sanketi, P. T. Miller, P. Yin, P. Wohlhart, P. Xu, P. D. Fagan, P. Mitrano, P. Sermanet, P. Abbeel, P. Sundaresan, Q. Chen, Q. Vuong, R. Rafailov, R. Tian, R. Doshi, R. Martín-Martín, R. Baijal, R. Scalise, R. Hendrix, R. Lin, R. Qian, R. Zhang, R. Mendonca, R. Shah, R. Hoque, R. Julian, S. Bustamante, S. Kirmani, S. Levine, S. Lin, S. Moore, S. Bahl, S. Dass, S. Sonawani, S. Song, S. Xu, S. Haldar, S. Karamcheti, S. Adebola, S. Guist, S. Nasiriany, S. Schaal, S. Welker, S. Tian, S. Ramamoorthy, S. Dasari, S. Belkhale, S. Park, S. Nair, S. Mirchandani, T. Osa, T. Gupta, T. Harada, T. Matsushima, T. Xiao, T. Kollar, T. Yu, T. Ding, T. Davchev, T. Z. Zhao, T. Armstrong, T. Darrell, T. Chung, V. Jain, V. Vanhoucke, W. Zhan, W. Zhou, W. Burgard, X. Chen, X. Wang, X. Zhu, X. Geng, X. Liu, X. Liangwei, X. Li, Y. Lu, Y. J. Ma, Y. Kim, Y. Chebotar, Y. Zhou, Y. Zhu, Y. Wu, Y. Xu, Y. Wang, Y. Bisk, Y. Cho, Y. Lee, Y. Cui, Y. Cao, Y.-H. Wu, Y. Tang, Y. Zhu, Y. Zhang, Y. Jiang, Y. Li, Y. Li, Y. Iwasawa, Y. Matsuo, Z. Ma, Z. Xu, Z. J. Cui, Z. Zhang, and Z. Lin, "Open x-embodiment: Robotic learning datasets and rt-x models: Open x-embodiment collaboration0," in *2024 IEEE International Conference on Robotics and Automation (ICRA)*, May 2024, p. 6892–6903. [Online]. Available: https://ieeexplore.ieee.org/document/10611477

[10] G. Singhal and A. Singh, "The large action model: Pioneering the next generation of web and app engagement," in *2024 11th International Conference on Reliability, Infocom Technologies and Optimization (Trends and Future Directions) (ICRITO)*, Mar. 2024, p. 1–6. [Online]. Available: https://ieeexplore.ieee.org/document/10522132/?arnumber=10522132

[11] Z. Wang, Z. Zhou, J. Song, Y. Huang, Z. Shu, and L. Ma, "Towards testing and evaluating vision-language-action models for robotic manipulation: An empirical study," no. arXiv:2409.12894, Sep. 2024, arXiv:2409.12894 [cs]. [Online]. Available: http://arxiv.org/abs/2409.12894