UT Austin Villa@Home
2026 Team Description Paper

Yuqian Jiang Lingyun Xiao Yoonwoo Kim Raghav Arora
Taijing Chen Sadanand Modak Noah Patton Eric Hsuing
Nathan Tsoi Justin Hart Joydeep Biswas Peter Stone

November 30, 2025

Abstract. UT Austin Villa has participated in eight RoboCup@Home competitions, performing respectably in each. What is more exciting, however, is that we have begun a strong program of research that has been in part inspired by our efforts in this competition. It is our intention to build a comprehensive service robot system which is used in our laboratories, in real-world deployments, and to compete in RoboCup@Home. This year, UT Austin is participating in the competition with two teams, one with a wheeled mobile manipulator and the other with a humanoid robot, with UT Austin Villa at Home as the wheeled mobile manipulator team. In this Team Description Paper, you will find the highlights of our performance in the competition in 2025 and our plans for 2026.

1 Introduction

Using the RoboCup@Home team as a focal point for inter-department and inter-laboratory collaboration, UT Austin Villa@Home has pursued an ambitious research program towards the goal of the development of a comprehensive service robot system. We want to enter RoboCup@Home not with a suite of different programs for each task, but with a single program which is capable of completing all the tasks.

UT Austin Villa@Home is a collaborative effort between PIs and students in Texas Robotics which is a united partnership between the Departments of Computer Science, Mechanical Engineering, Aerospace Engineering, Electrical and Computer Engineering at the University of Texas at Au stin, with a diverse set of research interests driving our team. We have competed in eight RoboCup@Home events. In 2007, we took second place. In 2017, we entered into the newly-formed Domestic Standard Platform League (DSPL) and took third place, having received our robot only a couple of months before the competition. In 2018, the team developed a design intended to allow us to develop a single system which would enter into all of the stages of the competition, encompassing knowledge representation, mapping, and architectural aspects. The team advanced to the second stage and was able to score in difficult tasks such as Enhanced General Purpose Service Robot (EGPSR). In 2019, we improved the system with better perception and manipulation modules. In 2021, we continued to develop our object recognition and manipulation capabilities using the HSR simulator, and finished in the 3rd place in the 2021 competition. In 2022, we continued to strengthen our perception pipeline and re-designed the person tracking module, and qualified for the second stage in Bangkok. In 2023, we explored methods to combine LLMs with task and motion planning for interactive mobile manipulation. In 2024, we upgraded our architecture with state-of-the-art models in perception, manipulation, and command understanding, leading to better task performance in various RoboCup@Home tests and our advancement to Stage 2. In 2025, we used integrated foundation models, like offline vision language models for adaptive Human-Robot-Interaction, and generalized performance across diverse objects and tasks. We used Behaviour Trees for modular decomposition of tasks and use it with HSR API for execution. We finished in the 2nd place of RoboCup@Home DSPL in 2025, and in particular, we scored one of the highest *EGPSR* points in RoboCup@Home DSPL, demonstrating the robustness and flexibility of our system. Our efforts have resulted in nine publications [1,2,3,4,5,6,7,8,9], with more in progress. Going into 2026, we are changing our robot from Toyota's HSRB to a custom-built cobot (refer to figure 1), and we plan to further improve the core components of our system and develop more rigorous approaches to the tasks. We will also extend our research efforts in knowledge representation and task-and-motion planning.

2 Software and Scientific Contributions

This section describes the component technologies we developed across multiple tasks for our robot architecture, knowledge representation, semantic perception, object manipulation, and person following. The underlying architecture [6] is designed in a manner consistent with our ongoing Building-Wide Intelligence project [10]. While using a different hardware platform, many of the objectives and capabilities are the same. Our robot for this year (as shown in figure 1) contains a CoBot [11] base, with a Kinovo arm, and a mounted camera for perception.

2.1 Robot Architecture

Our architecture is designed for service robots to handle dynamic interactions with humans in complex environments. The three-layer architecture, as shown in Figure 2, outlines integration of the robot's skill components, such as perception and manipulation, with high-level reactive and deliberative controls. The top layer sequences and executes skills, and is reactive during execution to respond to changes. A central knowledge base facilitates knowledge sharing from all the

Fig. 1: Cobot for the 2026 RoboCup@Home competition.

components. The deliberative control layer uses the knowledge base to reason about the environment, and can be invoked to plan for tasks that cannot be statically decomposed. Details on implementation of these layers can be found in our recent paper [6].

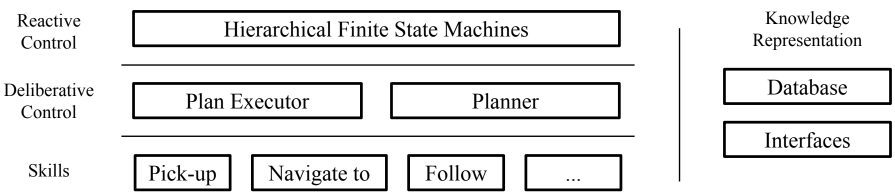


Fig. 2: Implementation of our robot architecture on Cobot.

2.2 Knowledge Representation and Planning

Our knowledge representation subsystem stores grounded robot knowledge in a SQL database in order to allow for fast access and easy querying. Queries can be formed using custom C++ and Python libraries. For instance, in the *Storing Groceries* and *GPSR* tasks, the knowledge base is used to query object properties such as categories and default locations. The knowledge base can be dynamically updated by our perception system described below. Fig. 3 shows the knowledge base after the robot has detected a ketchup bottle on the dining table.

The knowledge base can be interfaced through a simple predicate logic form which can be then imported for task planning. Core to our KR subsystem is the ability to reason about hypothetical objects that are requested by users but unseen by the robot. This capability is crucial to our solution of the incomplete commands in earlier versions of the *EGPSR* test. Details on our knowledge representation and planning system can be found in our paper [2].

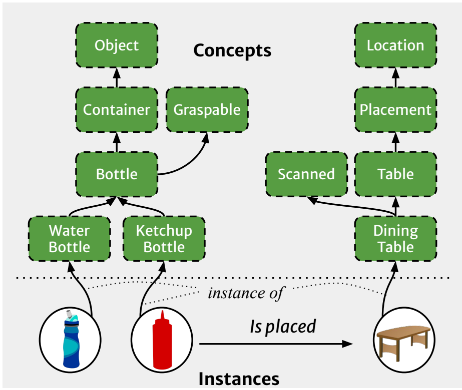


Fig. 3: Visualization of a knowledge base grounded in the robot's perception.

2.3 Command Understanding

To solve commands that are generated on the fly in *GPSR* and *EGPSR* tests, the robot has to accurately transcribe the operator's speech and parse it to a structured format for the downstream controller. Our command understanding pipeline performed well in the 2024 competition, successfully parsing all seven commands that were encountered in the tests. Our team's qualification video highlights one of our *GPSR* runs in Eindhoven, where the robot understood both commands and executed the correct steps to solve them.

Due to the unreliable network connectivity and speed for audio uploads at RoboCup, we deployed local models of OpenAI Whisper [12] and Vosk[1] for speech recognition. When the robot listens for a command, the audio is streamed from the Cobot's microphone using ROS and recorded on the backpack laptop. Our speech-to-text node integrates a Vosk model to detect the end of speech, processes the full audio recording by Whisper, and outputs the text.

[1] https://github.com/alphacep/vosk

For speech parsing, we leverage the ability of large language models (LLMs) like GPT-4o to translate natural language into structured outputs. We define a JSON schema according to the grammar of the current command generator, and prompt GPT-4o to parse the instruction into a valid JSON object while applying common-sense corrections to speech transcription errors. For *GPSR* tests, a state machine is assembled based on the task type and the parameters to execute the command. For arbitrary tasks given in natural language, we have shown that our framework is able to parse the commands to other formats such as PDDL problem definitions which can then be solved by planners [7][8].

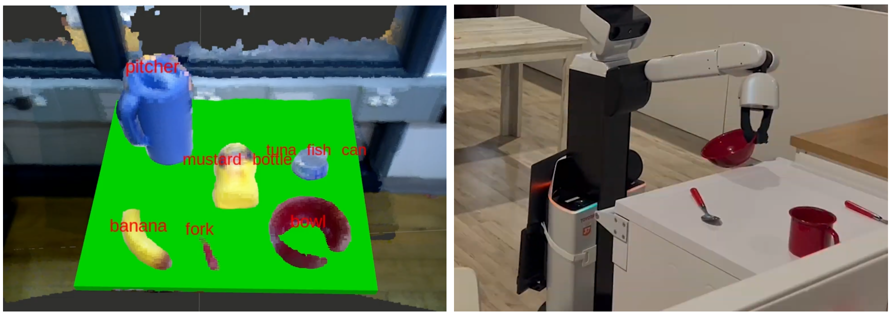
2.4 Semantic Perception

We employ a semantic perception module whose purpose is to process raw video and depth data from the robot's sensors and extract information that can be processed by the manipulation, navigation, and knowledge reasoning modules. The main output representations are a query-able scene graph of objects in the environment and a partial 3D map of the world.

The main input to our semantic perception module is RGBD camera data. Compressed RGB and depth images from the robot are streamed to an offboard computer that runs the perceptual system. This image data is then consumed by finding objects via the YOLO object detection network [13]. During set-up days, we annotate the set of objects, adjust the segmentation masks from Segment Anything [14], and fine-tune YOLOv8 segmentation models [15]. Next, semantic information about the world is synthesized in two main ways: an instance-level 3D segmentation of the local point cloud and a global scene graph. For the former, a 3D point cloud is integrated as the robot scans a location (e.g. kitchen table), and regions of the point cloud corresponding to detected objects are fused together from 2D to 3D based on geometric and semantic information. The scene reconstruction is implemented in the Open3D library [16] and the 2D-to-3D instance fusion is based on a recent approach [17]. For the latter, the objects are stored in a scene graph and wrapped with an efficient querying interface that integrates with our knowledge representation system.

The synthesized semantic information is then made available to plugins in an event-based model, where a plugin can request access to semantic information that it wants to operate on. Supported plugins include custom RANSAC plane detectors used to detect surfaces, and point cloud cropping with bounding box fitting for use in manipulation. Figure 4a shows a visualization of the synthesized point cloud with object labels and the detected plane after a table is scanned by our semantic perception module.

A significant limitation is the partial nature of the 3D environmental map. Only a partial map is constructed due to the realtime processing constraint; namely, full views of the world cannot be stitched together. Alternatively, GPU-based techniques for combining full point clouds could potentially overcome this limitation, and thus provides a direction for future development. Benefits of having full 3D environmental maps include the ability to directly localize objects with respect to the robot for task and motion planning. In 2024, we improved



(a) Object and plane detections

(b) Successful grasp of a bowl

our semantic perception framework with state-of-the-art approaches to generate open-vocabulary 3D scene graphs. For 2026, we are testing foundation models like Grounded-SAM [18], and our semantic perception has the ability to leverage such open-vocabulary detection and segmentation models. This improvement will enable our system to handle unknown objects and open-vocabulary queries.

2.5 Manipulation

The purpose of our manipulation system is to pick up diverse objects of different shapes and sizes and put them down on various surfaces. Our manipulation stack consists of three main components which we describe below: grasp and place pose sampling, concurrent motion planning, and closed-loop execution.

Sampling Goal Poses Our semantic perception system provides instance-level point clouds and 3D bounding boxes for objects of interest. We have integrated two grasp pose generators. The first is a state-of-the-art model GraspGen [19]. The model is trained for diffusion-based 6-DOF robotic grasp generation from table-top manipulator data. As we are using a kinova arm with the cobot this year, and we have found that GraspGen works the best for the cobot after transforming the target object's point cloud to look like it came from a top-down camera pointed at the surface. We use Grounded-SAM to segment the objects, and GraspGen to detect dense grasp poses on most objects including those with complex geometries. We post-process the poses and rank them according to their scores provided by the model. Figure 4b shows that GraspGen generated a grasp pose which pointed the gripper at the edge of the bowl, resulting in a successful grasp in the *Serve Breakfast* test in RoboCup 2024.

For flat objects (e.g. spoons and sponges), box-shaped objects (e.g. cereal boxes), and some deformable objects (e.g. bags of chips), we have found that sparse grasp poses can be computed from the bounding box with more consistent results. Based on tight 3D bounding boxes, potential grasp poses are computed that place the gripper on the top of the object as well as on all sides, with multiple possible rotations of the wrist. For rigid objects, invalid poses are filtered out by projecting the gripper onto the object and seeing if there is a collision.

For placing, we randomly sample poses on the target surface for the grasped object and compute the desired gripper pose. Collisions are checked between the placed bounding box with the bounding boxes of other objects already on the surface. If the object is being placed in a cabinet with multi-level shelves, we also check the height of bounding box against the vertical space above the target shelf, and rotate the gripper if necessary.

Motion Planning Once the gripper's target poses are determined, collision-free joint trajectories need to be planned in order for the robot to achieve a desired pose. Our solution uses MoveIt [20] motion planning stack with custom configurations for various pick and place scenarios. We plan to integrate CuRobo [21] along with moveit for faster motion planning. The bounding boxes of collision objects and surfaces from the perception module are populated into the collision world. Since motion planning takes a significant amount of time, reducing this bottleneck greatly improves the efficiency of the robot. For tasks such as *Storing Groceries*, the robot has to repeatedly visit the same location to manipulate objects. We have employed several strategies to speed up the manipulation pipeline. First, we pick up the objects in the ascending order of their distances to the edge, so the number of potential collisions are reduced. Second, we wrap the motion planning module in a concurrence container of the state machine, so that motion planners can be computed in parallel with execution.

Execution Next, executing a motion plan precisely is usually not feasible. This is because, as the plan is executed, the software solely uses odometry to control its position and the resultant drift can cause errors in how much the robot thinks it has moved. To overcome this obstacle, we slightly modify desired grasp poses by having the gripper be some offset away from the object. This way, after a motion plan is generated and executed, the robot's gripper is close to the object, but there remains a small gap. We take advantage of this small gap by employing a real-time, closed-loop grasp adjustment based on the fast YOLO detections applied to images from the kinovo's hand camera. We use the position of the generated 2D bounding box to align the gripper with the target object. A proportional controller is used to publish a velocity command to the robot base based on the distance between the center of the hand camera image and the center of the bounding box. This practically means that the robot shifts slightly to align the gripper perfectly with the centroid of the object. The gap is then closed by moving in a straight line towards the object.

2.6 Person Tracking, and Following

A home service robot must be able to find and track people in crowded environments. In 2024, and 2025, we improved our person recognition system for interactive tasks such as *Receptionist*, *Help me Carry EGPSR*.

Person Tracking Our vision-based person tracking module implements the BoT-SORT algorithm [22] with adaptations for a RGBD camera on a mobile robot. Instead of tracking the detected persons' bounding boxes in the image frame, we estimate and track their 2D positions in the map frame. A YOLOv12-pose model is used for detecting body keypoints. The keypoints are post-processed for recognizing gestures such as waving, raising arms, and pointing. A person re-identification model is equipped for when a person leaves the robot's view for some time and re-enters. Further, the module supports on-demand re-identification from a list of candidates in *Receptionist*.

Person Following To achieve robust and efficient person following in the *Help Me Carry* task, perception, robot gaze control, and navigation must be effectively integrated. Previously, we have developed person following capabilities using sensor fusion, active search using trajectory and waypoints predictions, and construct fully autonomous behaviors to follow people including temporary losses of the target being followed. Details on our person following approach can be found in our paper [3]. In 2025, we implemented RRT-star planner for sampling a goal pose close to the target person, and followed the person updating their position in parallel using Behaviour Trees. When a person moves out of the camera frame, the robot moves in place, looking for the person. In 2026, we plan to upgrade this person following framework for unknown environments and integrate with our new person tracker described above.

3 Conclusion

UT Austin Villa@Home has been a strong competitor and has a tradition of synergistic research our RoboCup@Home team and our other research efforts. RoboCup@Home has become a driving force in robotics research at UT Austin. We look forward to seeing everyone again in Incheon, South Korea in 2026.

References

1. Rishi Shah, Yuqian Jiang, Haresh Karnan, Gilberto Briscoe-Martinez, Dominick Mulder, Ryan Gupta, Rachel Schlossman, Marika Murphy, Justin Hart, Luis Sentis, and Peter Stone. Solving service robot tasks: Ut austin villa@home 2019 team report. In *AAAI Fall Symposium on Artificial Intelligence and Human-Robot Interaction for Service Robots in Human Environments (AI-HRI 2019)*, November 2019.
2. Yuqian Jiang, Nick Walker, Justin Hart, and Peter Stone. Open-world reasoning for service robots. In *Proceedings of the 29th International Conference on Automated Planning and Scheduling (ICAPS 2019)*, July 2019.
3. Minkyu Kim, Miguel Arduengo, Nick Walker, Yuqian Jiang, Justin W Hart, Peter Stone, and Luis Sentis. An architecture for person-following using active target search. *arXiv e-prints*, pages arXiv–1809, 2018.

4. Justin W. Hart, Rishi Shah, Sean Kirmani, Nick Walker, Kathryn Baldauf, Nathan John, and Peter Stone. Prism: Pose registration for integrated semantic mapping. In *Proceedings of the 2018 IEEE/RSJ International Conference on Intelligent Robots and Systems (IROS)*, October 2018.
5. Justin Hart, Harel Yedidsion, Yuqian Jiang, Nick Walker, Rishi Shah, Jesse Thomason, Aishwarya Padmakumar, Rolando Fernandez, Jivko Sinapov, Raymond Mooney, and Peter Stone. Interaction and autonomy in robocup@home and building-wide intelligence. In *Proceedings of the AAAI Fall Symposium on Artificial Intelligence and Human-Robot Interaction (AI-HRI)*, October 2018.
6. Yuqian Jiang, Nick Walker, Minkyu Kim, Nicolas Brissonneau, Daniel S Brown, Justin W Hart, Scott Niekum, Luis Sentis, and Peter Stone. Laair: A layered architecture for autonomous interactive robots. In *Proceedings of the AAAI Fall Symposium on Reasoning and Learning in Real-World Systems for Long-Term Autonomy (LTA)*, October 2018.
7. Bo Liu, Yuqian Jiang, Xiaohan Zhang, Qiang Liu, Shiqi Zhang, Joydeep Biswas, and Peter Stone. Llm+p: Empowering large language models with optimal planning proficiency, 2023.
8. Krish Agarwal, Yuqian Jiang, Jiaheng Hu, Bo Liu, and Peter Stone. L3m+p: Lifelong planning with large language models, 2025.
9. Zichao Hu, Francesca Lucchetti, Claire Schlesinger, Yash Saxena, Anders Freeman, Sadanand Modak, Arjun Guha, and Joydeep Biswas. Deploying and evaluating llms to program service mobile robots. *IEEE Robotics and Automation Letters*, 9(3):2853–2860, 2024.
10. Piyush Khandelwal, Shiqi Zhang, Jivko Sinapov, Matteo Leonetti, Jesse Thomason, Fangkai Yang, Ilaria Gori, Maxwell Svetlik, Priyanka Khante, Vladimir Lifschitz, et al. BWIBots: A platform for bridging the gap between AI and human–robot interaction research. *The International Journal of Robotics Research*, 36(5-7):635–659, 2017.
11. Manuela Veloso, Joydeep Biswas, Brian Coltin, Stephanie Rosenthal, Tom Kollar, Cetin Mericli, Mehdi Samadi, Susana Brandão, and Rodrigo Ventura. Cobots: Collaborative robots servicing multi-floor buildings. In *2012 IEEE/RSJ International Conference on Intelligent Robots and Systems*, pages 5446–5447, 2012.
12. Alec Radford, Jong Wook Kim, Tao Xu, Greg Brockman, Christine McLeavey, and Ilya Sutskever. Robust speech recognition via large-scale weak supervision. In *International conference on machine learning*, pages 28492–28518. PMLR, 2023.
13. Joseph Redmon and Ali Farhadi. Yolo9000: Better, faster, stronger. *arXiv preprint arXiv:1612.08242*, 2016.
14. Alexander Kirillov, Eric Mintun, Nikhila Ravi, Hanzi Mao, Chloe Rolland, Laura Gustafson, Tete Xiao, Spencer Whitehead, Alexander C. Berg, Wan-Yen Lo, Piotr Dollár, and Ross Girshick. Segment anything. *arXiv:2304.02643*, 2023.
15. Glenn Jocher, Jing Qiu, and Ayush Chaurasia. Ultralytics YOLO, January 2023.
16. Qian-Yi Zhou, Jaesik Park, and Vladlen Koltun. Open3D: A modern library for 3D data processing. *arXiv:1801.09847*, 2018.
17. Shiyang Lu, Haonan Chang, Eric Pu Jing, Abdeslam Boularias, and Kostas Bekris. Ovir-3d: Open-vocabulary 3d instance retrieval without training on 3d data. In *7th Annual Conference on Robot Learning*, 2023.
18. Tianhe Ren, Shilong Liu, Ailing Zeng, Jing Lin, Kunchang Li, He Cao, Jiayu Chen, Xinyu Huang, Yukang Chen, Feng Yan, Zhaoyang Zeng, Hao Zhang, Feng Li, Jie Yang, Hongyang Li, Qing Jiang, and Lei Zhang. Grounded sam: Assembling open-world models for diverse visual tasks, 2024.

19. Adithyavairavan Murali, Balakumar Sundaralingam, Yu-Wei Chao, Jun Yamada, Wentao Yuan, Mark Carlson, Fabio Ramos, Stan Birchfield, Dieter Fox, and Clemens Eppner. Graspgen: A diffusion-based framework for 6-dof grasping with on-generator training. *arXiv preprint arXiv:2507.13097*, 2025.
20. David Coleman, Ioan Sucan, Sachin Chitta, and Nikolaus Correll. Reducing the barrier to entry of complex robotic software: a moveit! case study. *arXiv preprint arXiv:1404.3785*, 2014.
21. Balakumar Sundaralingam, Siva Kumar Sastry Hari, Adam Fishman, Caelan Garrett, Karl Van Wyk, Valts Blukis, Alexander Millane, Helen Oleynikova, Ankur Handa, Fabio Ramos, Nathan Ratliff, and Dieter Fox. Curobo: Parallelized collision-free robot motion generation. In *2023 IEEE International Conference on Robotics and Automation (ICRA)*, pages 8112–8119, 2023.
22. Nir Aharon, Roy Orfaig, and Ben-Zion Bobrovsky. Bot-sort: Robust associations multi-pedestrian tracking. *arXiv preprint arXiv:2206.14651*, 2022.

Cobot Software and External Devices

We use a custom-built mobile manipulator, featuring cobot base, and kinova arm. An Azure Kinect camera is connected.

Fig. 5: Cobot

Robot's Software Description

We are using the following 3rd party software:

- Object recognition: GroundedSAM
- People and activity recognition: YOLOv8
- Manipulation: GraspGen
- Knowledge Base: PostgreSQL
- Speech Recognition: Whisper, Vosk
- Text to Speech: Kokoro
- Planning and reasoning: Clingo, PDDLStream

External Devices

We are using the following external devices:

- NVIDIA Jetson Thor

Cloud Services

We are using the following cloud services:

- Large language model: GPT-4o