“Hey Medbot”: Toward Autonomous Assistants in Operating Rooms

Joseph C. Liechty[1,2], Atharv Belsare[1,2], Zohre Karimi[1,2], Jack Koster[1], Daniel Brown[1,2] and Alan Kuntz[1,2]

***Abstract*— The term “robotic surgery” often conjures images of robots remotely controlled by a doctor performing complicated surgical procedures directly on a patient. There are many other roles in an operating theater, however, that medical robots can fulfill. One of those roles is assisting expert doctors and nurses in preparing tools for procedures. This work describes a preliminary system designed for this purpose which users interact with through voice commands, and which is capable of handing medical tools to the human. The current prototype of the system uses computer vision models to identify people and tools in the scene and humans interact with the system through natural language captured by a microphone and processed by a large language model. Robot systems like this one may serve as an extra hand in the operating room, making it easier for human experts to focus on the patient they are caring for.**

I. Introduction

End-to-end robotic systems capable of interacting with people in natural ways is the focus of a large subfield of research. Many groups are developing robots that can assist with household chores [1]–[3]. The system we are proposing is designed to complete crucial, life-saving tasks with the same level of natural interaction and predictability as its home assistance counterparts. *Medbot*, a robotic assistant for operating theaters and clinics, performs support tasks for advanced care providers such as surgeons, nurses, and paramedics. Currently the system is capable of responding to verbal requests for a variety of medical tools by locating the tools in the environment, picking them up, and then placing them in the person’s hand (see Fig. 1).

This paper outlines the current state of our system, including mathematical approaches to framing the problem and the performance of the more developed parts of our system. We also discuss some of our findings from building an end-to-end robotic system and areas of the project that we are further developing.

II. System Design

This section describes how the different models we use integrate with each other to give the robot its reasoning capabilities.

A. Natural Language Interface and High-Level Reasoning

In the system’s current form, the robot, a 7-DOF serial link manipulator arm (see Fig. 1), exists in a listening state until it detects that a human is making a request. The request triggers a chain of reasoning that results in the robot generating a high-level action to execute in order to complete the next step in the task. Generating this action can be thought of as sampling from the distribution of possible high-level actions conditioned on the language prompt, the previous action, and information about the scene observed by the robot,

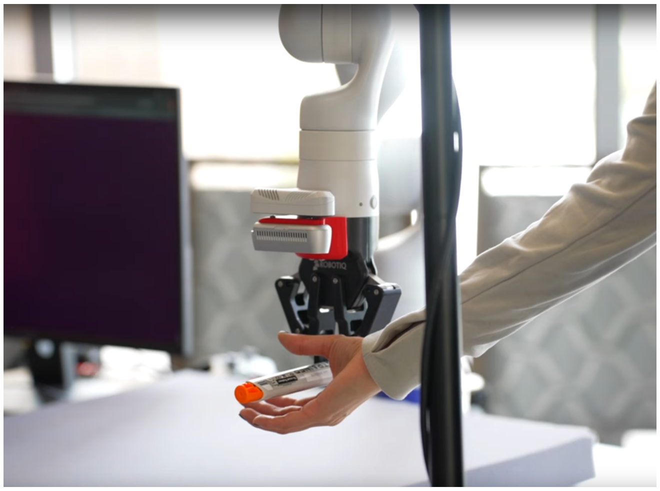

Fig. 1. The robot handing an epinephrine autoinjector to a person.

$$A_t \sim p_\phi(A_t|L_t, I_t, A_{t-1}) \tag{1}$$

where A_t is a representation of the high-level action for the current time step t, A_{t-1} is the previous high-level action, L_t is the language prompt from the human, I_t is the sensed scene and context, e.g., a depth image of the environment or other sensory input, and ϕ is the model parameters (e.g., weights).

Figure 2 shows how the robot moves through its decision making process and how the different conditioning variables of the reasoning model are changed. In the *listening* state (see Fig. 2), the robot waits for the human to say the keywords, “Hey robot”, and then translates the verbal command that follows into a string of text using speech recognition [4]. The string is concatenated with a custom prompt to form the language conditioning variable, L_t, which is passed through an LLM (in this instantiation, ChatGPT 4o [5]) to reason over the object that the person wants the robot to pick up and hand to them given the unstructured natural language command, L_{obj}. In this way, the human provider can interact with the robot as if it were a human, rather than needing to learn a specific language or other structured interface that adds mental burden to the interaction during potentially high-stakes and stressful patient care situations. For example, the human provider could say “I need to intubate this patient. They are no longer maintaining their own airway.” The LLM takes

joe.liechty@utah.edu, alan.kuntz@utah.edu

[1] Kahlert School of Computing, University of Utah, Salt Lake City, UT 84112, USA

[2] Robotics Center, University of Utah, Salt Lake City, UT 84112, USA

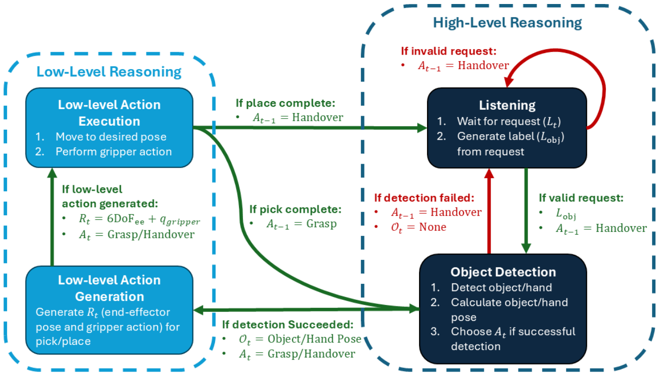

Fig. 2. State machine showing how the robot moves through its decision making process from natural language request to task completion.

this and a pre-structured prompt describing the available medical objects in the scene and other contextual information and it may determine that handing an endotracheal tube or laryngoscope to the provider would be helpful.

The state machine operates in the following way:

- If there is a language prompt, a successful object detection, and the previous high-level action was "handover" then the robot will select the high-level action "grasp".
- If there is no language prompt, the previous high-level action was "grasp", and a hand is detected in the scene then the robot will select the action "handover".
- If the detection of the object/hand fails then the robot will select the action "listening" and return to the listening state.

The stochasticity in the action generation comes from the use of an LLM to generate L_{obj}. This is similar to the stochasticity that is found in methods that use Vision-Language-Action models (VLAs) or Vision-Language modes (VLMs) to generate high-level plans [1]–[3], and notably is also similar to human medical assistants in training.

B. Vision Pipeline

Whether the robot can detect the required object for the next step of the task determines what action the robot will take. The vision pipeline locates and identifies objects in the workspace so that the robot can reason about how to manipulate the objects. This is done using discriminative models represented by

$$\mathcal{O}_t = f_\theta(I_t, A_t, L_{\text{obj}}) \tag{2}$$

where $\mathcal{O}_t$ is the 6DoF pose of the object's location in the workspace at time step t, I_t, A_t, and L_{obj} are as defined earlier, and θ the model parameters.

I_t is an RGBD image taken with a Realsense D435i camera that is observing the environment (not an egocentric camera). For potential pick actions, a fine-tuned Detectron2 model [6] and SegmentAnything (SAM) [7] were used to label objects of interest and extract their segmented point clouds. The Iterative Closest Point (ICP) algorithm was used to align the segmented point clouds with surface models of the objects, registering the object's coordinate frame and origin with the segmented point cloud.

The Detectron2 model was fine-tuned on a dataset of 2000 annotated images containing instances of the following medical objects: syringes, forceps, shears, epinephrine autoinjectors (e.g., EPIPEN), tourniquets, gauze pads, gauze rolls, bag-valve masks, scalpels, and endotracheal tubes.

To determine how to hand the object to the provider, the MediaPipe hand detection model [8] was used to find the providers' hands in the scene. Keypoints on the palm were used to assign a coordinate frame to the hand. Depending on L_{obj} and A_t, $\mathcal{O}_t$ would be either the medical object pose (for object name and grasp) or the person's hand pose (for handover).

C. Low-Level Skills and Object Manipulation

To manipulate objects, we must generate appropriate low-level robot actions or "skills" conditioned on object location, the high-level action, and other observations about the objects. This is akin to sampling a low-level action from the conditional distribution

$$R_t \sim p_\psi(R_t | \mathcal{O}_t, A_t, L_{\text{obj}}) \tag{3}$$

where R_t is the low-level action, $\mathcal{O}_t$, A_t, and L_{obj} are as defined earlier, and ψ is the model weights.

We use two models to generate low-level actions, one to generate grasps and the other to generate handovers. For both pick (grasp) and place skills (handover), R_t is a 6DoF end-effector pose and a gripper command. For the pick skill, we generated grasps by sampling from a set of valid grasps collected by teleoperating the robot and conditioned on the name of the object, L_{obj}. For the place skill, handover poses were generated by sampling from a Mixture Density Network (MDN) conditioned on L_{obj}. The MDN was trained on object-to-hand relative poses collected during human demonstrations of the handovers.

III. Results

We evaluate each component of the system, as well as overall task success below.

To evaluate the preliminary performance of the individual components and the system as-a-whole, we enlisted two participants to interact with the robot 10 times each. In each interaction, the participant asked the robot to pass them the endotracheal tube 5 times and the epinephrine autoinjector 5 times. Table I shows how many times the robot was able to complete each sub-task in the handover without human intervention (i.e. assistance). In the cases where the robot failed to complete a sub-task, the human helped the robot until the robot was able to succeed at that sub-task and continue to the next sub-task.

All of the system failures in these trials occurred in natural language processing and in the pick skill. In these experiments, the vision models were able to identify the object of interest each time and our place skill was successful each time. Fig. 3 shows the distribution of the number of human interventions that were required for the natural language and pick sub-tasks to succeed.

TABLE I

HANDOVER TASK AND SUB-TASK INTERVENTION-FREE SUCCESS RATE.

Object	**[Intervention-Free Trials/Total Trials]**				
	Nat. Lang.	**Detect Object**	**Pick**	**Detect Hand**	**Place**
Epinephrine Autoinjector	8/10	10/10	6/10	10/10	10/10
Endotracheal Tube	8/10	10/10	6/10	10/10	10/10

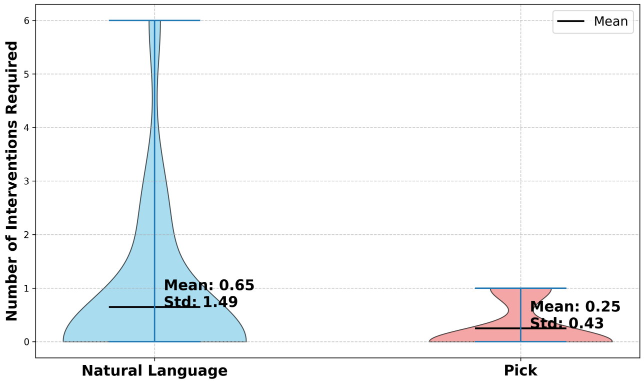

Fig. 3. Distribution of human interventions needed for sub-tasks to be successful.

We observed qualitatively that the vision model occasionally falsely assigned labels to nonmedical objects in the scene (for example, labeling a Coke can as ”epinephrine autoinjector”) and that it would sometimes fail to detect the object in the first frame. To account for these failure modes, we removed out-of-distribution objects (e.g., objects the model was not fine-tuned to detect) from the scene and had the robot make 20 attempts (< 1s) to find the object in the scene before returning a “no object found” message.

We believe that both our pick and place skills need to be further refined. The place skill appeared to perform better during the user study but we believe that was due to the person making small adjustments to their hand pose to better match the robot’s pose during the handover. Since the pick involves only the robot and the inanimate object, the need for improving these skills was more apparent during that phase.

We also observed that the deficiencies in the natural language processing may be improved via LLM prompt engineering or fine-tuning LLM models to medical settings.

IV. DISCUSSION AND FUTURE WORK

We created this preliminary system to begin to identify the bottlenecks in creating an end-to-end robotic assistant for operating theaters and other medical settings. Based on what we have learned during this process, there are several areas of the system that we are actively improving.

The first area is task planning. Many medical tasks are structured and humans who perform those tasks often follow a set of predefined procedures. This structure lends itself well to our framework of combining state machines that act in certain, predictable ways, with language models that stochastically interpret human speech. In less structured environments VLMs and VLAs have been used as task planners with some success [1]–[3]. We are exploring combining the flexibility of VLMs with the structure of medical procedures to create a task planning policy that has a decision making process humans understand and feel safe interacting with.

The second area is perception. Our low-level skills we are developing use visual information to generate robot actions and robust object detection and segmentation methods will improve their performance. We are developing a vision model based on YOLOv8 [9] that is capable of producing segmented point clouds in a single pass. We are also working on improving model performance generally by gathering more data for object classes that have lower performance.

The third area we are focusing on improving is our low-level skills. Specifically we are focusing on further developing the pick and place skills so that they can be conditioned on multiple high-level action contexts. This will enable our robot to perform more complicated, multi-step tasks. Improving these skills is also important for removing the need for human intervention in the pick sub-task (see Fig. 3), where having a human help the robot defeats the purpose of the handover task.

There are many areas for improvement in our system, but the framework we have developed so far has helped us to gain a wholistic view of how to build an end-to-end robotic system that is flexible, predictable, and useful. In future versions of this project we will evaluate the areas we have improved on and how well the system interacts with people while providing patient care in a medical setting.

REFERENCES

[1] K. Black, N. Brown, D. Driess, A. Esmail, M. Equi, C. Finn, N. Fusai, L. Groom, K. Hausman, B. Ichter, S. Jakubczak, T. Jones, L. Ke, S. Levine, A. Li-Bell, M. Mothukuri, S. Nair, K. Pertsch, L. X. Shi, J. Tanner, Q. Vuong, A. Walling, H. Wang, and U. Zhilinsky, “Pi 0: A Vision-Language-Action Flow Model for General Robot Control,” Nov. 2024, arXiv:2410.24164 [cs]. [Online]. Available: http://arxiv.org/abs/2410.24164

[2] G. R. Team and et al., “Gemini Robotics: Bringing AI into the Physical World,” Mar. 2025, arXiv:2503.20020 [cs]. [Online]. Available: http://arxiv.org/abs/2503.20020

[3] Y. Li, Y. Deng, J. Zhang, J. Jang, M. Memmel, R. Yu, C. R. Garrett, F. Ramos, D. Fox, A. Li, A. Gupta, and A. Goyal, “HAMSTER: Hierarchical Action Models For Open-World Robot Manipulation,” Feb. 2025, arXiv:2502.05485 [cs]. [Online]. Available: http://arxiv.org/abs/2502.05485

[4] A. Zhang, “Speech recognition (version 3.11),” https://github.com/Uberi/speech_recognition, 2017, software.

[5] OpenAI, “Chatgpt-4.0,” https://chat.openai.com, 2025, large language model.

[6] Y. Wu, A. Kirillov, F. Massa, W.-Y. Lo, and R. Girshick, “Detectron2,” https://github.com/facebookresearch/detectron2, 2019.

[7] A. Kirillov, E. Mintun, N. Ravi, H. Mao, C. Rolland, L. Gustafson, T. Xiao, S. Whitehead, A. C. Berg, W.-Y. Lo, P. Dollár, and R. Girshick, “Segment anything,” 2023. [Online]. Available: https://arxiv.org/abs/2304.02643

[8] C. Lugaresi, J. Tang, H. Nash, C. McClanahan, E. Uboweja, M. Hays, F. Zhang, C.-L. Chang, M. G. Yong, J. Lee, W.-T. Chang, W. Hua, M. Georg, and M. Grundmann, “Mediapipe: A framework for building perception pipelines,” 2019. [Online]. Available: https://arxiv.org/abs/1906.08172

[9] Ultralytics, “Yolov8: A novel object detection algorithm with enhanced performance and robustness,” in *2024 International Conference on Advances in Data Engineering and Intelligent Computing Systems (ADICS)*, 2024.