2025 IEEE/SICE International Symposium on System Integration (SII)
January 21-24, 2025, Munich, Germany

Adaptive Tidying Robots: Learning from Interaction and Observation

Ryuichi Maeda[1], Adel Baselizadeh[2], Shin Watanabe[2], Ryo Kurazume[3] and Jim Torresen[2]

Abstract— **Designing service robots capable of tidying up in unfamiliar and dynamic human environments presents a significant challenge. Such robots must not only recognize and manipulate a wide range of objects but also align their actions with tidying up rules, which may vary greatly from one individual to another. To address these challenges, we propose a comprehensive software framework that integrates Large Language Model (LLM) and Vision-Language Models (VLMs) for service robots. Our framework enables robots to learn human-specific tidying up rules through interaction and observation, and to identify and handle previously unseen objects and receptacles. This adaptive framework offers a unified solution for recognizing, learning, and acting upon diverse and dynamic human environments. We evaluate our framework using both a text-based benchmark dataset to assess tidying up rule learning and a simulated environment to demonstrate practical tidying up performance. In the evaluation using the text-based benchmark dataset, our framework selects appropriate receptacles for unseen objects with high accuracy (87.4%), including unseen receptacle categories. The simulation evaluation confirms the effectiveness of our framework in realistic environments and scenarios. This research advances the field of service robotics by presenting an integrated software solution that leverages LLM and VLMs for more personalized and adaptable robot behavior in real-world tasks.**

I. INTRODUCTION

In the rapidly developing robotic landscape, service robots play an important role in improving human lives, especially within our home. Among the numerous tasks aimed at performing daily chores, the tidying up represents an important but complex challenge. This complexity is due to the multifaceted nature of the task, which requires seamless integration of advanced technologies such as object detection, navigation, and manipulation. Previous research [1]–[5] has made significant progress towards efficient tidying up robots, but the adaptation of these technologies to real-world scenarios remains a challenge.

One of the most important aspects of these challenges is the need for robots to adapt their tidying up activities to tidying up rules, which are intrinsically diverse and change over time. For example, individuals have varying preferences for organizing their belongings: some may prefer to store clothes in a chest of drawers, while others might hang them in a closet. This variability extends to the objects themselves, depending on the type, size, and appropriate storage bin called a receptacle, in which it should be placed. Furthermore, the dynamic nature of human environments, where new objects are constantly introduced, emphasizes the need for robots to operate effectively even when facing previously unseen objects.

To address these challenges, we introduce a comprehensive software framework that leverages state-of-the-art technologies. Our framework stores the tidying up data through Human-Robot Interaction (HRI) and learns tidying up rules using Large Language Model (LLM). Traditional tidying up robot applications train their object detection models with specific labels, limiting their detection capabilities to pre-trained objects. In contrast, our framework utilizes Vision-Language Models (VLMs) to enable open vocabulary object detection, allowing the robot to recognize objects of any category. This capability empowers the robot to tidy up in environments where it is unfamiliar with the objects or the tidying up rules.

Our two-stage evaluation strategy, including a text-based benchmark dataset and a simulation environment for practical tidying up tasks, demonstrates the effectiveness of our framework. The simulation results indicate that the proposed framework is effective in adapting to unfamiliar environments and making informed decisions based on user-specific tidying up rules, marking a significant advance in service robotics.

In this work, we contribute the following: (i) Development of a prototype framework that learns tidying up rules with no prior knowledge through HRI during tidying up tasks by integrating LLM and VLMs. (ii) Demonstration of enhanced robot adaptation and decision-making capabilities through evaluations using a text-based benchmark dataset and simulation environments.

II. RELATED WORK

A. Embodied AI

Embodied Artificial Intelligence (AI) has made significant progress in several areas such as navigation [6]–[9], object search [10]–[12], and question answering [13]–[16], showing the capabilities of AI agents to perform complex tasks in real environments. Tidying up as an application of embodied AI represents unique challenges that require a deep understanding of the physical world and human-specific tidying up rule. Previous studies [4], [5] have applied general rules for tidying up tasks, but the importance of tailoring behavior to individual tidying up rule is being recognized [17], [18]. This leads to an innovative method [19] to improve the adaptability and personalization of tidying up robots by using

[1]Ryuichi Maeda is affiliated with the Graduate School of Information Science and Electrical Engineering, Kyushu University, Fukuoka, Japan. maeda@irvs.ait.kyushu-u.ac.jp

[2]Adel Baselizadeh, Shin Watanabe and Jim Torresen are affiliated with the Department of Informatics, University of Oslo, Oslo, Norway. {adelb, shinwa, jimtoer}@ifi.uio.no,

[3]Ryo Kurazume is affiliated with the Faculty of Information Science and Electrical Engineering, Kyushu University, Fukuoka, Japan. kurazume@ait.kyushu-u.ac.jp

LLM to summarize a tidying up rule from a small number of tidying up examples.

Despite these advances, existing methods still rely on a dataset that is prepared before tidying up. This requirement often limits the ability of AI agents to adapt to dynamic environments with frequently introduced new objects and scenarios. Our research aims to overcome this limitation and propose a new framework that integrates advanced novel technologies such as LLM and VLMs. Our framework allows the robots to adapt to unfamiliar environments and thus push the limits of what can be done with functional AI in real tasks.

B. Large Language Models in Robotics

LLM, particularly since the introduction of Transformer [20] style architectures, has undergone significant development, playing an important role in advancing the field of AI. These models [21], [22], pre-trained on vast amounts of web-scale textual data, have been increasingly used in robotics.

The integration of LLM has opened new avenues for developing more intuitive and human-like interactions between robots and their environments. Previous studies have focused on using LLM to inject commonsense knowledge into navigation [23]–[25], code generation [26], [27], and manipulation [28]–[30]. In the tidying up task, LLM is used to understand the tidying up rule [4], [19].

Our research builds on these foundations, but with a particular focus on the management of input/output data for LLM and data obtained through HRI, building a framework that enables learning during the tidying up process.

III. METHOD

A. Architecture of the Proposed Framework

The tidying up workflow of our framework is illustrated in Fig. 1, while an overview of the system architecture is shown in Fig. 2. The framework is composed of the following key components:

- **User Interface:** This component provides a seamless interaction between the user and the system. It allows users to input commands, provide feedback, and monitor the detected objects.
- **Task Manager:** The Task Manager oversees the execution of tidying up tasks. It coordinates between various subsystems, schedules tasks, and ensures that all processes are carried out efficiently and in the correct sequence.
- **Object Search:** Utilizing advanced VLMs, this module performs open vocabulary object detection. It identifies and locates objects and receptacles based on user-provided names, even if they are previously unseen by the robot.
- **Receptacle Selection:** This component utilizes an LLM to select the most appropriate receptacle for each object. By understanding user-specific tidying up rules through interactions and learning, it can make informed decisions on where to place objects.
- **Robot Controller:** Responsible for the physical manipulation tasks, the Robot Controller manages the navigation, grasping, and placing actions. It integrates with hardware components to execute the movements necessary for picking up objects and placing them in the designated receptacles.
- **Feedback Handler:** This module processes user feedback to correct any errors in real-time. It updates the database with corrected tidying up rules and placements, ensuring continuous learning and improvement of the robot's performance.
- **Database (DB):** stores data obtained during tidying up tasks. The database stores data acquired during tidying up tasks, including object-receptacle pairs and receptacle positions. This persistent storage enables the system to learn and adapt over time, improving its tidying efficiency and accuracy.

The framework is built on the Robot Operating System (ROS) [31], which provides a flexible and robust platform for developing and integrating the various software components of the system.

B. Finding and Grasping Target Objects

Firstly, the user provides the name of the target object, such as "coke can" and "toy car". Then, our method searches for the targets with Segment Anything Model (SAM) [32] and Contrastive Language-Image Pre-training (CLIP) [33] model. We utilize existing pre-trained models. It is not necessary to construct a new dataset, train a model from scratch, or engage in any fine-tuning procedures. These models allow our system to find unseen objects and receptacles.

SAM is an AI model designed to identify and segment any object in an image, regardless of its type or category. Our framework uses images (Fig. 3) captured by an RGB-D camera mounted on the robot and SAM to obtain bounding boxes and segmented masks for all objects in the image (Fig. 4). The bounding boxes are used to crop detected object images for the next step. The segmented masks and the depth data are used to obtain the center point of the target object for grasping.

CLIP is a machine learning technique that learns visual concepts from natural language descriptions by training on a large dataset of images and their corresponding text captions, enabling it to understand and generate representations for a wide range of visual concepts described in text. In our method, CLIP is used to compare the target object name, which is given by the user, with all detected object images cropped from the original image using bounding boxes. This process predicts similarity scores between the target object name and the detected object images. The detected object images are ordered by similarity score, and our system asks the user whether they are the target or not one by one, starting with the image with the highest similarity score (Fig. 5).

Once the target object is found, the center point of the object is calculated from the segmented mask and depth data and the robot tries to pick up the target object. Additionally, while moving around in the environment, 2D-LiDAR

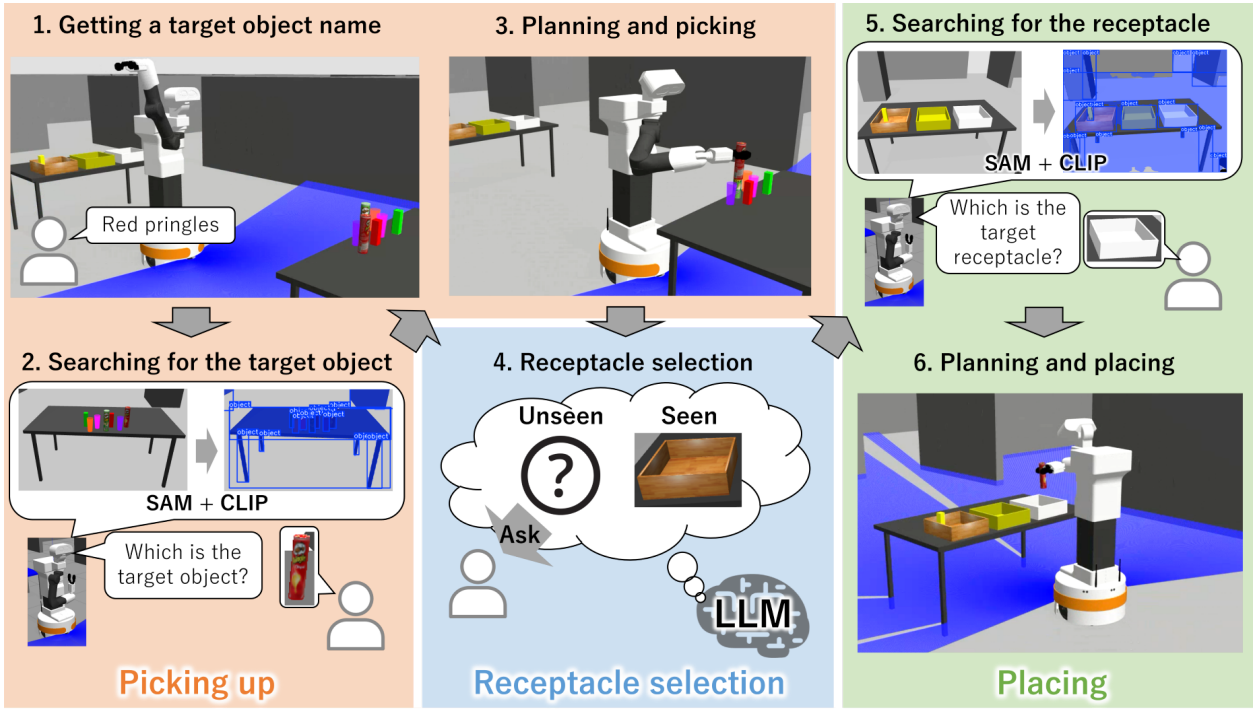

Fig. 1. Overview of tidying up workflow. There are three phases in one tidying up task. The three phases are "picking up", "receptacle prediction", and "placing". In the picking up phase, the robot searches for and picks up a target object using a given object name, image processing, and interaction with the user. In the receptacle selection phase, the robot selects an appropriate receptacle for the target object using an LLM. In the placing phase, the robot searches for the appropriate receptacle and place the target object in the receptacle.

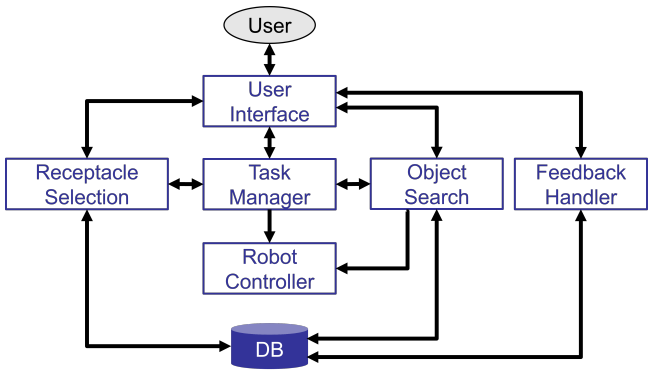

Fig. 2. Architecture of our framework. Our framework consists of multiple components that work together to enable adaptive tidying up tasks. The User Interface accepts target object names from users and passes them to the Task Manager, which coordinates the task execution. The Object Search module leverages VLMs to detect objects, while the Robot Controller handles the physical manipulation of these objects. The Receptacle Selection utilizes LLM to select the appropriate receptacle based on tidying up data stored in the database (DB). Finally, the Feedback Handler processes user input to update tidying up data, allowing continuous improvement of tidying performance.

SLAM [34] is used to generate a map of the environment. Receptacle positions obtained during this object search process are stored in a database. The next time the same receptacle is used, an efficient navigation path is generated by A-star algorithm based on the map and the receptacle's position and the robot automatically places the target object in the receptacle without further user interaction.

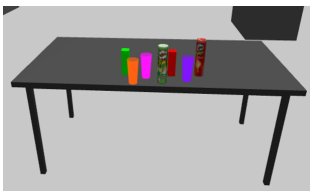

Fig. 3. This is a RGB image captured by the RGB-D camera. In this scene, there are 5 objects on the table and some walls behind it.

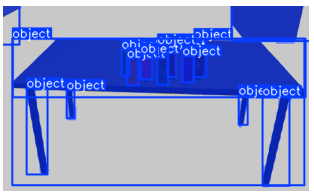

Fig. 4. This is an image after SAM is applied. Table top objects, a table, walls, and some other parts are all detected as "object."

C. Selection of Appropriate Receptacles

In this process, an extended method of the summarization method using LLM [19] is applied. Their method summarizes the tidying up rule based on a given tidying up data using LLM before tidying up task. The tidying up data consists of a few pairs of object and receptacle and this data is given by the user in advance. During the tidying up task, the summary, target objects, and receptacle candidates which is in tidying up data are given to LLM and LLM selects appropriate receptacle for each object. Their method can predict the appropriate receptacle with high accuracy. However, their method can only select receptacles from the given tidying up data, which must be provided before doing tidying up tasks.

We extend their method to be able to handle receptacles which is not in tidying up data. In our framework, an "others" category is added to receptacle candidates as an outlier class.

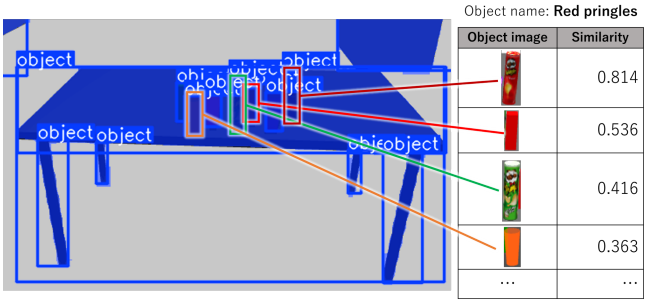

Fig. 5. On the left, the image after applying segmentation is shown, and the table on the right shows the similarity scores sorted in order of highest to lowest after comparing the target object name and detected object images. In this case, "Red pringles" is the target object name. The user is asked whether the detected object is the target object or not, starting from the top in this table.

If outlier class is selected as the appropriate receptacle by LLM, the robot asks the user which receptacle is appropriate for the target object and store new tidying up data in the database. Unlike [19], this method of integrating LLM and HRI eliminates the need for the user to provide the system with an example of the placements before doing tidying up tasks.

In our method, there are two steps to select the appropriate receptacle for the target object. The two steps are summarization step and prediction step. In the summarization step, we use two types of prompts. The first prompt is a system prompt, which gives instruction and example responses to LLM (Fig. 6). The system prompt is defined before doing the tidying up tasks. The second prompt is a user prompt (Fig. 7). After the user prompt is given to LLM, they return the summary. The user prompt is used in each process of receptacle selection. In the prediction step, we use two types of prompts much like the summarization step, to predict the appropriate receptacle. The first prompt is a system prompt (Fig. 8) and the second prompt is a user prompt (Fig. 9). After the user prompt is given to LLM, they select an appropriate receptacle. Not only does our method not require the user to provide some examples before doing the tidying up tasks, but furthermore, our system can update itself to make it more personalized.

D. Management of Tidying Up Data

When the robot is first introduced into the environment, the database is empty. Our framework stores the following tidying up data in a database during tidying up tasks:

- ***Pairs of object and appropriate receptacle for the object***
  This data is stored through conversation with user during the tidying up task. Our framework uses this data to understand the rules for tidying up.
- ***Position of each receptacle***
  This data is stored through receptacle search process. The next time the same receptacle is used, the robot automatically places the target object in the receptacle using this data.

System prompt:

Instruction
Summarize object-receptacle pairs into general rules.
These pairs, representing the relationship between an object and its suitable receptacle, are provided by users.

Examples
Example1
User:
pairs = {banana: coffee table, Lego brick: storage box, toy car: storage box, apple: coffee table}

Your response:
Put toys in the storage box and fruits on the coffee table.

Example2
User:
pairs = {dress: basket, Play-Doh: drawer, pants: basket, toy block: drawer}

Your response:
Put clothes in the basket and toys in the drawer.

Fig. 6. An example system prompt to summarize placements preference of the user. There are two sections. In the first section, the instruction is stated. In the second section, there are two examples to adjust the LLM's response.

User prompt:

pairs = {history book: coffee table, cookbook: coffee table, brochure: trash can, paper towel: trash can}

LLM's response:

Put books on the coffee table. Put paper products in the trash can.

Fig. 7. Pairs of objects and receptacles are given as a user prompt. The LLM's response is expected to return the summary of the user preference or rule from the given pairs.

By utilizing this stored data and LLM, the robot understands user-specific tidying up rules and gradually performs tidying up tasks automatically.

1) User Feedback for Adaptive Tidying Up: The robot could select the wrong receptacle and the wrong tidying up data could be stored. To address such cases, our framework includes a feedback system. This system operates asynchronously with the tidying up operation. If the user finds that an object is placed in the wrong receptacle, the user can provide the correct receptacle for the object via feedback, and our framework updates the incorrect data in the database. Therefore, in our framework, the robot flexibly learns the tidying up rule through continuous interaction with the user. It can then improve its performance overtime.

IV. EXPERIMENTS

We evaluate the performance of our framework through two perspectives. Firstly, we evaluate the method proposed by TidyBot [19], particularly focusing on the introduction of outlier classes. This evaluation aims to measure the framework's ability to handle previously unseen receptacle

System prompt:

Instruction
A tidying up rule and a target object are given by the user. Return an appropriate receptacle for the target object based on the tidying up rule. If the target object is not included in the tidying up rule, return "others".

Examples
Example1
User:
tidying up rule = Put toys in the storage box and fruits on the coffee table.
target object = orange

Your response:
coffee table

Example2
User:
tidying up rule = Put toys in the storage box and fruits on the coffee table.
target object = paper towel

Your response:
others

Example3
User:
tidying up rule = Put clothes in the basket and toys in the drawer.
target object = sweater

Your response:
basket

Fig. 8. An example system prompt to select the appropriate receptacle for the given target object. There are two sections. In the first section, the instruction is stated. In the second section, there are three examples to adjust the LLM's response.

categories effectively. Secondly, a prototype of our framework is implemented in a simulator environment to evaluate its practical performance. In our method, the GPT [35] series model is used as the core of our LLM. Specifically, we use the **gpt-4o** model by an API provided by OpenAI [36]. For consistency and controlled experimentation, the temperature parameter is set to 0.

A. Evaluation of the Receptacle Selection Method

In this experiment, we assess the impact of introducing outlier classes on the accuracy of the TidyBot [19] method. To do so, we utilize two benchmarks: the original proposed in [19] and an extended version designed to test adaptability to unseen receptacles.

1) Dataset: The original benchmark consists of 96 scenarios; each scenario has a set of objects, a set of receptacles, a set of placement examples of seen objects, and a set of placements of unseen objects, all specified as text. Each scenario has between two and five seen receptacles, with two seen objects and two unseen objects assigned to each seen receptacle. The task is to select the appropriate receptacle for the unseen objects based on the seen examples.

User prompt:
tidying up rule = Put books on the coffee table. Put paper products in the trash can
target object = paper wad

LLM's response:
trash can

Fig. 9. User prompt consists of **tidying up rule**, and **target object**. The LLM's response is expected to return the appropriate receptacle for the target object from receptacles.

This benchmark is extended to evaluate the adaptability of our method to unseen receptacles. In the expansion, two objects unrelated to the scenario are added to all scenarios (excluding a fully classified scenario, "Put heavy items on the left shelf and light items on the right shelf"). The appropriate receptacle for these newly added objects is categorized "others", representing an outlier class. In this extended benchmark, the challenge is to select that the appropriate receptacles for the newly added object do not exist among the earlier seen receptacles.

2) Results: Table I shows the accuracy results on the original benchmark dataset that does not include unseen receptacles. The TidyBot method [19] achieved higher accuracy (89.1%) compared to our method (85.4%). The reason for the lower accuracy of our method is that our method includes an outlier class for receptacles, meaning it selects the appropriate receptacle from a set that includes unnecessary option, whereas TidyBot does not. Nevertheless, the margin of accuracy remains relatively modest at 3.7%. On the extended benchmark dataset, the TidyBot method with outlier class integration achieved an accuracy of 87.4% (Table II), which is close to its accuracy on the original benchmark dataset (89.1%).

When an outlier class is selected during the tidying up process, the robot verifies the appropriate receptacle with the user, ensuring successful tidying up. Assuming that the selection of the outlier class always leads to the identification of the appropriate receptacle, the accuracy of the TidyBot method with outlier class integration on the extended benchmark dataset increases to 92%. These results indicate that the receptacle selection method used in our framework maintains reliable accuracy even with the integration of the outlier class. Furthermore, the inclusion of the outlier class enhances the success rate of tidying up task in realistic environments, demonstrating the practical benefits of our approach.

B. Simulation Experiment

In this experiment, we evaluate a prototype of our framework using the TIAGo (Fig. 10) robot within the Gazebo simulator [37]. The robot is equipped with an RGB-D camera for object detection and manipulation. For the object search process, we employ the FastSAM model [38], based on the YOLOv8 architecture [39], along with the ViT-B/32 model of CLIP [33].

TABLE I

ACCURACY RESULTS ON ORIGINAL BENCHMARK DATASET (WITHOUT UNSEEN RECEPTACLES)

Methods	Accuracy (%)
TidyBot [19]	**89.1**
TidyBot [19] + Outlier Class	85.4

TABLE II

ACCURACY RESULTS ON EXTENDED BENCHMARK DATASET (WITH UNSEEN RECEPTACLES)

Methods	Accuracy (%)
TidyBot [19] + Outlier Class	**87.4**

1) Experimental Environment: Fig. 11 illustrates the experimental environment. The experiment consists of 10 complete tidying up scenarios using three types of containers as receptacles (Fig. 12) and 11 objects categorized into four types (Fig. 13). Each scenario involves tidying up objects from three randomly selected categories out of the four available categories.

Before starting each scenario, the database is cleared to ensure no prior knowledge affects the robot's performance. Within each scenario, the order in which the objects are picked up and placed into containers is randomized. Additionally, the assignment of objects to specific containers is randomized at the category level to maintain structured variability. If the robot selects the wrong receptacle and places the object in it, immediate feedback is provided and incorrect tidying up data is correctly updated accordingly.

This approach allows us to test the robustness and adaptability of our framework under varying conditions while maintaining consistency in the experimental parameters.

2) Evaluation Method: The evaluation of this experiment is based on the following metrics:

- **Tidying up success rate:** In 10 scenarios, the robot performs 83 tidying up tasks. This metric indicates whether the target object is placed in the appropriate receptacle. In addition to calculating the success rate, we also calculate the percentage of each failure factor.
- **Average number of user interactions:** Our framework has some processes that require user interactions. This evaluation focuses on interactions in the process required during object search and receptacle selection, which are particularly important.

3) Results: Table III shows the tidying up success rate along with the rates for each failure factor. The failure factors are defined as follows:

- **Failed to grasp:** The robot physically failed to grasp the object due to issues such as slippage.
- **Failed to place:** The robot selected the appropriate receptacle but physically failed to place the object in it due to issues such as misjudged positioning.
- **Placed in wrong receptacle:** The robot selected the wrong receptacle for the target object and placed it there. In such cases, the feedback system was used in Section III-D.

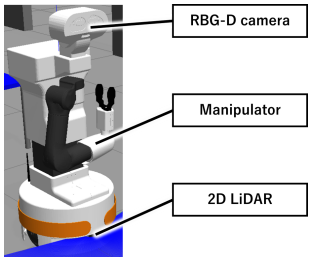

Fig. 10. This is a 3D model of a mobile service robot called TIAGo [40].

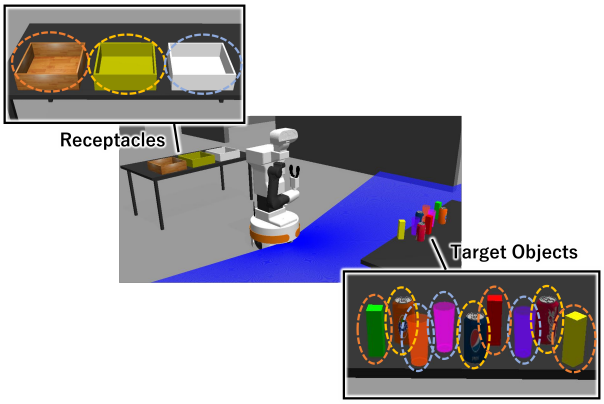

Fig. 11. This is a simulation environment and an example scenario. There are three types of containers as receptacles and nine objects. In each scenario, all containers and three categories of objects out of four categories were used. In this example scenario, the appropriate receptacles of bricks (outlined by an orange circle), cans (outlined by a yellow circle), and cups (outlined by a gray circle) are a brown container, a yellow container, and a white container respectively.

The low rate of *Placed in wrong receptacle* shows that receptacle selection method works well. However, it is found that improving the accuracy of grasping and placing is crucial for enhancing the overall success rate.

Table IV shows the average number of user interactions required during object search and receptacle selection. During object search, multiple interactions were often needed when the similarity between the target object image and the user-given object name was not the highest. This occurred in cases mostly where: (i) Similar-looking objects were present, or (ii) one part of the target object segmented by SAM had a higher similarity than the image of the entire object. During receptacle selection, in the second half of the scenario, the robot began to understand the tidying up rule and selected receptacles without user interaction, resulting in an average number 0.6. When LLM selects the outlier class during receptacle selection, one user interaction is required to identify the appropriate receptacle. However, if LLM selects a specific receptacle, no user interaction is needed. Therefore, a lower average number indicates that the robot requires less user assistance when placing objects in receptacles.

Table V details the rates at which user interactions were required during receptacle selection, based on the number of objects from the same category that had already been tidied up. A lower rate indicates that the robot is able to tidy up the object automatically without user interaction Therefore,

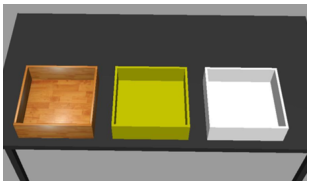

Fig. 12. Three types of containers used in the simulation experiment.

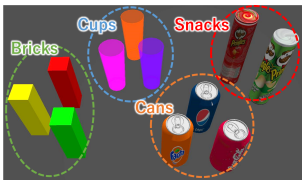

Fig. 13. Eleven objects used in the simulation experiment. They are categorized into four types: bricks, cups, cans, and snacks.

TABLE III

TIDYING UP SUCCESS RATE AND FAILURE FACTOR RATES

	Rate (%)
Success	**72.3**
Failed to grasp	18.1
Failed to place	6.0
Placed in wrong receptacle	3.6

this result shows that if no objects of the same category have been tidied up, interaction is required to learn the appropriate receptacle for the target object. On the other hand, if two objects of the same category have been tidied up, interaction is not required to select the appropriate receptacle. This demonstrates that the robot gradually understands the tidying up rule as it tidies up.

V. DISCUSSION

In the experiments, the integration of LLM and VLMs demonstrated significant improvements in the robot's ability to recognize and adapt to unseen objects and receptacles, achieving a high accuracy rate of 87.4% for unseen objects in a text-based benchmark dataset. This advancement suggests that leveraging language models can effectively enhance the robot's contextual understanding and decision-making capabilities in dynamic environments.

Furthermore, the simulation results indicate that the proposed framework can successfully operate in realistic scenarios, showing its potential for practical deployment in domestic and service settings. The user feedback mechanism proved crucial in refining the robot's tidying up rules, ensuring continuous learning and adaptation over time.

However, despite these promising outcomes, there are several areas required further investigation. For instance, optimizing the object search process to reduce user interactions and incorporating more sophisticated navigation strategies could significantly enhance the overall efficiency of the system. Additionally, one of the most critical areas for improvement is the grasping process, as failures in grasping significantly impact the overall success rate. Enhancing the grasping mechanism by integrating advanced grasp planning algorithms and utilizing more dexterous manipulators could reduce the frequency of grasping failures.

A. Limitations

Despite the promising results, our framework has several limitations that need to be addressed. One significant limitation lies in the object search process. Although the integration of SAM and CLIP models enhances the robot's capability to detect and identify a wide range of objects, the efficiency of this process decreases when the target object is not in close proximity to the robot's initial position. This scenario often requires multiple user interactions, leading to extended search times and reduced overall efficiency.

Moreover, the current brute-force search approach is not optimal for real-world applications where rapid and precise identification of objects is crucial. To overcome this, future works of our framework should incorporate advanced search algorithms that prioritize specific or highly similar objects based on user preferences or contextual clues.

By addressing these limitations, we can significantly improve the practical usability and performance of our tidying-up robots in dynamic and cluttered environments.

TABLE IV

AVERAGE NUMBER OF USER INTERACTIONS IN EACH PROCESS

	Average
Object search	1.4
Receptacle selection	0.6

TABLE V

RATE OF TIMES WHEN INTERACTION WAS REQUIRED IN RECEPTACLE SELECTION

Number of objects tidied up in the same category	Rate(%)
0	100.0
1	46.2
2	0.0

VI. CONCLUSION

In this study, we introduced a framework enabling robots to adapt to unfamiliar environments by learning human-specific tidying up rules using Large Language Model (LLM) and Vision-Language Models (VLMs). Leveraging SAM and CLIP models, our framework effectively handles unseen objects and receptacles, enhancing the robot's tidying capabilities.

Our experiments with a text-based benchmark dataset and a simulator demonstrated the framework's high accuracy (87.4%) in predicting appropriate receptacles and adapting to new objects. The user feedback mechanism was essential for continuous learning and rule refinement.

Future work will focus on optimizing the object search process to minimize user interactions and improving the grasping mechanism to reduce failures. Additionally, we plan to validate the framework with real-world testing and integrate more user-centric design features to enhance overall efficiency and user experience. These enhancements aim to make our framework a practical solution for real-world service robotics applications.

ACKNOWLEDGMENT

This work is partially supported by The Research Council of Norway (RCN) as a part of the projects: Collaboration on Intelligent Machines (COINMAC) project, under grant agreement no. 309869, Vulnerability in the Robot Society

(VIROS) under Grant Agreement No. 288285, Predictive and Intuitive Robot Companion (PIRC) under grant agreement no. 312333 and through its Centres of Excellence scheme, RITMO with project no. 262762.

REFERENCES

[1] R. Rasch, D. Sprute, A. Pörtner, S. Battermann, and M. König, "Tidy up my room: Multi-agent cooperation for service tasks in smart environments," *Journal of Ambient Intelligence and Smart Environments*, vol. 11, no. 3, pp. 261–275, 2019.

[2] Z. Yan, N. Crombez, J. Buisson, Y. Ruichck, T. Krajnik, and L. Sun, "A quantifiable stratification strategy for tidy-up in service robotics," in *2021 IEEE International Conference on Advanced Robotics and Its Social Impacts (ARSO)*. IEEE, 2021, pp. 182–187.

[3] A. Taniguchi, S. Isobe, L. El Hafi, Y. Hagiwara, and T. Taniguchi, "Autonomous planning based on spatial concepts to tidy up home environments with service robots," *Advanced Robotics*, vol. 35, no. 8, pp. 471–489, 2021.

[4] Y. Kant, A. Ramachandran, S. Yenamandra, I. Gilitschenski, D. Batra, A. Szot, and H. Agrawal, "Housekeep: Tidying virtual households using commonsense reasoning," in *European Conference on Computer Vision*. Springer, 2022, pp. 355–373.

[5] G. Sarch, Z. Fang, A. W. Harley, P. Schydlo, M. J. Tarr, S. Gupta, and K. Fragkiadaki, "Tidee: Tidying up novel rooms using visuo-semantic commonsense priors," in *European Conference on Computer Vision*. Springer, 2022, pp. 480–496.

[6] E. Kolve, R. Mottaghi, W. Han, E. VanderBilt, L. Weihs, A. Herrasti, M. Deitke, K. Ehsani, D. Gordon, Y. Zhu *et al.*, "Ai2-thor: An interactive 3d environment for visual ai," *arXiv preprint arXiv:1712.05474*, 2017.

[7] P. Anderson, A. Chang, D. S. Chaplot, A. Dosovitskiy, S. Gupta, V. Koltun, J. Kosecka, J. Malik, R. Mottaghi, M. Savva *et al.*, "On evaluation of embodied navigation agents," *arXiv preprint arXiv:1807.06757*, 2018.

[8] D. Batra, A. X. Chang, S. Chernova, A. J. Davison, J. Deng, V. Koltun, S. Levine, J. Malik, I. Mordatch, R. Mottaghi *et al.*, "Rearrangement: A challenge for embodied ai," *arXiv preprint arXiv:2011.01975*, 2020.

[9] C. Gan, S. Zhou, J. Schwartz, S. Alter, A. Bhandwaldar, D. Gutfreund, D. L. Yamins, J. J. DiCarlo, J. McDermott, A. Torralba *et al.*, "The threedworld transport challenge: A visually guided task-and-motion planning benchmark towards physically realistic embodied ai," in *2022 International Conference on Robotics and Automation (ICRA)*. IEEE, 2022, pp. 8847–8854.

[10] R. Ramrakhya, E. Undersander, D. Batra, and A. Das, "Habitat-web: Learning embodied object-search strategies from human demonstrations at scale," in *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, 2022, pp. 5173–5183.

[11] J. Park, T. Yoon, J. Hong, Y. Yu, M. Pan, and S. Choi, "Zero-shot active visual search (zavis): Intelligent object search for robotic assistants," in *2023 IEEE International Conference on Robotics and Automation (ICRA)*. IEEE, 2023, pp. 2004–2010.

[12] W. Liu, D. Bansal, A. Daruna, and S. Chernova, "Learning instance-level n-ary semantic knowledge at scale for robots operating in everyday environments," *Autonomous Robots*, pp. 1–19, 2023.

[13] A. Das, S. Datta, G. Gkioxari, S. Lee, D. Parikh, and D. Batra, "Embodied question answering," in *Proceedings of the IEEE conference on computer vision and pattern recognition*, 2018, pp. 1–10.

[14] D. Gordon, A. Kembhavi, M. Rastegari, J. Redmon, D. Fox, and A. Farhadi, "Iqa: Visual question answering in interactive environments," in *Proceedings of the IEEE conference on computer vision and pattern recognition*, 2018, pp. 4089–4098.

[15] E. Wijmans, S. Datta, O. Maksymets, A. Das, G. Gkioxari, S. Lee, I. Essa, D. Parikh, and D. Batra, "Embodied question answering in photorealistic environments with point cloud perception," in *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, 2019, pp. 6659–6668.

[16] L. Yu, X. Chen, G. Gkioxari, M. Bansal, T. L. Berg, and D. Batra, "Multi-target embodied question answering," in *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, 2019, pp. 6309–6318.

[17] N. Abdo, C. Stachniss, L. Spinello, and W. Burgard, "Robot, organize my shelves! tidying up objects by predicting user preferences," in *2015 IEEE international conference on robotics and automation (ICRA)*. IEEE, 2015, pp. 1557–1564.

[18] I. Kapelyukh and E. Johns, "My house, my rules: Learning tidying preferences with graph neural networks," in *Conference on Robot Learning*. PMLR, 2022, pp. 740–749.

[19] J. Wu, R. Antonova, A. Kan, M. Lepert, A. Zeng, S. Song, J. Bohg, S. Rusinkiewicz, and T. Funkhouser, "Tidybot: Personalized robot assistance with large language models," *arXiv preprint arXiv:2305.05658*, 2023.

[20] A. Vaswani, N. Shazeer, N. Parmar, J. Uszkoreit, L. Jones, A. N. Gomez, Ł. Kaiser, and I. Polosukhin, "Attention is all you need," *Advances in neural information processing systems*, vol. 30, 2017.

[21] A. Radford, K. Narasimhan, T. Salimans, I. Sutskever *et al.*, "Improving language understanding by generative pre-training," 2018.

[22] J. Devlin, M.-W. Chang, K. Lee, and K. Toutanova, "Bert: Pre-training of deep bidirectional transformers for language understanding," *arXiv preprint arXiv:1810.04805*, 2018.

[23] B. Yu, H. Kasaei, and M. Cao, "L3mvn: Leveraging large language models for visual target navigation," *arXiv preprint arXiv:2304.05501*, 2023.

[24] K. Zhou, K. Zheng, C. Pryor, Y. Shen, H. Jin, L. Getoor, and X. E. Wang, "Esc: Exploration with soft commonsense constraints for zero-shot object navigation," *arXiv preprint arXiv:2301.13166*, 2023.

[25] A. Majumdar, F. Xia, D. Batra, L. Guibas *et al.*, "Findthis: Language-driven object disambiguation in indoor environments," in *7th Annual Conference on Robot Learning*, 2023.

[26] I. Singh, V. Blukis, A. Mousavian, A. Goyal, D. Xu, J. Tremblay, D. Fox, J. Thomason, and A. Garg, "Progprompt: Generating situated robot task plans using large language models," in *2023 IEEE International Conference on Robotics and Automation (ICRA)*. IEEE, 2023, pp. 11 523–11 530.

[27] J. Liang, W. Huang, F. Xia, P. Xu, K. Hausman, B. Ichter, P. Florence, and A. Zeng, "Code as policies: Language model programs for embodied control," in *2023 IEEE International Conference on Robotics and Automation (ICRA)*. IEEE, 2023, pp. 9493–9500.

[28] M. Ahn, A. Brohan, N. Brown, Y. Chebotar, O. Cortes, B. David, C. Finn, C. Fu, K. Gopalakrishnan, K. Hausman *et al.*, "Do as i can, not as i say: Grounding language in robotic affordances," *arXiv preprint arXiv:2204.01691*, 2022.

[29] A. Brohan, N. Brown, J. Carbajal, Y. Chebotar, X. Chen, K. Choromanski, T. Ding, D. Driess, A. Dubey, C. Finn *et al.*, "Rt-2: Vision-language-action models transfer web knowledge to robotic control," *arXiv preprint arXiv:2307.15818*, 2023.

[30] K. Lin, C. Agia, T. Migimatsu, M. Pavone, and J. Bohg, "Text2motion: From natural language instructions to feasible plans," *arXiv preprint arXiv:2303.12153*, 2023.

[31] M. Quigley, K. Conley, B. Gerkey, J. Faust, T. Foote, J. Leibs, R. Wheeler, A. Y. Ng *et al.*, "Ros: an open-source robot operating system," in *ICRA workshop on open source software*, vol. 3, no. 3.2. Kobe, Japan, 2009, p. 5.

[32] A. Kirillov, E. Mintun, N. Ravi, H. Mao, C. Rolland, L. Gustafson, T. Xiao, S. Whitehead, A. C. Berg, W.-Y. Lo *et al.*, "Segment anything," *arXiv preprint arXiv:2304.02643*, 2023.

[33] A. Radford, J. W. Kim, C. Hallacy, A. Ramesh, G. Goh, S. Agarwal, G. Sastry, A. Askell, P. Mishkin, J. Clark *et al.*, "Learning transferable visual models from natural language supervision," in *International conference on machine learning*. PMLR, 2021, pp. 8748–8763.

[34] G. Grisetti, C. Stachniss, and W. Burgard, "Improved techniques for grid mapping with rao-blackwellized particle filters," *IEEE transactions on Robotics*, vol. 23, no. 1, pp. 34–46, 2007.

[35] T. Brown, B. Mann, N. Ryder, M. Subbiah, J. D. Kaplan, P. Dhariwal, A. Neelakantan, P. Shyam, G. Sastry, A. Askell *et al.*, "Language models are few-shot learners," *Advances in neural information processing systems*, vol. 33, pp. 1877–1901, 2020.

[36] OpenAI, "OpenAI Platform," https://platform.openai.com/overview, accessed: 2024-08-30.

[37] N. Koenig and A. Howard, "Design and use paradigms for gazebo, an open-source multi-robot simulator," in *IEEE/RSJ International Conference on Intelligent Robots and Systems*, Sendai, Japan, Sep 2004, pp. 2149–2154.

[38] X. Zhao, W. Ding, Y. An, Y. Du, T. Yu, M. Li, M. Tang, and J. Wang, "Fast segment anything," *arXiv preprint arXiv:2306.12156*, 2023.

[39] G. Jocher, A. Chaurasia, and J. Qiu, "Yolo by ultralytics," https://github.com/ultralytics/, accessed: 2024-08-30.

[40] PAL Robotics, "Tiago - mobile manipulator robot," https://pal-robotics.com/robot/tiago/, accessed: 2024-08-30.