International Journal of Social Robotics
https://doi.org/10.1007/s12369-025-01234-5

UJI-Butler: A Symbolic/Non-symbolic Robotic System that Learns Through Multi-modal Interaction

Abdelrhman Bassiouny[1,2] · Ahmed H. Elsayed[1,4] · Zoe Falomir[3] · Angel P. del Pobil[1]

Accepted: 11 February 2025

Abstract

This paper introduces UJI-Butler, an innovative multi-robot framework that blends symbolic and non-symbolic artificial intelligence methods. Unlike previous systems, UJI-Butler integrates large language models (LLMs) with a knowledge base akin to RAG-based systems, while imposing logical reasoning on LLM-generated results. It facilitates multi-modal interaction with human users through speech, sign language, and physical interaction, fostering a human-in-the-loop learning paradigm. By acquiring new knowledge through verbal communication and mastering manipulation skills via human-lead-through programming, UJI-Butler enhances transparency and trust by incorporating human feedback during operations. Experimental results demonstrate that UJI-Butler's combination of symbolic and non-symbolic AI offers intuitive interaction and accelerates the learning process with experience. It adeptly stores and utilizes knowledge gained from verbal communication, recognizing hand gestures for requests. Additionally, UJI-Butler successfully performs user-taught physical skills and generalizes them to varying object sizes and locations. The explicit nature of acquired knowledge enables seamless transferability to other platforms and modification by human users. The code of the whole project is available on Github, in addition, video demonstrations of the UJI-Butler system are available online in a Youtube Playlist.

Keywords Human-robot interaction · Collaborative robotics · Cognitive robotics · Ontology · Reasoning · Knowledge bases · Symbolic AI · Multi-robot · Machine learning · Large language models · Sign language · Lead-through-programming · Life-long learning

✉ Abdelrhman Bassiouny
bassioun@uni-bremen.de

Ahmed H. Elsayed
ahmed.elsayed@dfki.de

Zoe Falomir
zfalomir@cs.umu.se

Angel P. del Pobil
pobil@uji.es

[1] Engineering and Computer Science Department, Universitat Jaume I, 12071 Castelló de la Plana, Spain

[2] Institute For Artificial Intelligence, Universität Bremen, Am Fallturm 1, 28359 Bremen, Germany

[3] Computing Science Department, Umeå University, 90187 Umeå, Sweden

[4] German Research Center for Artificial Intelligence GmbH (DFKI), 26129 Oldenburg, Lower Saxony, Germany

1 Introduction

Robots have always been imagined to help in household environments [1], but to the best of our knowledge, that has not been achieved yet, at least not at scale. For robots to enter households, a key feature is interaction; this is not required in regular industrial situations where most robots are found.

Intuitive human-robot interaction[1] is required to enable robots to enter our homes by helping with our elderly or automating our daily activities like cleaning, and cooking [2]. In addition, human-robot collaboration is key in small and medium-sized industries to perform complex and adaptive operations in the production line [3].

The UJI-Butler framework presented in this paper allows interaction with various robots in different modalities which include speech, sign language, and physical interaction. This multi-modality facet of UJI-Butler interaction has been cho-

[1] Intuitiveness here is meant to convey a sense of intelligence that the human user attributes to the robot.

Published online: 21 March 2025

sen to be as inclusive as possible to different groups of people with special requirements.

The UJI-Butler also introduces a way for robots to learn from humans throughout their existence by taking advantage of human-robot interaction situations. The ability to keep learning with new experiences during the robot's operational life is termed in the literature as Life Long Learning [4]. Life Long Learning (also referred to as continual or incremental learning) [5–7] requires that the learning agent keeps upgrading its knowledge so that it becomes more accurate (or at least maintains its accuracy) by updating the old learned data. However, if the learning process involves gradient descent update of weights of a neural network, the models suffer from catastrophic forgetting [8], since the models forget previous knowledge and/or skills. This is one of the reasons why the learning agent in the UJI-Butler framework uses *KnowRob*, an ontology-based knowledge base with reasoning capabilities [9] as a part of its architecture.

Moreover, to map between natural language and robot knowledge, the UJI-Butler agent incorporates GPT-3 [10] large language model in combination with its knowledge component to provide a more intuitive human-robot interaction.

UJI-Butler is a multi-robot system that also involves robot-robot communication and cooperation to complete a manipulation task (e.g. drink preparation by a UR5e robot) and delivery (by a Turtle-bot), including mapping, localization, and navigation in the environment while also avoiding obstacles. In this way, it can successfully deliver the requested preparation anywhere in the mapped environment.

Figure 1 illustrates the main parts of the UJI-Butler framework.

In light of this, the key capabilities of UJI-Butler can be summarized as follows:

1. Combining symbolic and non-symbolic artificial intelligence approaches to interpret and perform tasks requested by human users.
2. Learning new manipulation skills using Lead-Through programming techniques.
3. Acquiring new knowledge through verbal communication.
4. Ensure safety and trust by having a Human in the loop for verification and feedback.
5. Rule-based fact-checking and correction on the results of generative AI methods to minimize hallucinations.
6. Concurrent improvement of both the generative AI and the knowledge-based reasoning with new experiences.
7. Provide multi-modal interaction capabilities like voice and sign language to be more inclusive.
8. A perception system that can detect objects with different states and synonyms as indicated by the user request.
9. Multi-robot communication and execution for the requested tasks.

The rest of the paper is organized as follows. Section 2 describes the main components of UJI-Butler. It includes Sub-Sect. 2.1 which discusses the Knowledge base, Sub-Sect. 2.2 which presents how GPT-3 language models are used in combination with the knowledge base and the logical reasoning. Then, Sub-Sect. 2.3 explains how interaction works in the UJI-Butler framework. Sub-Sect. 2.4 describes how the UJI-Butler agent learns and acquires new knowledge. The perception of objects and sign language is explained in Sub-Sect. 2.5.

Section 3 presents the robotic platforms, tools, and the experimental setup in UJI-Butler. This section includes Sub-Sect. 3.1 which describes the setup and tools used to perform manipulation tasks. In addition, Sub-Sect. 3.2 presents multi-robot collaboration using Turtlebot-2 (Kobuki) for performing meal/drink delivery in collaboration with the robotic arm. Then, Sect. 4 shows the experimental results and provides a detailed discussion. Finally, the related work and the conclusions are presented in Sect. 5 and Sect. 6 respectively.

2 UJI-Butler Components

2.1 Knowledge Base

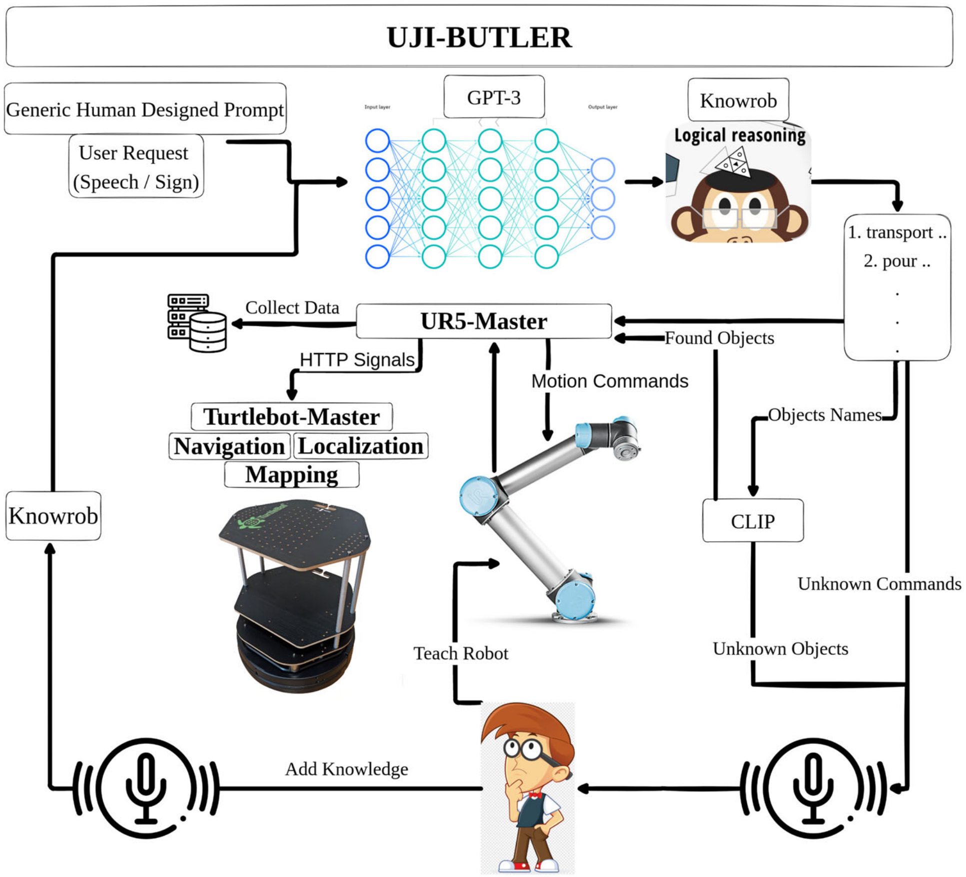
The UJI-Butler uses *KnowRob* [9], an ontology knowledge base with reasoning capabilities. *KnowRob* uses ontologies based on OWL [11] and SWI-Prolog [12] as a logic language for querying and reasoning. *KnowRob* uses a knowledge representation based on temporalized triples. These triples consist of subject, predicate, and object, with an additional element specifying the time frame when the statement holds true. Thus all facts can be described in this triple format with the ability to define predicates, subjects, and objects as needed by the application.

KnowRob has been chosen for several reasons: firstly it is an active project and has been active for more than a decade, secondly it has interfaces with common robotics tools like ROS (Robot Operating System), finally it is mostly used for tasks involving everyday activities which are the focus of a butler or a personal robot as is the case for UJI-Butler. Moreover, *KnowRob* seems to be the most popular knowledge base in service robotics for performing everyday activities [13].

SWI-Prolog is the query interface that is available for querying and interacting with KnowRob, also it is available through ROS services which makes it very convenient to use in the UJI-Butler framework.

In UJI-Butler, *KnowRob* is used as a source of knowledge that is needed by the robot to perform its tasks. *KnowRob* is also used for fact-checking on the results of GPT-3, and

Fig. 1 UJI-Butler Framework



also for adding new knowledge (e.g. adding new activities learned from experience). In that sense, what UJI-Butler adds to *KnowRob* is the ability to add more knowledge from robot experience gained through answering human requests related to tasks that are part of everyday activities (e.g. meal and drink preparation). This is facilitated by the use of large language models like GPT-3.

The UJI-Butler queries activities that can be performed, such as *Making-CoffeeTheBeverage* and *Making-TeaTheBeveage*. It can also query the objects and steps used to make these activities. For that, the UJI-Butler takes into account the objects needed to accomplish the activity as well as the outputs of the activity. The UJI-Butler knowledge base can be logically queried on all activities that involve preparing a drink or a meal:

```
subclass_of(B, Sb),
subclass_of(Sb, PreparingFoodOrDrink)
```

where *B* would be an activity, *Sb* would be the superclass of the activity, which would be either *PreparingFood* or *PreparingABeverage* and those two are a subclass of *PreparingFoodOrDrink*.

Any activity that involves acting on objects is expressed as an OWL restriction in Prolog as follows:

```
is_restriction(A, some(objectActedOn, C)),
subclass_of(B, A)
```

where *A* is an OWL description of the restriction, *C* is the object that is acted on during the activity, and *B* is again the activity. The activity is related to the restriction by being a subclass of the restriction. An activity also usually has an output, which in Prolog is also expressed as an OWL restriction:

```
is_restriction(A, some(outputsCreated, C)),
subclass_of(B, A)
```

where *A* is an OWL description of the restriction, *C* is the output created from the activity, and *B* is the activity. So, by using these simple logic Prolog queries one can retrieve information about the activity and its components.

The next section presents how the KB at UJI-Butler can be used with language models.

2.2 Retrieval and Insertion of Activities Using LLMs, Knowledge Base, and Rule-Based Fact-Checking

Large Language Models (LLM) require context to work as intended, so a very powerful solution can be combining knowledge from KBs with the ability of LLMs to process this information, complete it with missing information, or transform it into another representation. This is known as a RAG (Retrieval Augmented Generation) system [14].

RAG systems have noticeably increased the accuracy and reliability of LLMs by making them ground their outputs in facts represented in the knowledge sources provided to them. While RAG systems are more reliable, they are less expressive. RAG systems neutralize the ability of LLMs to answer more complex or completely different inputs/queries that cannot be answered through the provided knowledge base.

In UJI-Butler, the human user could ask for meals or drinks that are not previously known to the system (i.e. activities that do not exist in the knowledge base). The key question is how to make use of the power of LLMs while keeping them as reliable as possible. The answer to this in UJI-Butler is rule-based fact-checking. In UJI-Butler, the answers to human commands or requests can contain information that does not exist in the knowledge base, but still, they will abide by the rules defined in the reasoning system. This is a key difference and improvement that UJI-Butler provides over RAG-based systems. This is especially important when working in safety-critical domains such as social robotics.

The reason why GPT-3 was selected for UJI-Butler is that it was arguably the most powerful LLM that was freely available at the time of development of this work. Still, the LLM used is not a fixed part of UJI-Butler: it can be easily changed or upgraded with newer versions when they are available. Our focus is rather on how it is used with the knowledge base to design the prompt, which should be general to any other LLM that works with prompting. In that sense, UJI-Butler improves the use of LLMs by giving them the ability to use a knowledge base to design their prompt and also to make use of logic rules for fact-checking the results generated from the LLMs. For comparisons and results see Sect. 4.1.

2.2.1 Find the Requested Activity: From Text Instructions to Keywords

To find the requested activity by the user, the UJI-Butler framework applies the pipeline shown by Fig. 2. It uses GPT-3 as an LLM to extract keywords from text instructions given by the user, query the possible activities from the knowledge base, and finally find the best matching activity.

Firstly, a large language model is used as a keyword extractor from human speech input which is in natural language. For example, the sentence “We would love to have a cup of tea" is converted to “tea" and “cup" which can be used to find the best matching activity that has these keywords in its description or its components in the knowledge base. In this case, it should find the activity *Making-TeaTheBeverage*. To allow GPT-3 to give these specific keywords as an answer, a prompt is designed that contains a couple of examples showing the required behavior, an example prompt is as follows:

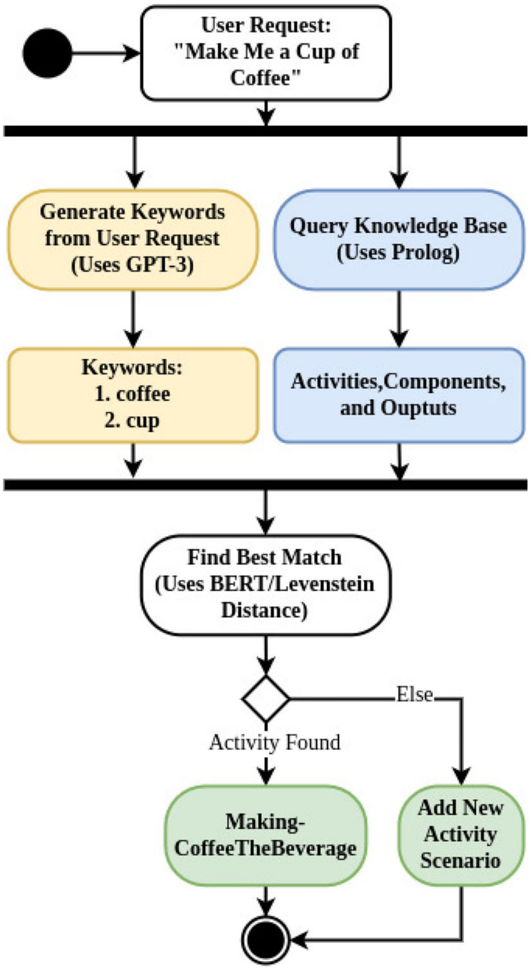

User Request:
"Make Me a Cup of Coffee"
Generate Keywords from User Request (Uses GPT-3)
Query Knowledge Base (Uses Prolog)
Keywords:
1. coffee
2. cup
Activities,Components, and Ouputs
Find Best Match (Uses BERT/Levenstein Distance)
Else
Activity Found
Making-CoffeeTheBeverage
Add New Activity Scenario

Fig. 2 Finding the requested activity pipeline for UJI-Butler

Q: Put lemon on water please:

1. lemon

2. water

Q: Prepare a meal for dinner please:

1. meal

2. dinner

Using the keywords that were the output of *GPT-3*, one can find matching activities in the KB by finding similar words in their descriptions or the objects that are used to perform this activity. The UJI-Butler finds the similarity between these keywords by applying the Levenshtein distance [15] between characters. However, sometimes two keywords can be not similar in writing but very similar in meaning (e.g. cup and drinking-mug), so in the cases when users say “drinking mug" then the Levenshtein distance might not help

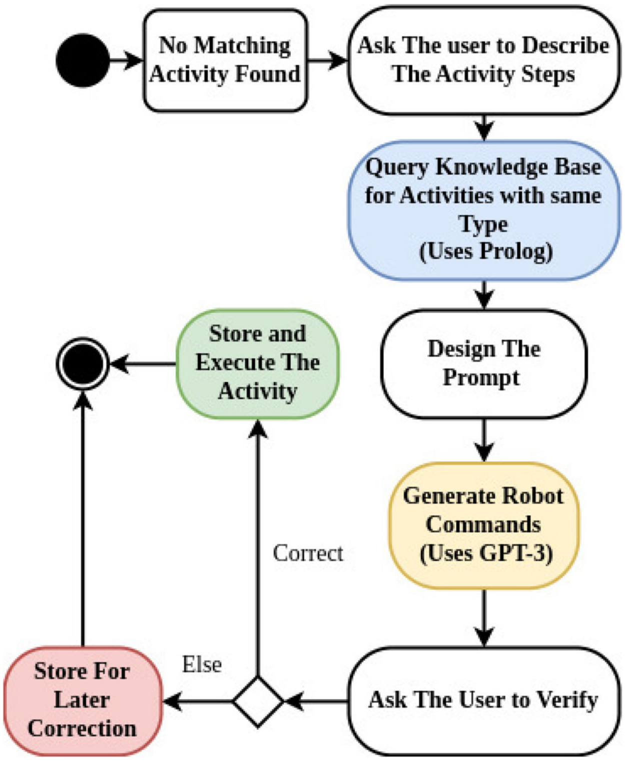

No Matching Activity Found
Ask The user to Describe The Activity Steps
Query Knowledge Base for Activities with same Type (Uses Prolog)
Design The Prompt
Generate Robot Commands (Uses GPT-3)
Ask The User to Verify
Correct
Else
Store and Execute The Activity
Store For Later Correction

Fig. 3 Adding a new unknown activity for UJI-Butler

find the instructions in the robot KB. In these cases, BERT embeddings are used to find the needed keywords as they provide statistical similarity by comparing the distance of their embeddings, but through experimentation, it was found that although Levenshtein distance lacks expressivity, it is the safer option. The Levenshtein distance would have a better true positive rate and lower false positive and this is more important in human-robot interaction because, in our opinion, a robot performing the wrong task (False Positive) is much more unacceptable than not performing the task at all (False Negative). For these reasons, the Levenshtein distance is chosen as the default method to use.

2.2.2 Adding a New Activity to UJI-Butler KB: From text to Commands

The UJI-Butler adds a new activity to its KB as Fig. 3 illustrates: first it asks the user about the activity type and what steps are involved in this activity, then it generates robot commands using GPT-3 with a prompt designed from the KB using similar activities. Finally, it verifies with the user the correctness of the generated commands; this improves safety and ensures transparency and trust while also guaranteeing that the data is correct before inserting it into the KB.

Here GPT-3 is used to convert user instructions describing the activity into steps that can be performed by a robot. Let us illustrate this with an example of a prompt:

```
Q:You put tea packet in a cup and then you put water in the cup:
1.transport(tea-packet, cup)
2.pour(water, cup)
container(cup)
Q:Put oats in a bowl, add milk, add honey, mix it all together, and enjoy:
1.transport(oats, bowl)
2.pour(milk, bowl)
3.pour(honey, bowl)
container(bowl)
```

Normally the last line in the prompt where the container is mentioned is not necessary, but this helps the fact-checking afterward to ensure that the container is an object that can contain the required objects in the transport and pouring steps above it. Currently, only one container is considered but it can be easily extended to handle more containers by adjusting the example prompt.

Further analysis and experimentation of different ideas that led to this final design are laid out in detail in section 4.1.

2.2.3 Closed Loop Reasoning

Closed-loop reasoning is the icing on top of the cake, since it constrains the GPT-3 output to conform with facts. These facts can come from two sources in the current version of UJI-Butler: (1) the knowledge base (KB), where an ontology exists and provides relationships and attributes to and between entities, and (2) the human user during operation when requested by UJI-Butler, thus serving as a human in the loop.

UJI-Butler is currently limited to correcting or verifying two kinds of situations: (i) referring to the container and (ii) clarifying the action and the source involved in it.

Regarding the container identification, that is, given that the type of the activity meal or drink is known, and given that the KB has containers for liquid (used for drink serving like a *cup* or a *glass*), and also containers for food (used for meal serving like a *plate* or a *bowl*), if the GPT output contradicts itself by assigning a food container for a drink or vice versa, the reasoning system can find that mistake when verifying, and then UJI-Butler would query the human user about changing the container or not. This is solved by using the following Prolog query:

```
holds(Container, transitive(subClassOf), Type).
```

where *Container* is the container in question, and *Type* is either a *DrinkingVessel*, if the activity type is a drink, or a *EatingVessel*, if the activity type is a meal. This query checks if this relationship holds transitively, i.e. if the *Type* is a superclass either direct or indirect of the container.

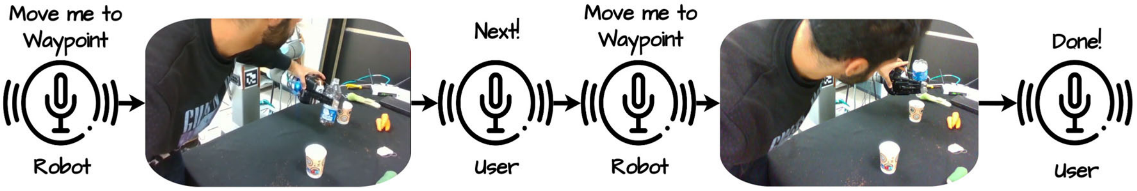


Fig. 4 User teaching the UJI-Butler how to *pour* from a bottle to a cup. On the left, the first waypoint, and on the right, the second waypoint. For results of the taught sequence on different objects see Fig. 17

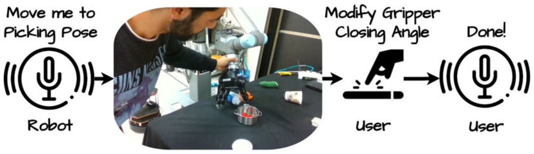


Fig. 5 Teaching UJI-Butler how to *pick* a cooking pot

The second situation where the UJI-Butler reasoner intervenes concerns the first argument of the *pour* and *transport* functions that the GPT assigns. The first argument for *pour* should not be a solid substance, while for *transport* it should not be a liquid or granular substance. In this case, a correction is done automatically –without asking the human user– by changing the function from *transport* to *pour* or vice versa. In Prolog, it is checked as follows:

```
holds(Object, transitive(subClassOf), Type).
```

where *Object* is the first argument of *transport* or *pour*, and the *Type* is the *LiquidTangibleThing*, if the function is *transport*, and *SolidTangibleThing*, if the function is *pour*. Thus, if the query answers true, then this is a mistake and the functions are changed.

2.3 The Interaction Between Users and UJI-Butler

In the UJI-Butler framework, the human and the robot can collaborate not just in physical tasks, but also by exchanging information using natural language (Fig. 5).

2.3.1 Teaching Using Human-Robot Physical Interaction

Physical interaction requires the human to move the robot arm to teach, or show it how to perform a certain action. In our approach the user specifies some waypoints to accomplish the desired motion (see Fig. 4 for an illustration showing a user teaching how to pour from a bottle to a cup). The reason behind it is to make the taught skill as general as possible to that object. A complete motion is useless if the object is displaced, while the waypoints offer more freedom to the robot's motion. For example, the pouring action is carried out using two waypoints, the first is the pouring position, and the second is the pouring angle. To make these waypoints general in any other position of the object, they are collected relative to the object centroid instead of being relative to the robot base or world frame. This is calculated using equations 1 & 2.

$$^{o}\mathbf{P}_t =^{r} \mathbf{P}_t -^{r} \mathbf{P}_o \tag{1}$$

where $^{o}\mathbf{P}_t$ is the tool position relative to the object centroid, $^{r}\mathbf{P}_t$ is the tool position relative to the robot base, and $^{r}\mathbf{P}_o$ is the object centroid position relative to the robot base.

$$^{o}\mathbf{q}_t = (^{r}\mathbf{q}_o)^{-1} *^{r} \mathbf{q}_t \tag{2}$$

where $^{o}\mathbf{q}_t$ is the tool orientation relative to the object, $^{r}\mathbf{q}_o$ is the object orientation relative to the robot base, and $^{r}\mathbf{q}_t$ is the tool orientation relative to the robot base.

Equation 1 yields the position of the robot tool relative to the object centroid and Eq. 2 finds the orientation (quaternion) of the robot tool relative to the object centroid (note that the object frame has the same orientation as the robot base frame). To use the stored position and orientation in a new object position, equations 1 & 2 are solved for $^{r}\mathbf{P}_t$ and $^{r}\mathbf{q}_o$ respectively.

Even though the UJI-Butler framework uses waypoints for robot teaching, it records the whole motion from the robot sensors and the camera, saving the collected data in the form of episodes to enable offline learning. The waypoints mode is convenient for instant online learning of a new task that is general for different positions of the object. Still, offline learning is required to generalize and make a model that uses all the data collected from the different skills on different objects.

2.3.2 Teaching Using Speech and Computer Interface

Knowledge exchange in the UJI-Butler framework can take place by using natural language or computer input (keyboard/touch screen). For example, when a user describes the steps to perform a certain task, s/he can use natural language, and the robot could speak to the user to ask them if it has detected an ambiguous situation (e.g. a required object is missing, or there are multiple instances of the same object, or there is a specific action that has not been done before, so it needs the user to show how to perform this new task, etc.). If the required information is numerical or digital it can be directly entered through a computer interface by the user. For example, the user indicates to the robot *"you need to sense a specific force in a specific direction"*, then the robot will ask the user to add this information through the computer interface.

Using Google Cloud Speech to Text,[2] a human user can communicate with the UJI-Butler to ask for a cup of tea or coffee, tomato, or whatever activity the user needs. If the robot knows how to execute it, then it will perform it right away, assuming that everything that is required is available within the robot's reach and field of view. Otherwise, it will ask the human about what is missing, either from the environment or from the robot knowledge itself. The human can then respond using natural language –if the answer is descriptive– or by showing and moving the robot physically –if the question concerns how to perform a certain action.

To have a more intuitive way of interacting with the robot, Google Cloud Text to Speech API[3] is used to let the robot ask questions to the human when needed, or to describe a problem that the robot currently faces. For example, the UJI-Butler can ask a human about the steps of an activity, or tell the human that it cannot detect a certain object, either because it is not there or because the robot model cannot detect this instance of the object. Moreover, the UJI-Butler can also ask the user to reduce the ambiguity in a situation: for example, if there are multiple cups in front of the robot, but only one of them is needed to make tea, the UJI-Butler would ask the human which cup to use.

The UJI-Butler framework incorporates the possibility of asking the user for help in carrying out unknown physical tasks. For instance, when the robot does not know how to pick or place an object, or do any action in general, it would ask the human to show how to perform this action. Also, it would ask if the task is similar to any of the already known actions –like picking or placing another object.

2.4 Learning

The UJI-Butler learns by augmenting its KB, which improves GPT-3 predictions, and it also collects data for offline learning.

2.4.1 *Augmenting Robot Knowledge Base*

After a user answers a question about *"what are the steps of a certain activity"* or *"what is the description of a certain object"*, the UJI-butler converts this information into robot commands and keywords that are stored in the *Knowrob* ontology for later usage. This allows knowledge to be gained transparently and explicitly. It also allows the knowledge to be modified by humans themselves later, either to add new knowledge or to modify existing ones.

Further reasoning can be triggered from the newly gathered facts (i.e. inference of similar situations, or inference of new relations that are predicted from the other existing relations). Importantly, this helps prevent catastrophic forgetting.

2.4.2 *GPT-3 Improved Prediction*

The larger the robot KB is, the better *GPT-3* predictions will be. This is simply because the *GPT-3* prompt can be modified from the KB to be more relevant and suitable for a certain prediction. For example, a human asks the robot to prepare coffee but the robot does not know how coffee is made. If still it knows that it is a drink, then it can prompt *GPT-3* with a drink that it knows about from the ontology, and tell *GPT-3* to generate commands that are similar to the ones that are used in the known drink. Thus, the more similar the prompt to the required prediction is, the better and more robust the *GPT-3* output will be.

2.4.3 *Data Collection For Later Offline Learning*

We have already seen how data about classes of objects and task steps are added directly to the ontology after user verification, giving the robot an online or instant learning capability. Moreover, all the data from human-robot interaction can be stored in a KB to be used later for offline machine learning. This is especially relevant for data coming from the robot sensors and actuators which cannot be included in the ontology but require motor skill learning.

2.5 Perception

The perception module at UJI-Butler includes object detection and sign language interpretation.

2.5.1 *Object Detection*

Perception needs in UJI-Butler cannot be satisfied with off-the-shelf state-of-the-art object detectors like YOLO [16]. The reason is that perception in UJI-Butler not only requires detection of objects in the environment but also linking them to what the human user is saying and/or asking for. In addition, objects in the environment have states that are important to recognize for the correct execution of the tasks (e.g. to place a tomato inside a cooking pot, the cover of the cooking put has to be removed).

[2] https://cloud.google.com/speech-to-text.

[3] https://cloud.google.com/text-to-speech.

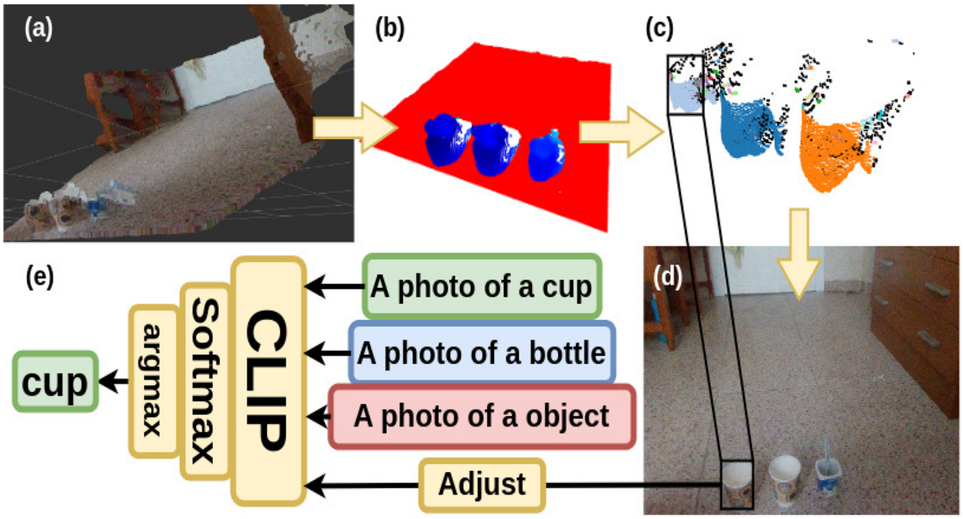


Fig. 6 Object detection pipeline: (a) the point cloud is retrieved (the environment is constrained in x,y,z), then the largest plane is segmented; (b) a density-based clustering is performed; (c) calculating a de-projection; (d) CLIP classification by passing the text sentences for each class and the de-projected images after adjusting/enlarging the object frame; (e) the 'object' is chosen to represent classes other than the requested ones. For results see Fig. 22, and Fig. 21

For that purpose, CLIP [17] is used. CLIP is a deep neural network model that can give a numeric measure for the similarity between text describing an image and the image itself. This allows grounding human descriptions in objects in the environment and would also allow for the recognition of object states by comparing their similarity to a text describing a certain state.

Since CLIP is not an object detector (i.e. it cannot find the location of objects in an image), it cannot be used alone. For this reason in UJI-Butler a perception pipeline was developed to first find all the objects in the working area of the robot, and then query these objects using CLIP. This perception pipeline complements the interaction experience of human users with UJI-Butler resulting in more versatility to perceive various situations and react accordingly. Moreover, UJI-Butler can ask for help depending on the perceived state/situation: e.g. it can recognize that the cooking pot is covered and it needs to be uncovered; since currently it cannot do that on its own, it would communicate with the human about this issue.

The object detection process at the UJI-Butler is illustrated by Fig. 6. The object detection pipeline was designed to be agnostic to the object class. This is a very important property since it allows the detection of unknown objects that are visible in the robot workspace. They can then be matched to what the human user is describing using CLIP, thus enabling UJI-Butler to handle new situations and learn new activities without requiring (or minimizing) the need for learning a new classification or detection model. The object detection steps are explained in detail below.

2.5.2 *(a) Constraining the Environment*

The environment is constrained by only allowing certain points that lie in a certain range in the x, y, and z axes. This is achieved by first transforming all the point-cloud points with respect to the robot base so that everything is described from the robot base as a reference system (RB$_{RS}$), making it independent of camera location or orientation. This helps in deciding on the needed ranges of the axis, since the base frame is aligned with the table frame/plane. An illustration showing the view before and after constraining the environment is shown in Fig. 7. Note that the environment is only constrained once (within a specific range of the x, y, and z-axis) assuming that the robot base and the working table are fixed in a place, while the camera can move anywhere as long as the table and the robot are in its field of view.

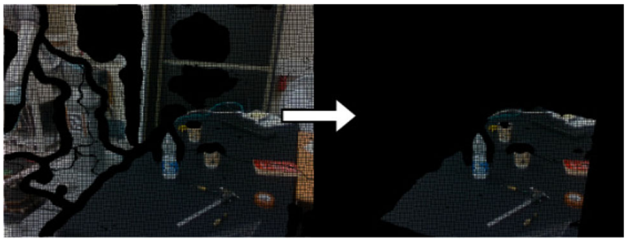


Fig. 7 The left image shows the overlayed depth image on the RGB image before constraining the environment (i.e. adjusting the lower and upper bounds of the x,y, and z values), the right one shows the same overlay of the depth and RGB images but after constraining the environment

Finally, a plane segmentation algorithm is used to separate the objects from the table plane as shown in Fig. 6(b). This algorithm is based on RANSAC [18].

2.5.3 *(b) Density Based Clustering*

The second step involves Density-Based Clustering (DBC) which is also provided by Open3D and it joins points that are densely packed together and defines them as a single cluster. Thus each object has one cluster of 3D points associated with it as shown in Fig. 6(c).

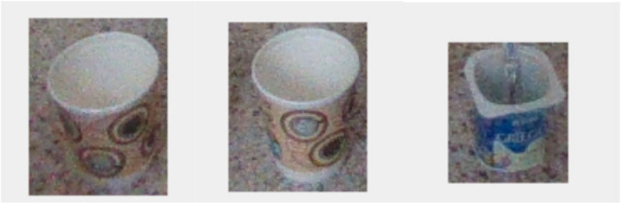


Fig. 8 Deprojected Objects to the RGB Image from the 3D Point Cloud shown in Fig. 6(a-d)

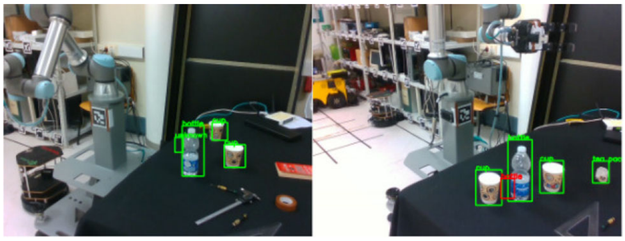


Fig. 9 Unknown object (left), rejected duplicate detection (right)

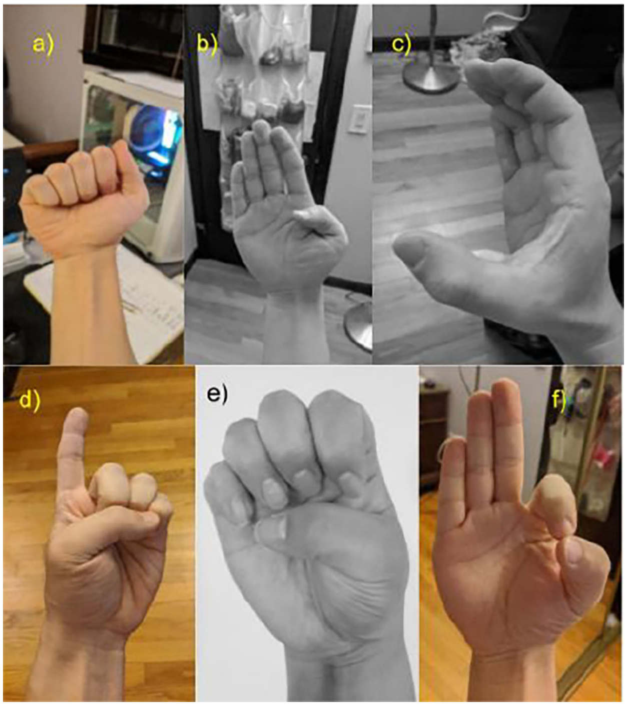


Fig. 10 American Sign Language Alphabet (samples from the dataset) from A to F

2.5.4 *(c) De-Projection*

After obtaining the 3D points belonging to each object, a de-projection step is required to get the RGB image part of each object. This is done by applying the camera's intrinsic and extrinsic parameters. This step is also required for the classification step by CLIP.

A correct de-projection is aligned with the object location and area in the RGB image as shown in the sample results in Fig. 8, which correspond to the inputs shown in Fig. 6 (a,d).

2.5.5 *(d) Classification Using CLIP*

CLIP [17] is a deep neural network that uses a transformer architecture. Its advantage is that it was trained using self-supervision on a very large internet dataset, and so it can classify thousands of classes of objects. CLIP is not just a classifier, it also tells the users how similar is a text description to an image. In a classification, a user can ask how similar is the image of an unknown object to the following text "a photo of a cup", and once a similarity value above a certain tuneable threshold is obtained, this image is classified as a cup. The challenge is how to define a proper question to CLIP so that it is useful in the UJI-Butler setting. If one uses just one sentence (e.g. "a photo of a cup"), the model could give a relatively high similarity between this sentence and an image containing something that looks like a cup (a concave cylindrical object) but is not exactly a cup. To deal with this issue, an extra sentence "a photo of an object" is added, which was found experimentally to perform well in this work. This extra sentence made the model less certain and distributed its probability nearly equally between the two sentences. Then, if there is a cup in the image, the model produces a much higher similarity with "a photo of a cup" than with "a photo of an object".

To handle the outlier clusters which are deemed as objects by the clustering algorithm, CLIP is asked if they belong to the classes needed for the task, if they are unknown, or if they are some other object (Fig. 9, left). On some occasions, parts of the same object are clustered as separate objects (this happens mostly when the object is transparent or reflective). This is solved using a non-maximum suppression technique which removes overlapping detected objects that are lower in confidence than the maximum threshold (Fig. 9, right).

2.5.6 *Sign Language Interpretation*

Sign language is not a universal method to communicate, it has many forms, such as American Sign Language (ASL), British Sign Language (BSL), and Australian Sign Language (Auslan) [19]. For the UJI-Butler framework, the dataset selected was the American Sign Language for Letters [20] which was labeled and annotated using bounding boxes around the gestures (signs). It has 26 classes for the English alphabet (Fig. 10 shows samples of this dataset) and the number of images in the original dataset was 784, which was enhanced with extra images from a YouTube video with about 250 images that explain how to produce American Sign Language signs. These images were annotated manually [21].

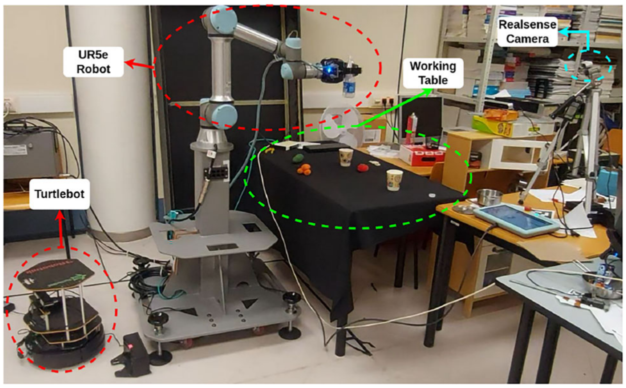


Fig. 11 Experimental Lab Setup for the UJI-Butler

YOLOv7 *(You Only Look Once version 7)* was used to detect sign language for the English alphabet. YOLOv7 is a real-time object detection, it predicts the bounding box and classifies the objects in the image or video in real-time [16]. YOLOv7 has outperformed other object detectors such as YOLOR, YOLOX, Scaled-YOLOv4, and YOLOv5 [16]. YOLOv7 Tiny is a smaller version of the YOLOv7 model that is designed to run faster and make predictions more quickly. YOLOv7 Tiny is preferred to ensure real-time object detection and decrease the training. It uses the Leaky ReLU (Rectified Linear Unity) activation function.

3 UJI-Butler Platforms and Experimental Setup

3.1 Manipulation Setup and Tools

The UJI-Butler framework includes a manipulation arm UR5e[4] which has force-torque sensing integrated, and joint torque measurements. The arm is endowed with a 3-finger Robotiq gripper.[5] An Intel-Realsense D435[6] stereo camera with infrared and laser is also included (Fig. 11 shows the manipulation setup).

3.1.1 *Robot & Environment Description*

At the UJI-Butler framework, for the manipulation to be performed successfully, the following goals should be met: (i) the robots have to avoid collisions, (ii) the manipulator should be able to plan with the tool-tip, and also (iii) be able to perform hand-camera calibration correctly. For these goals to be achieved, the correct modeling of the robot's kinematics and the environment is essential.

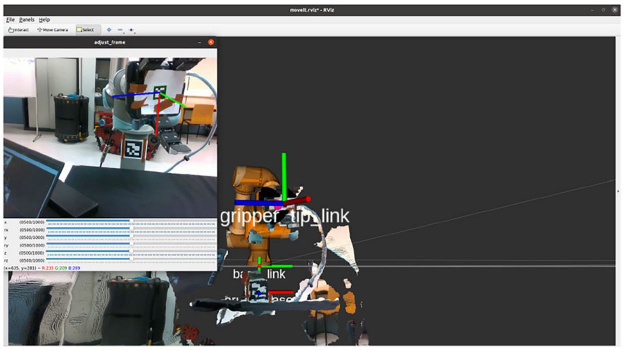


Fig. 12 Calibration Frames. The RGB image shows the detected aruco frame on the gripper palm, and the 3D visualization shows the URDF setup with the overlaid point cloud and the aruco frame of the gripper in 3D. This shows the correct calibration and detection of the aruco frames

It is important to note that due to time constraints in the development of the project, only static parts of the environment are included in the description, although some objects (e.g., cups, coffee, or tea) are dynamic and might change their position. So under this constraint, usual collisions with these objects can occur, but this does not hinder the demonstration of the project given that the robot will not collide with well-placed objects during operation. To this end, the robot, the robot mounting base, the gripper, and the working table are the only objects that are included in the description. The description is a URDF file that contains all the links, all robot joints, and all the other fixed objects that are attached as fixed joints and are considered parent links to the robot base.

3.1.2 *Robot-Camera Calibration*

After setting up the robot & environment description with the gripper attached to the manipulator, the UJI-Butler camera can detect the gripper tip by detecting an aruco[7] tag on the gripper palm area. OpenCV[8] is used for the aruco detection as seen in Fig. 12. Using the URDF, the transformation between the gripper palm and base can be calculated. Using the aruco detection, the transformation from camera to gripper palm can be found. By using these two transformations, the camera-to-base (robot base) transformation can be calculated using Equation. 3.

$$^{b}\mathbf{T}_c =^{b} \mathbf{T}_g *^{g} \mathbf{T}_c \tag{3}$$

where $^{b}\mathbf{T}_c$ is the camera to robot base transform, $^{b}\mathbf{T}_g$ is the gripper palm to robot base transform, and $^{g}\mathbf{T}_c$ is the camera to gripper transform.
For the sake of simplicity, another aruco was attached to the robot base. Thus, once the camera-to-base transformation is

[4] https://www.universal-robots.com/products/UR5-robot/.

[5] https://robotiq.com/products/3-finger-adaptive-robot-gripper.

[6] https://www.intelrealsense.com/depth-camera-d435/.

[7] https://chev.me/arucogen/.

[8] https://opencv.org/.

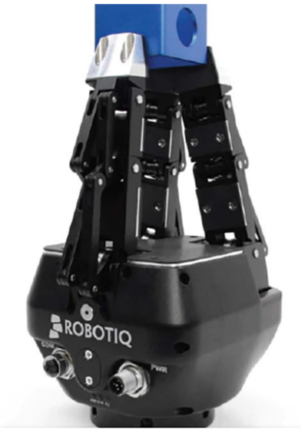


Fig. 13 Robotiq 3F Gripper in Pinch Mode

calculated, then the base aruco can be detected and its transformation to the base link of the robot can be calculated. The key here is that this transformation between the base link and the aruco attached near the robot base is static, so it needs to be calculated only once. So, whenever the camera-to-base transformation is needed for any camera position or orientation, the only thing required is to detect this aruco attached to the base. This calculation is obtained using Equation. 4.

$$ {}^{b}\mathbf{T}_{c} = {}^{b}\mathbf{T}_{ba} * {}^{ba}\mathbf{T}_{c} \tag{4} $$

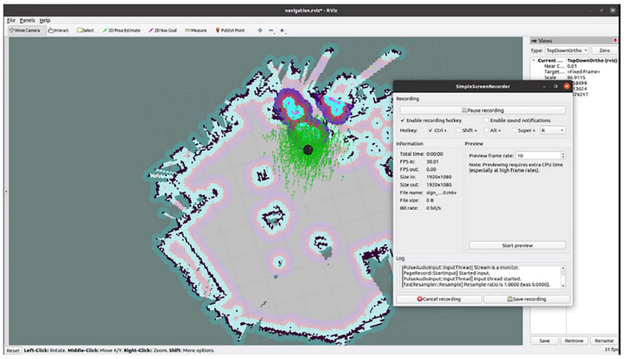
where ${}^{b}\mathbf{T}_{c}$ is the camera to robot base transform, ${}^{b}\mathbf{T}_{ba}$ is the base aruco to robot base transform, and ${}^{ba}\mathbf{T}_{c}$ is the camera to base aruco transform.

3.1.3 *Motion Planning*

The UJI-Butler framework uses *Moveit!*[9] for motion planning, in particular, it uses Cartesian path planning since there are Cartesian constraints during robot motion. For example, when carrying a cup, it must always stay in the upright position so that, if it contains a drink, it does not spill it.

3.1.4 *Gripping*

For almost all UJI-Butler gripping tasks a pinch grip is used (Fig. 13), but with different orientations depending on the object location from the base and also on the object height. Note that short objects can make the gripper collide with the table so a pitch shift is needed to avoid collisions. For gripping cups, a better option is to use a horizontal power grip instead, since it provides more support when the cup is filled with a drink.



Type: TopDownOrtho

Current ... TopDownOrtho (rviz)
Near C... 0.01
Target... <Fixed Frame>
Scale 86.9115

☑ Enable recording hotkey ☐ Enable sound notifications

Hotkey: ☑ Ctrl + ☐ Shift + ☐ Alt + ☐ Super + R

Total time: 0:00:00
FPS in: 30.01
FPS out: 0.00
Size in: 1920x1080
Size out: 1920x1080
File name: sign_...0.mkv
File size: 0 B
Bit rate: 0 bit/s

Preview frame rate: 10

Note: Previewing requires extra CPU time (especially at high frame rates).

```
[PulseAudioInput::InputThread] Stream is a monitor.
[PageRecord::StartInput] Started input.
[PulseAudioInput::InputThread] Input thread started.
[FastResampler::Resample] Resample ratio is 1.0000 (was 0.0000).
```

Left-Click: Rotate. Middle-Click: Move X/Y. Right-Click: Zoom. Shift: More options.

31 fps

Fig. 14 Navigation stack on ROS1 working at UJI-lab

3.2 Multirobot Platforms: Turtlebot2 (Kobuki) Collaborating with UR5e in a Delivery Service

While the UR5e robot manipulator is preparing the beverage/meal, the Turtlebot2 waits for the drink to be served on top of it. After that, the robot goes to a predefined location on the map where the beverage/meal should be delivered.

3.2.1 *Navigation and SLAM*

The Turtlebot2 uses GMapping [22, 23] to build the map of the laboratory, so that this map is used later for navigation. Astra RGB-D camera was used in this context and only the depth topic was used in GMapping acting as a lidar. The map shown in Fig. 14 was used for robot navigation at the UJI-lab scenario.

3.2.2 *Visual Servoing*

Visual servoing is used for the Turtlebot to move to a location near the manipulator, so it can receive the beverage on top of it. The visual servoing is carried out using a single Aruco marker. The interaction matrix is calculated along with the error, which is pixel error, and the difference in x, y, and z. This error is converted into velocities, which are passed to the robot so that it can move to this defined location. The robot was able to go near a place for its ultimate goal but not with 100% accuracy. This did not hinder the UR5e manipulator from carrying out the mission, so further improvements in the visual servoing were not required at the moment. The Aruco marker was placed on the UR5e's base as shown in Fig. 15.

3.2.3 *Autonomous Docking*

The mobile robot can dock using three infrared (IR) sensors. For that, a predefined location on the map near the docking

[9] https://moveit.ros.org/.

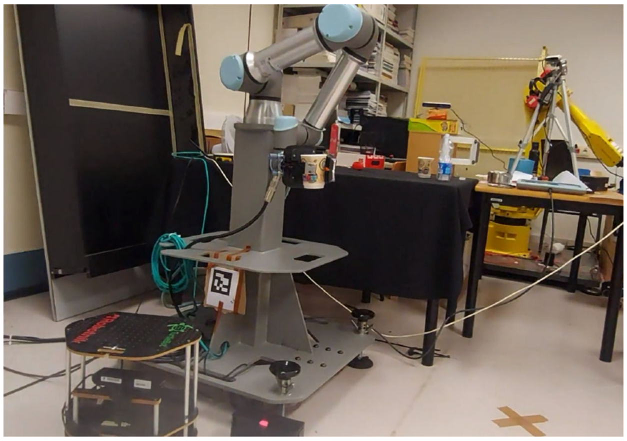


Fig. 15 Turtlebot2 (Kobuki) performing visual servoing on Aruco marker on the base of the UR5e robot, while the UR5e robot is placing the drink on top of the Turtlebot

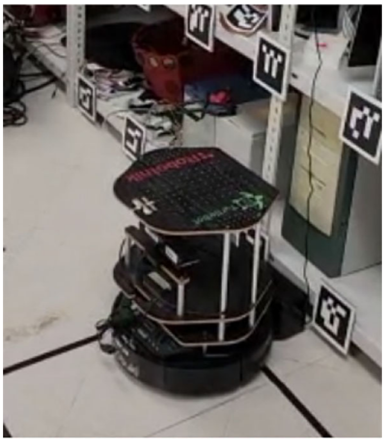


Fig. 16 Turtlebot2 (Kobuki) charging at dock station

station is needed, as the IR sensors have limited range. Figure 16 shows the Turtlebot2 (Kobuki) in the docking station.

4 Experimentation

The UJI-Butler framework was tested in the lab using a UR5e manipulator arm, which has integrated force-torque sensing, and joint torque measurements. An Intel-Realsense D435[10] camera was also used, which is a stereo camera coupled with infrared and laser distance sensors. A 3-finger Robotiq gripper[11] was also part of the UJI-Butler framework. Figure 11 shows the setup.

4.1 Results on Reasoning Using KnowRob & GPT-3

The UJI-Butler receives information or requests from the user. Given this information on one activity from the user, one can divide the type of information into three categories:

- **Request**: a request from the user expressed in language e.g. asking UJI-Butler to make a meal or a drink.
- **Description**: Language description of the steps required to perform an activity.
- **Description & Request**: A combination of both. This is done using the following template *"Request; To do it; Description"*.

Given a request for an activity that the UJI-Butler does not recognize, UJI-Butler will ask the user to describe the steps required to perform the activity in natural language. It would have been better if UJI-Butler with the power of LLMs like GPT-3 would be able to perform the activity given only the request without the need to ask the human user for further information. However, through experimentation, it was found that providing GPT-3 with only the request results in a much lower accuracy of the generated steps. For this reason, it was decided to compare the results given the two types of information and whether experience could improve or mitigate these issues. In light of that, a multi-dimension comparison of the possible information and the application of reasoning and experience was conducted as seen in Fig. 18 and Table 1.

Furthermore, when having multiple examples to use from the KB, one would think that using as many examples as the prompt can handle would yield the best results. While experimenting with GPT prompts, it was observed that, the more concise and relevant the prompt, the more accurate and predictable the result. The point is not only how large and varied the prompt is, but also how concise it is [12]. To put this idea to the test, three more prompt design categories were added for comparison:

- **Fixed Prompt (Baseline)**: using a fixed prompt that is human-designed and does not change at all.
- **Augmented Prompt**: using a prompt that gets augmented with new examples obtained from new knowledge of any activity.
- **Type Filtered Prompt**: This could be either fixed or augmented, but with the addition that the prompt contains only activity examples that belong to the same type as the requested activity.

[10] https://www.intelrealsense.com/depth-camera-d435/.

[11] https://robotiq.com/products/3-finger-adaptive-robot-gripper.

[12] *"Expressing what needs to be said without unnecessary words"* - https://dictionary.cambridge.org/dictionary/english/concise.

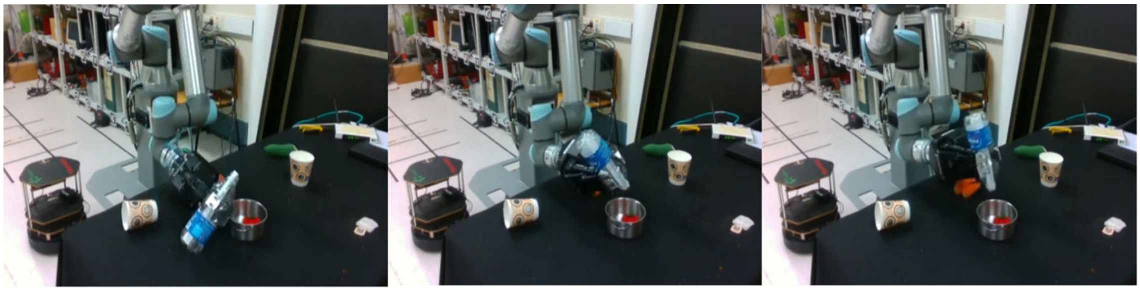


Fig. 17 Pouring to cooking pot from teaching that was performed on a cup, sequence from left to right, the leftmost image being the first waypoint registered during teaching, and the rightmost image is the second waypoint, and the middle image is an example point from the generated plan by the motion planner from waypoint 1 to waypoint 2

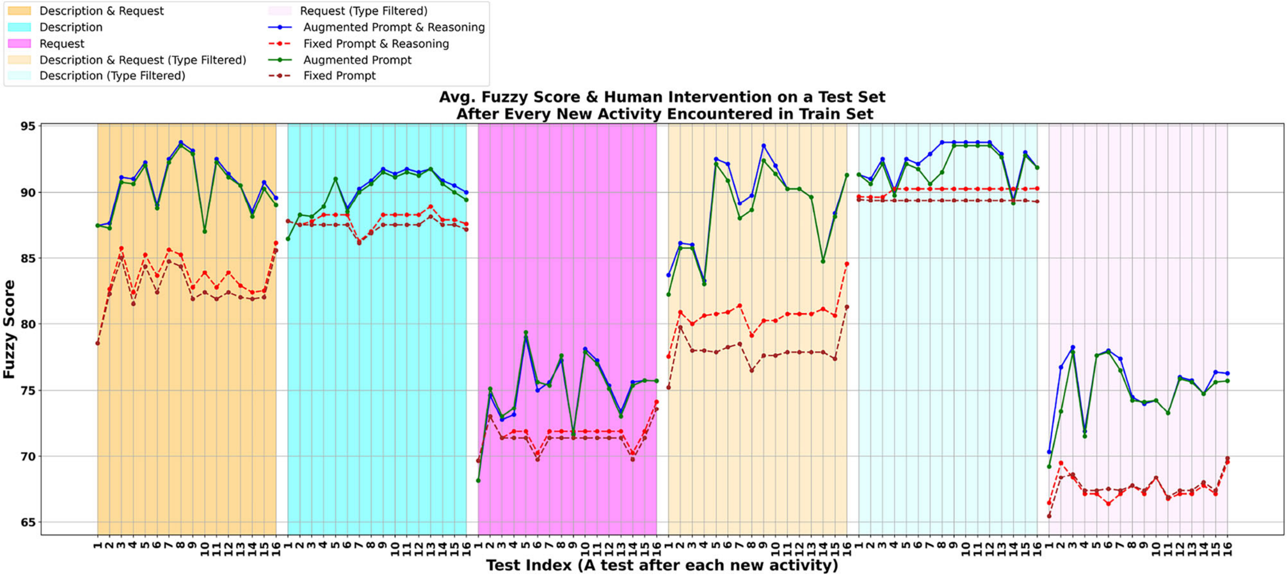


Fig. 18 Results regarding the different prompts and reasoning options used: average *Fuzzy Score* for each test on the *Test Set* after training using the *Train Set* and 24 configurations resulting from combining description-requests inputs with the type of prompts. In vertical, the description-requests communication categories: Description & Request (in orange), Description (in cyan), Request (in magenta), Description & Request filtered by type *Meal/Drink* (in light orange), Description filtered by type *Meal/Drink* (in light cyan), Request filtered by type *Meal/Drink* (in pink). For each category, lines connect the average *Fuzzy Scores* obtained as a result of the tests using 4 kinds of prompts: Augmented Prompt & Reasoning (in dark blue), Fixed Prompt & Reasoning (in red), Augmented Prompt (in green) and Fixed Prompt (in dark red)

Figure 18 shows the experimentation results using different configurations of prompts (using GPT-3 with a Knowledge base) and reasoning (with a human in the loop). A total of 24 configurations are tested which result from 6 possible input communication situations to describe each activity (*Description & Request*, Description, Request, *Description & Request* filtered by type *Meal/Drink*, Description filtered by type *Meal/Drink*, Request filtered by type *Meal/Drink*) and 4 possible prompt features: Augmented Prompt as Experience is Gained with Reasoning & human in the Loop, Fixed Prompt with Reasoning, Augmented Prompt with no Reasoning, and Fixed Prompt with no benefit of Experience). The *Train Set* consists of 16 tests composed of 8 activities asking for meals and 8 activities asking for drinks. And the *Test Set* is composed of 16 activities, equally distributed between meals and drinks.

Note that in all categories, the Augmented prompts (dark blue and green lines) are performing better than fixed prompts (light and dark red lines). And also note that the more successful communication results are obtained when a Description is provided, either alone or together with a request (vertical columns 1, 2, 4, 5 in Fig. 18). The low successful results are obtained when a request is provided without a description (vertical columns 3 and 6 in Fig. 18).

From the results note that when one compares *Description & Request* versus *Description* and versus *Request* independent of whether type filtration is used or not, one finds that *Description* is better by at least 4% when the GPT prompt is

Table 1 *Fuzzy scores –Mean* (μ) & *Standard deviation* (σ)– for each of the 24 configurations across all 16 tests on the *Test Set*, where each one of the 16 tests is done after each new activity encountered from the *Train Set*. The coloring of the rows and columns headings' are the same as in Fig. 18. The three top μ values and the top averages are highlighted in bold, and the three worst values are highlighted in italic

Prompts	Description & request				Description				Request				$\overline{X}$	
	Not F.		Type F.		Not F.		Type F.		Not F.		Type F.			
	μ	σ	μ	σ	μ	σ	μ	σ	μ	σ	μ	σ	μ	σ
Augmented & reasoning	**90.59**	2.05	88.91	3.14	90.13	1.54	**92.39**	1.32	74.9	2.59	75.34	2.16	**85.36**	2.13
Augmented	90.23	2.02	88.4	3.13	89.94	1.48	**91.89**	1.31	74.96	2.61	74.84	2.24	85.04	2.13
Fixed & reasoning	83.51	1.82	80.64	1.34	87.89	0.59	90.14	0.24	71.71	1.01	*67.55*	0.93	80.24	0.99
Fixed (baseline)	82.7	1.69	77.97	1.25	87.41	0.41	89.37	0.03	*71.3*	0.99	*67.65*	0.9	79.4	0.88
$\overline{X}$	86.74	1.9	83.98	2.22	88.84	1.0	**90.95**	0.72	73.22	1.8	71.34	1.56	-	-

fixed (i.e. when the prompt does not increase or change with experience) compared to *Description & Request* and by 20% compared to *Request* which shows that

Description is superior. Thus, extra unnecessary information gives a disadvantage by increasing the ambiguity of the task that makes the results of GPT unstable, which is clear when the standard deviation of *Description & Request* is compared to that of *Description*, which is also very clear from the graphs shown in Fig. 18, and from the averaged values of the standard deviation shown in Table 1.

The activity type can be used to further reduce the prompt size while keeping only the most relevant examples. This technique seems to improve the results when using only the *Description* of the activity (see Table 1 column *Description*). This suggests that a selective in-context approach to the available examples yields better results than considering all examples.

Note that GPT cannot make use of the activity type (even when given explicitly) as efficiently as the knowledge base. The knowledge base inherently distinguishes between different types, and it can infer where to look, but GPT cannot do that. Thus, improving GPT results by only giving examples of the requested type helps in confining its results to that type –although this does not enforce it on GPT– so it greatly decreases the probability of GPT giving results for a different type of activity.

A crucial advantage to UJI-Butler compared to the traditional RAG-based systems is closed-loop reasoning, and human-in-the-loop, which are the last steps where the GPT output is verified and corrected against known facts. If a mistake correction can be done from the knowledge base, it is done, if not, and it is recognized as a mistake, then UJI-Butler asks the human user to correct it. This allows UJI-Butler to make use of the power of LLMs without requiring them to only get data from the knowledge base, but also it can generate new knowledge and the reliability will come from the fact-checking step which ensures that the results follow the logical rules defined. Almost all the results shown in Fig. 18 and in Table 1 show that reasoning gives a better result, also it is important to note that the reasoner makes use of the type of activity as well in order to detect mistakes in container type. Example corrections of the results are shown in Table 2.

Another important measure is the number of human interactions (verbal or physical) needed to learn a new task, this could be a feasible measure for the intuitiveness of the interaction. Table 3 shows the mean and standard deviation of the number of human interventions on the three types of input (Description & Request, Description only, and Request Only) and two types of prompts (Augmented Prompt of previous relevant experience vs Fixed Prompt). The results in Table 3 show that augmenting the prompt with experience has approximately 8 times lower mean number of human interventions compared to using a fixed prompt in the case where the task *Description* is used as input (which was found to be the best type of input for better accuracy), and a 130 times lower as well in case of *Description & Request*. Meanwhile, in the case of *Request* only, the interventions are less in the augmented prompt compared to the fixed prompt. This is believed to be so because the *Request* only input of new activities does not contain the information needed to perform the activity, and thus augmenting the prompt with this type of input

increases the uncertainty of the LLM –as seen from the high standard deviation marked in red in Table 3– and thus worsens the results, requiring more human intervention.

4.1.1 Scalability of the Knowledge Base and GPT-3 Combined System

The knowledge base in UJI-Butler is used for storing knowledge that is gained from experience. It is not constrained by the examples provided so far in this work. The other usage of the knowledge base is for checking the adherence of the results generated by the LLM to the logical rules by applying reasoning to them. These rules are currently human-informed or human-designed and can be further enhanced by adding more rules as required for new situations, which would inherently improve the system. Future work could, for example,

Table 2 Example corrections of GPT-3 output by the reasoning system

Steps description	Output Transport (milk, *mug*)	Correction Pour
Pour milk into a *mug*	Container (*mug*)	-
	Transport (chocolate-powder, *bowl*)	Cup
You have to use chocolate powder, and milk	Pour (milk, *bowl*)	Cup
	Container (*bowl*)	Cup

focus on the ability to extract these rules from the knowledge contained in LLMs and then verify their correctness manually.

Moreover, the knowledge acquired from new activities can be semantic knowledge like in the case of the steps for performing an activity. This type of knowledge is also not constrained by the type of activity, and so it should be able to scale with more complex activities.

The scalability of the LLM used in UJI-Butler can be of two types, first by having more experience the system will have better and more relevant prompts to new situations, thus improving the accuracy and usage of the system. The second is by having a model more powerful than GPT-3 (note that the LLM used is not a fixed part of the system and it can be replaced in the future.

The knowledge base used in UJI-Butler is not robot-dependent. The execution steps are activity specific, not platform specific. Thus, UJI-Butler knowledge and reasoning system can be transferred with ease to another robot platform.

4.2 Results on Skill Teaching

Figure 17 shows the robot pouring from a bottle to a cooking pot. It is interesting to note that this was not taught on a cooking pot, and UJI-Butler was aware that it had not poured into a cooking pot before. In consequence, it asked the user whether it was similar to one of the skills it has already learnt. The user answered with the name of the skill that it was similar to, which was: "pouring from bottle to cup". So, the UJI-Butler executed the operation successfully. This was possible because the relation between the robot pose to the centroid of the tool objects was stored. Since the pouring action is carried out with respect to the center of the object that is being poured into, it generalized well to this case.

4.2.1 Scalability of Skill Teaching

The knowledge gained and stored in the knowledge base during skill teaching is numerical data, in particular poses of the key points recorded during the human-physical interaction. If a task requires that the number of key points increase significantly, this would be cumbersome for the human user, and would make no sense to store such information in the knowledge base. This is the case for activities that require a specific motion profile instead of specific key points. A mitigation of this issue would be the pre-definition of the motion profiles as a possible skill that could be performed by the robot, and then just requiring the start and end key points of such motions in different situations. Then, only these key points need to be stored in the knowledge base.

Obviously, the method employed in this work for skill teaching does not target dexterous manipulation. Instead, it deals with the variations of some key points depending on the context of the task. Moreover, this system depends on the existence of pre-defined procedures for accomplishing the dexterous part of the manipulation while using these procedures correctly in the different contexts or situations that are extracted from the knowledge base.

4.3 Results on Object Detection

An extra robustness to the vision pipeline implemented here is the independence of the different orientations of the objects (or the camera) when segmenting objects that are on the table (see Fig. 19). However, a bottleneck comes from the classification of CLIP which is actually affected sometimes by the orientation of objects.

Figure 20 shows the results of asking CLIP to classify a cooking pot with lid and another without a lid. Note that it was able to correctly classify it after some tuning of the similarity thresholds.

Figure 21 shows our experimentation when different users named the same object differently (e.g. the cooking pot or the cooking bowl). The UJI-Butler framework was able to handle that case using CLIP.

Figure 22 shows the results of the object detection pipeline (Fig. 6).

The UJI-Butler perception was tested against 7 classes of objects (cup, bottle, tea packet, tomato, cooking pot, lid, bowl), still, the system shows the ability to be versatile to many more classes since the tested classes were not predefined in any way.

The system accuracy is affected by the object size since small objects tend to have fewer points in the point cloud and if they are smaller than a certain threshold they will be considered noise. Also, transparent objects cannot be detected

Table 3 *Mean* (μ) & *Standard deviation* (σ) *Human Intervention* across all 16 tests on the *Test Set*. The lower the mean intervention value the better, also for the standard deviation. The coloring of the rows and columns headings' are the same as in Fig. 18, All of the inputs are type filtered and have a reasoning correction step performed on the results

Prompts	Description & request		Description		Request		$\overline{X}$	
	μ	σ	μ	σ	μ	σ	μ	σ
Augmented	0.44	0.61	**0.12**	0.33	1.06	*1.03*	**0.54**	0.66
Fixed	*3.56*	0.61	1.0	**0.0**	0.31	0.46	1.62	0.36
$\overline{X}$	2.0	0.61	**0.56**	0.16	0.68	0.74	–	–

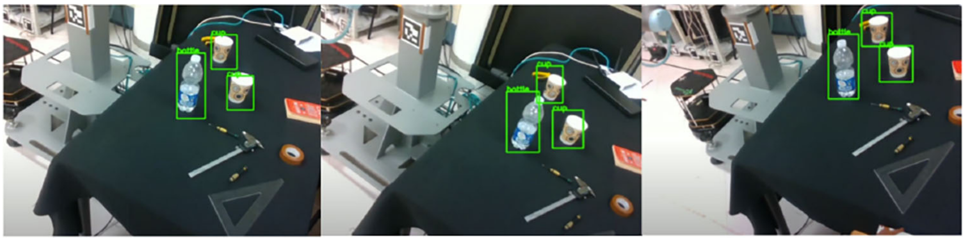


Fig. 19 Object detection pipeline working well with different camera orientations

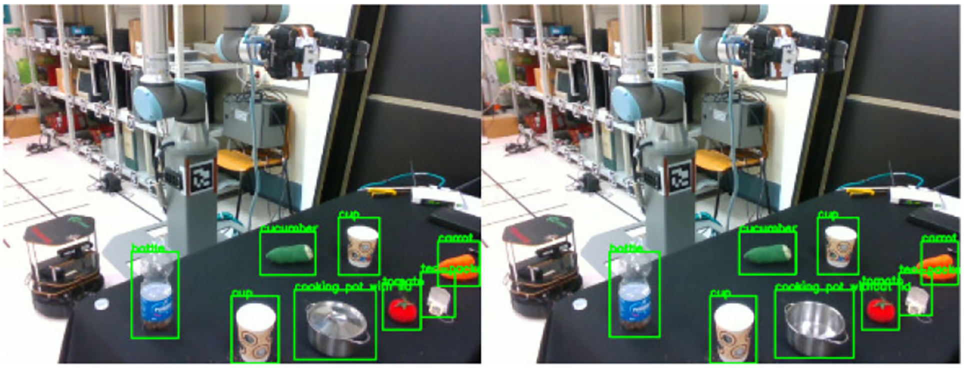


Fig. 20 Cooking bot with the lid on (left), cooking bot with the lid off (right)

very well because they affect the depth image used to generate the point cloud. A better option for transparent objects is the use of the RGB image also in the detection phase and not only in the recognition phase.

4.4 Results on Sign Language Interpretation

The YOLOv7 model was tested on UJI-Butler, where the robot was instructed by a human to prepare either tea or coffee. Figure 23 depicts an operator using sign language to instruct the robot to prepare a cup of tea.

The YOLOv7 algorithm demonstrated good performance on an American Sign Language (ASL) dataset, with an overall accuracy of approximately 60%. When the model was applied to a real camera feed, the accuracy decreased slightly but remained relatively high. Figure 24 shows that the model achieved an accuracy of 61% on the "W" letter from the dataset and 42% on the camera feed.

The lower accuracy on the real camera feed might be due to differences in lighting conditions between the dataset and the real camera feed. In particular, the lighting in the real camera feed was lower, which may have affected the model's performance.

5 Related Work

This section explains the contribution of the UJI-Butler framework by differentiating it from other works in the literature.

A number of research works have dealt with the challenge of a robot butler before [24–27]. A humanoid butler with a wheeled mobile base was located inside a smart home with sensors and smart appliances [24]. It could interact with human users by voice as part of its human-robot interface but without instant learning from the interaction with the user. Care-O-bot [25] is a robot butler with one hand that is designed to do tasks like Fetch-and-Carry for home and elderly assistance, one of its features is the ability to interact using speech. In their future research, they mentioned some interaction capabilities that the more advanced robot butlers should have, two of which are present in UJI-Butler:

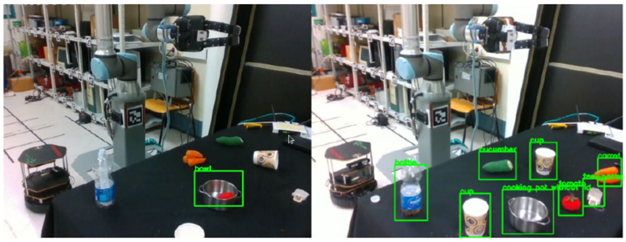


Fig. 21 CLIP can correctly classify the same object when this object could have different names by which it could be classified

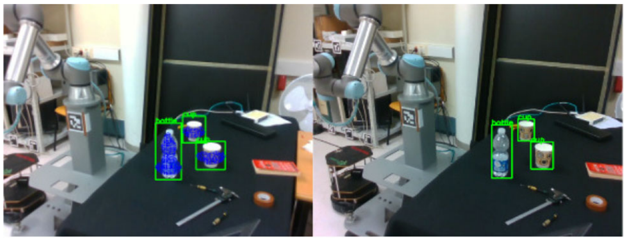


Fig. 22 CLIP Detected Objects, with overlapped points (left), without overlapped points (right)

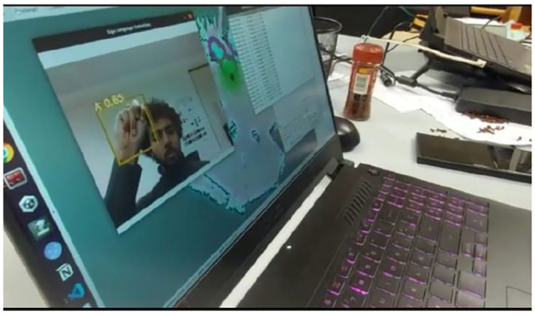


Fig. 23 An operator using sign language to instruct the UJI-Butler to prepare a cup of tea

(1) the ability to learn instantly during interaction by making use of episodic or case-based learning techniques (one-shot learning), and (2) the combination of the state-of-the-art language models with a conceptual framework/ontology that is grounded in the given domain of cognitive behavior. M-Hubo [26] is a humanoid butler with a wheeled base, it could perform tasks like fetching and serving drinks, but it lacks user interaction features and thus cannot learn from users. E-Butler [27] is a mobile robot without a manipulator that could only deliver amenities in hotels, it was connected to a local database from which it received new delivery orders from users. The users could only give orders to E-Butler using a smartphone application designed to communicate with the database, so there is no direct interaction with the user.

With the advent of big data and deep learning, purely deep learning based approaches have shown promise in language understanding and robotic task execution like in SayCan [28]. But, they can face limitations in real-world robotics applications. Deep learning models often lack interpretability and reasoning capabilities, making it difficult to understand their decision-making process and ensure safety in dynamic environments. Additionally, their reliance on large datasets can lead to data dependency and the potential for hallucinations.

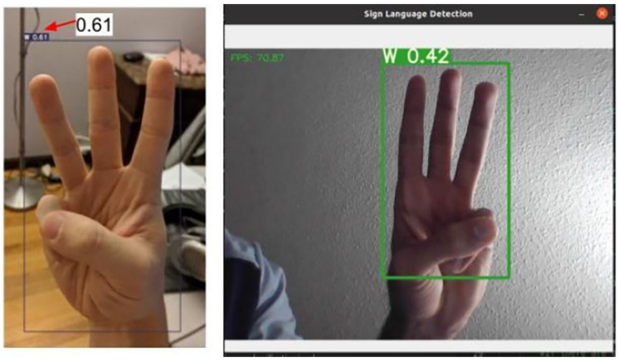


Fig. 24 The Left image shows the inference on the letter "W" from the dataset, and the right image shows inference on real camera feed showing a good detection of the "W" letter

Our proposed system addresses these concerns by combining symbolic AI techniques with deep learning. The knowledge base provides context and allows for explicit reasoning about actions, improving interpretability and safety. This also helps reduce hallucinations by grounding the robot's outputs in the real world. Furthermore, our approach facilitates user interaction for robot skill acquisition, something not directly addressed by these approaches as they lack flexibility and require large amounts of data to accommodate changes.

5.1 *From Natural Language to Knowledge Base and Vice Versa*

Language models can act as an open knowledge base using their implicit knowledge gained from pre-training on trillions of words [29]. As it was shown by Floridi and Chiriatti [30], GPT-3 can be unreliable when trying to answer logical questions, thus users cannot depend on it when it comes to safety-critical applications like physical actions performed by robots. However, this issue can be overcome when relevant context and examples are used in its prompt. Thus UJI-Butler combines KnowRob [9] with GPT-3 [10] to offer reliable, and in-context results.

A similar approach was recently carried out by Li et al. [31] which used general examples in GPT-3 prompt for the model to learn from, in contrast to UJI-Butler framework, which performs an extra similarity search on the ontology knowledge to choose the most relevant activities to that requested by the user. For example, if the request is a drink, UJI-Butler framework adds other activities related to preparing drinks, and thus the context is richer than adding only generic activities.

In UJI-Butler, knowledge of new activities is added to the knowledge base only after correct execution and human verification which is not present in any of the works found in the literature. Moreover, the use of LLMs in UJI-Butler is not limited to the creation of knowledge, but they are combined with the knowledge base to offer a new combined system that benefits from the advantages of both, similarly to the concept described in [32], while adding logical reasoning and human in the loop to ensure reliability and safety.

5.2 *Human included in the learning Loop*

Adding the human in the loop increases the reliability of the learning process and adds trust and transparency to the usage of statistical models in real life. The survey by Wu et al. [33] shows human-in-the-loop approaches for different domains in machine learning –including natural language processing and computer vision– by training prediction models like a neural network. Similarly, facts are added to the knowledge base at UJI-Butler, which can be used instantly after being learned. As it was successfully shown before by Liu et al. [34], UJI-Butler stores data for off-line machine learning and fine-tuning learning models, which allows further testing without the need for hardware.

UJI-Butler guarantees the relevance of the new information added to its knowledge base by asking a human to confirm or correct the instructions generated by GPT-3 to perform the task, given the partial state of the environment retrieved from the robot sensors and perception models. This new information enlarges the robot knowledge base, enables GPT-3 to produce better prompts, and add the possibility to do offline fine-tuning on newly collected data.

5.3 *HRI using Hand gestures*

To increase the inclusivity and accessibility of our UJI-Butler, the use of hand gestures in human-robot interaction was included. Hand gestures have been used in robotics in different applications, such as to allow non-expert users to interact with the robots[35], to teach sign language to hearing-impaired children [36], to instruct and control robots [37], in underwater scenarios when a diver needs to communicate with an underwater robot [38, 39], etc. Hand gestures provide a flexible and reliable method of communicating especially if conventional speaking methods are impossible.

Regarding the state-of-the-art in American Sign Language (ASL) which is used in this paper: [40–42] have a better overall accuracy but depth images are utilized, which we don't use in our case. Also, the main reason for not using depth cameras is that the framework is aimed at using webcams in the future for better accessibility for most users. Koller et al. used the iterative EM approach and achieved 62.8% overall recognition accuracy for 3000 images dataset, while we are using only 720 images [43].

6 Conclusion

This paper presents the UJI-Butler framework, a novel multi-robot system that, unlike previous systems, integrates large language models (LLM) with a knowledge base akin to RAG-based systems, while imposing logical reasoning on LLM-generated results. This framework allows continual online learning supported by intuitive human-robot multi-modal interaction through verbal communication, sign language, hand gestures, and physical interactivity. This interaction yields new knowledge that is stored, enhancing in this way the predictions of the LLM. The system can successfully perform user-taught physical skills and generalize them to varying object sizes and locations. Importantly, this combination of symbolic and non-symbolic AI accelerates the learning process with experience, improving the transparency and explicability of robot abilities and mistakes due to the explicit representation of knowledge. For instance, UJI-Butler can ask the human user about missing information, translating natural language into logical representations, which can be used to carry out even new tasks that are added to its knowledge base. Seamless transferability is also enabled by the explicit nature of the acquired knowledge.

Through our investigation, we have identified several limitations inherent in the current implementation of UJI-Butler, which we outline below:

1. The matching method employed for activity retrieval relies on string matching, without proper contextual understanding. Although BERT embeddings offer contextualization, they suffer from nondeterminism and higher false positives.
2. Current activity search queries in the knowledge base focus on different aspects of the activity separately, leading to not fully complete activity descriptions and potentially reduced accuracy in matching.
3. The utilization of Lead-Through programming for task learning, while effective for basic tasks, struggles with the complexity of tasks requiring detailed object perception.
4. Language interpretation accuracy is contingent upon the efficacy of language models, a field rapidly evolving with the introduction of newer, more accurate models.
5. The perception module, reliant on CLIP for classification and point-cloud processing for object segmentation, faces challenges in accurately detecting small, distant, or occluded objects, necessitating user notification to mitigate ambiguous situations.

Despite these limitations, UJI-Butler represents a significant step forward in interactive and learning robotic systems, offering a flexible framework for further development and improvement. Future work will focus on addressing these limitations through advancements in activity matching, task learning, language interpretation, and integration with perception capabilities, ultimately enhancing UJI-Butler's effectiveness and applicability in real-world scenarios. Furthermore, possible future work would be to:

(i) develop a stronger merge of KnowRob and its reasoning with the outputs of LLMs like GPT-3 which will enhance the robustness and trust in the outputs (i.e. improve and increase usage of fact-checking capabilities); (ii) improve the activity retrieval or search mechanism by finding a more context-aware approach while keeping the results deterministic and safe. (iii) test the P-tuning method for fine-tuning offline learning after data collection to improve GPT-3 model [34]; (iv) test conversational language models like Chat-GPT; (v) Extend the objects' description in KnowRob (adding colors and parts, for example) to enhance the ability of GPT-3 to predict new activity steps from just the components and their descriptions; (vi) Add the ability to get data from the internet (not just the knowledge base) would be very useful in zero-shot predictions, and in cases when the UJI-Butler framework cannot ask a human; (vii) include semantic mapping and spatial reasoning, that is, the robot will be given directions such as "go to the delivery location near to the elevator, where you will find a fire extinguisher and a trash can" and the delivery robot will follow the instructions or will ask for more features in case of multiple occurrences.

Appendix A GPT-3

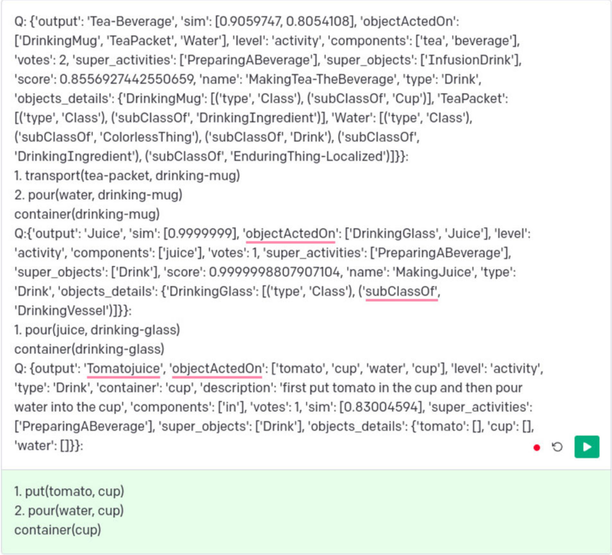
Results of an example prompt are shown in Fig. 25 which also shows the prediction for a new input given the prompt with similar inputs.

Videos

Turtlebot2 Autonomous docking
Exploring navigation capabilities with Nav2 on ROS2 and Turtlebot3 at the UJI Multi-Robots Lab
Teaching The Robot Tomato Juice Making
After Teaching the Robot Tomato Juice
Full Demo of UJI-Butler
UJI-Butler Playlist

Github Repos

Github Code Link



```
Q: {'output': 'Tea-Beverage', 'sim': [0.9059747, 0.8054108], 'objectActedOn':
['DrinkingMug', 'TeaPacket', 'Water'], 'level': 'activity', 'components': ['tea', 'beverage'],
'votes': 2, 'super_activities': ['PreparingABeverage'], 'super_objects': ['InfusionDrink'],
'score': 0.8556927442550659, 'name': 'MakingTea-TheBeverage', 'type': 'Drink',
'objects_details': {'DrinkingMug': [('type', 'Class'), ('subClassOf', 'Cup')], 'TeaPacket':
[('type', 'Class'), ('subClassOf', 'DrinkingIngredient')], 'Water': [('type', 'Class'),
('subClassOf', 'ColorlessThing'), ('subClassOf', 'Drink'), ('subClassOf',
'DrinkingIngredient'), ('subClassOf', 'EnduringThing-Localized')]}}:
1. transport(tea-packet, drinking-mug)
2. pour(water, drinking-mug)
container(drinking-mug)
Q:{'output': 'Juice', 'sim': [0.9999999], 'objectActedOn': ['DrinkingGlass', 'Juice'], 'level':
'activity', 'components': ['juice'], 'votes': 1, 'super_activities': ['PreparingABeverage'],
'super_objects': ['Drink'], 'score': 0.9999998807907104, 'name': 'MakingJuice', 'type':
'Drink', 'objects_details': {'DrinkingGlass': [('type', 'Class'), ('subClassOf',
'DrinkingVessel')]}}:
1. pour(juice, drinking-glass)
container(drinking-glass)
Q: {output': 'Tomatojuice', 'objectActedOn': ['tomato', 'cup', 'water', 'cup'], 'level': 'activity',
'type': 'Drink', 'container': 'cup', 'description': 'first put tomato in the cup and then pour
water into the cup', 'components': ['in'], 'votes': 1, 'sim': [0.83004594], 'super_activities':
['PreparingABeverage'], 'super_objects': ['Drink'], 'objects_details': {'tomato': [], 'cup': [],
'water': []}}:
```

```
1. put(tomato, cup)
2. pour(water, cup)
container(cup)
```

Fig. 25 GPT-3 ontology data to commands

Acknowledgements Abdelrhman Bassiouny and Ahmed H. Elsayed acknowledge the funding and support of the Erasmus Mundus Joint Masters Programme (EMJMD) of the European Union in the Masters of Marine and Maritime Intelligent Robotics (MIR). Zoe Falomir acknowledges the funding by the Wallenberg AI, Autonomous Systems and Software Program (WASP) awarded by the Knut and Alice Wallenberg Foundation, Sweden. This paper describes research conducted at UJI Robotic Intelligence Laboratory. Support for this laboratory is provided in part by Universidad Jaume I (UJI-B2021-42), and by Ministerio de Ciencia e Innovacion (PLEC2023-010360).

Funding Open Access funding enabled and organized by Projekt DEAL.

Data availability The authors declare that the data supporting the findings of this study are available inside this python file which is part of the code repository and is publicly available.

References

1. Gates B (2007) A robot in every home. Sci Am 296(1):58–65
2. Martinez-Martin E, Pobil AP (2018). In: Costa A, Julian V, Novais P (eds) Personal robot assistants for elderly care: an overview. Springer, Cham, pp 77–91

3. Taesi C, Aggogeri F, Pellegrini N (2023) Cobot applications-recent advances and challenges. Robotics 12(3):79
4. Thrun S, Mitchell TM (1995) Lifelong robot learning. Robot Auton Syst 15(1–2):25–46
5. Schlimmer JC, Fisher D (1986) A case study of incremental concept induction. In: Proceedings of the Fifth AAAI National Conference on Artificial Intelligence, pp 496–501
6. Sutton RS, Whitehead SD (1993) Online learning with random representations. In: Proceedings of the Tenth International Conference on International Conference on Machine Learning. ICML'93, pp 314–321. Morgan Kaufmann Publishers Inc., San Francisco, CA, USA
7. Ring MB (1997) Child: a first step towards continual learning. Mach Learn 28(1):77–104
8. McCloskey M, Cohen NJ (1989) Catastrophic interference in connectionist networks: the sequential learning problem. In: Psychology of Learning and Motivation, vol 24, pp 109–165. Academic Press
9. Tenorth M, Beetz M (2009) Knowrob-knowledge processing for autonomous personal robots. In: 2009 IEEE/RSJ International Conference on Intelligent Robots and Systems, pp 4261–4266
10. Brown T, Mann B, Ryder N, Subbiah M, Kaplan JD, Dhariwal P, Neelakantan A, Shyam P, Sastry G, Askell A et al (2020) Language models are few-shot learners. Adv Neural Inf Process Syst 33:1877–1901
11. Bechhofer S, Harmelen F, Hendler J, Horrocks I, McGuinness D, Patel-Schneijder P, Stein LA (2004) OWL web ontology language reference. Recommendation, World Wide Web Consortium (W3C)
12. Wielemaker J, Schrijvers T, Triska M, Lager T (2012) Swi-prolog. Theory Pract Logic Program 12(1–2):67–96
13. Thosar M, Zug S, Skaria A, Jain A (2018) A review of knowledge bases for service robots in household environments. In: 6th International Workshop on Artificial Intelligence and Cognition
14. Lewis P, Perez E, Piktus A, Petroni F, Karpukhin V, Goyal N, Küttler H, Lewis M, Yih W-T, Rocktäschel T et al (2020) Retrieval-augmented generation for knowledge-intensive NLP tasks. Adv Neural Inf Process Syst 33:9459–9474
15. Levenshtein VI et al (1966) Binary codes capable of correcting deletions, insertions, and reversals. Soviet Phys Doklady 10:707–710
16. Wang C-Y, Bochkovskiy A, Liao H-YM (2023) Yolov7: trainable bag-of-freebies sets new state-of-the-art for real-time object detectors. In: Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition, pp 7464–7475
17. Radford A, Kim JW, Hallacy C, Ramesh A, Goh G, Agarwal S, Sastry G, Askell A, Mishkin P, Clark J et al (2021) Learning transferable visual models from natural language supervision. In: International Conference on Machine Learning, pp 8748–8763
18. Fischler MA, Bolles RC (1981) Random sample consensus: a paradigm for model fitting with applications to image analysis and automated cartography. Commun ACM 24(6):381–395
19. McGlinn I (2021) Sign language alphabets from around the world - ASL - AI-media. Ai-Media creating accessibility, one word at a time
20. Lee D (2022) American sign language letters object detection dataset. https://public.roboflow.com/object-detection/american-sign-language-letters
21. SmartHandsCA (2017) Easiest way to learn your ASL ABCS | slowest alphabet lesson. YouTube
22. Grisetti G, Stachniss C, Burgard W (2007) Improved techniques for grid mapping with Rao-Blackwellized particle filters. IEEE Trans Robot 23(1):34–46
23. Murphy K, Russell S (2001). In: Doucet A, Freitas N, Gordon N (eds) Rao-Blackwellised particle filtering for dynamic Bayesian networks. Springer, New York, NY, pp 499–515
24. Chen C, Wu X, Han L, Ou Y (2011) Butler robot. In: 2011 IEEE International Conference on Information and Automation, pp 732–737
25. Moore RK (2013). In: Trappl R (ed) Spoken language processing: Where do we go from here? Springer, Berlin, Heidelberg, pp 119–133
26. Lee M, Heo Y, Park J, Yang H-D, Jang H-D, Benz P, Park H, Kweon IS, Oh J-H (2019) Fast perception, planning, and execution for a robotic butler: Wheeled humanoid m-hubo. In: 2019 IEEE/RSJ International Conference on Intelligent Robots and Systems (IROS), pp 5444–5451
27. Gunawan AAS, Clemons B, Halim IF, Anderson K, Adianti MP (2023) Development of e-butler: introduction of robot system in hospitality with mobile application. Procedia Comput Sci 216:67–76
28. Brohan A, Chebotar Y, Finn C, Hausman K, Herzog A, Ho D, Ibarz J, Irpan A, Jang E, Julian R et al (2023) Do as I can, not as I say: grounding language in robotic affordances. In: Conference on Robot Learning, pp 287–318
29. Salaberria A, Azkune G, Lacalle OL, Soroa A, Agirre E (2023) Image captioning for effective use of language models in knowledge-based visual question answering. Expert Syst Appl 212:118669
30. Floridi L, Chiriatti M (2020) GPT-3: its nature, scope, limits, and consequences. Minds Mach 30:681–694
31. Li F, UK A, Hogg DC, Cohn AG (2022) Ontology knowledge-enhanced in-context learning for action-effect prediction. In: Advances in Cognitive Systems
32. Jovanović M, Campbell M (2023) Connecting AI: merging large language models and knowledge graph. Computer 56(11):103–108
33. Wu X, Xiao L, Sun Y, Zhang J, Ma T, He L (2022) A survey of human-in-the-loop for machine learning. Futur Gener Comput Syst 135:364–381
34. Liu X, Zheng Y, Du Z, Ding M, Qian Y, Yang Z, Tang J (2023) GPT understands, too. AI Open 5:208
35. Nuzzi C, Pasinetti S, Pagani R, Ghidini S, Beschi M, Coffetti G, Sansoni G (2021) Meguru: a gesture-based robot program builder for meta-collaborative workstations. Robot Comput Integr Manuf 68:102085
36. Uluer P, Akalın N, Köse H (2015) A new robotic platform for sign language tutoring: Humanoid robots as assistive game companions for teaching sign language. Int J Soc Robot 7:571–585
37. Mazhar O, Ramdani S, Navarro B, Passama R, Cherubini A (2018) Towards real-time physical human-robot interaction using skeleton information and hand gestures. In: 2018 IEEE/RSJ International Conference on Intelligent Robots and Systems (IROS), pp 1–6
38. Islam MJ, Ho M, Sattar J (2018) Dynamic reconfiguration of mission parameters in underwater human-robot collaboration. In: 2018 IEEE International Conference on Robotics and Automation (ICRA), pp 6212–6219
39. Jiang Y, Zhao M, Wang C, Wei F, Wang K, Qi H (2021) Diver's hand gesture recognition and segmentation for human-robot interaction on AUV. Signal Image Video Process 15(8):1899–1906
40. Ameen S, Vadera S (2017) A convolutional neural network to classify American sign language fingerspelling from depth and colour images. Expert Syst 34(3):12197
41. Li S-Z, Yu B, Wu W, Su S-Z, Ji R-R (2015) Feature learning based on SAE-PCA network for human gesture recognition in RGBD images. Neurocomputing 151:565–573
42. Tang A, Lu K, Wang Y, Huang J, Li H (2015) A real-time hand posture recognition system using deep neural networks. ACM Trans Intel Syst Technol 6(2):1–23
43. Koller O, Ney H, Bowden R (2016) Deep hand: How to train a CNN on 1 million hand images when your data is continuous and weakly labelled. In: Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition, pp 3793–3802

Publisher's Note Springer Nature remains neutral with regard to jurisdictional claims in published maps and institutional affiliations.

Abderhman Bassiouny is a researcher at the Institute for Artificial Intelligence, Mathematics and Computer Science Department, University of Bremen, Germany. He is currently working on a research project for robotic scene understanding and video action segmentation for the application in robot action learning and knowledge gain. He obtained his double MSc Erasmus Mundus Degree in Marine & Maritime Intelligent Robotics from the University of Toulon, France, and the University Jaume I, Spain. He also obtained his BSc Degree from Ain Shams University. He worked as a research Assistant at the Robotics and Control Lab at Ain Shams University (Egypt). His research interests are Knowledge Representation & Reasoning, Human-Robot Interaction, Cognitive Systems, and Artificial Intelligence.

Ahmed H. Elsayed is a researcher at the German Research Center for Artificial Intelligence (DFKI) in the Marine Perception Department. He holds a double degree Erasmus Mundus MSc in Marine & Maritime Intelligent Robotics from Toulon University (France) and Jaume I University (Spain). Before that, he worked as an Innovation Engineer and researcher in underwater soft robotics at Nile University. He earned his BSc in Electromechanical Engineering from Alexandria University in Egypt. His research interests focus on perception and explainable AI (XAI), particularly in marine applications.

Zoe Falomir is Wallenberg Faculty and associate professor at the Computing Science Department, Umeå University, Sweden. She is leading a research project on "Autonomous systems that learn how to reason spatially by interaction" funded by WASP, the Wallenberg AI, Autonomous Systems and Software Program. She obtained her double Ph.D. title at the University of Bremen, Germany, and at the University Jaume I, Spain. She was a postdoc researcher at Bremen Spatial Cognition Centre (Germany), where she was Marie Curie Fellow and Junior Fellow at Hansewissenschaftkolleg. Then she earned a Ramon-y-Cajal tenure-track professorship, awarded by the Spanish Ministry of Science. Over the past two decades, she has authored more than 40 AI-related journal articles, collaborating with more than 50 different co-authors, her Erdös number is 4. She also contributed 2 books, 5 proceedings books, 19 book chapters and 50+ papers in conferences. Her research interests are multidisciplinary and include Spatial Reasoning, Knowledge Representation, Human-Robot-Interaction, Machine Learning, Cognitive Systems, and Creative Problem Solving

Angel P. del Pobil is a Professor of Computer Science and Artificial Intelligence at Jaume I University (Spain), where he is the founding director of the UJI Robotic Intelligence Laboratory. He was a visiting Professor at Sungkyunkwan University (2009-2013). He has over 330 publications, including 4 authored and 14 edited books. His award-winning research in the last 33 years, include contributions to humanoid robots, service robotics, motion planning, mobile manipulation and grasping, robot perception, robot physical and human interaction, robot learning as well as the interplay between neurobiology and robotics. He was a Distinguished Lecturer of the IEEE Robotics and Automation Society (2019-2024) and has presented 80 invited lectures at events around the world.