Robotica (2024), 1–22
doi:10.1017/S0263574724001322

CAMBRIDGE UNIVERSITY PRESS

RESEARCH ARTICLE

A multi-modal learning method for pick-and-place task based on human demonstration

Diqing Yu[1], Xinggang Fan[1,2], Yaonan Li[2], Heping Chen[2], Han Li[1] and Yuao Jin[2]

[1]Zhejiang University of Technology, Hangzhou, China
[2]Shenzhen Academy of Robotics, Shenzhen, China
Corresponding author: Yaonan Li; Email: ynli@szarobots.com

Received: 20 March 2024; **Revised:** 2 July 2024; **Accepted:** 2 August 2024

Keywords: Multi-modal robot agent; one-shot imitation; task learning; action prediction; real-world demonstration dataset

Abstract

Robot pick-and-place for unknown objects is still a very challenging research topic. This paper proposes a multi-modal learning method for robot one-shot imitation of pick-and-place tasks. This method aims to enhance the generality of industrial robots while reducing the amount of data and training costs the one-shot imitation method relies on. The method first categorizes human demonstration videos into different tasks, and these tasks are classified into six types to symbolize as many types of pick-and-place tasks as possible. Second, the method generates multi-modal prompts and finally predicts the action of the robot and completes the symbolic pick-and-place task in industrial production. A carefully curated dataset is created to complement the method. The dataset consists of human demonstration videos and instance images focused on real-world scenes and industrial tasks, which fosters adaptable and efficient learning. Experimental results demonstrate favorable success rates and loss results both in simulation environments and real-world experiments, confirming its effectiveness and practicality.

1. Introduction

The use of robots has been integrated into various aspects of modern society. Robots are often used to complete high-precision and highly repetitive tasks. The most widely and frequently encountered task among them is the pick-and-place task. For some industrial assembly line tasks, simply predefined fixed trajectories, speeds, and pick-and-place poses to the robot end can be done. However, when tasks have certain constraints, such as selection tasks, sorting tasks, and assembly tasks, simply predefined trajectories are far from sufficient to complete the task. One method is to select different robots and pick-and-place strategies for different tasks, but this method is not only costly but also very cumbersome to implement. Therefore, how to enhance the generality of robots by implementing intelligent pick-and-place is always a huge challenge. Recent advancements in Transformers [1] have paved the way for large-scale language models (LLMs) such as GPT-4 [2, 3], showcasing remarkable progress in various downstream natural language processing tasks [4–7]. These advancements have facilitated the widespread adoption of GPT models in various robotic applications, including social robots [8–10] and collaborative efforts with computer vision systems [11–13], enabling robots to acquire and process information more effectively to accomplish complex tasks with greater precision.

Acquiring perception before picking is a prerequisite for robot pick-and-place tasks [14]. To achieve intelligent and precise control of modern robots, various methods for obtaining and processing information have emerged. Generative models based on multi-modal prompts [15–18] have been proposed to integrate different forms of information and produce a unified output. This integration has led to the development of robot agents based on multi-modal prompts, also known as multi-modal robot agents.

Several notable multi-modal robot agents have demonstrated promising results, such as Palm-E [19], a leading example of embodied artificial intelligence. Palm-E possesses strong capabilities in visual and language tasks and proposes further control through decision-makers [20]. This approach mimics human behavior by involving a step-by-step process of "understanding" and "execution." However, in many practical scenarios, utilizing multi-modal prompts for robot control may not be convenient, as it is impractical to expect every base operator to understand how to operate using multi-modal prompts.

To address this challenge, "one-shot imitation" has emerged as an effective approach [21–24], requiring only a demonstration of the task under the camera for the robot to replicate. Refs. [25] and [26] are end-to-end learning methods for robots based on deep learning. Although deep learning methods perform well in handling complex data and tasks, they often lack interpretability and always have a boundary where probability information must be "explained" and transformed into precise logical decisions. However, current autonomous systems have not effectively disseminated this probability information in the logical decision-making process, resulting in the loss of overall information. For example, when robots face uncertain scenarios, they usually still need to make a "yes/no" decision, and complex behavioral responses require designers to specify the robot's behavior in each situation [27]. But if artificially specify one by one what robots should do in these uncertain situations, it will be a huge workload. In such applications, robots interact with human users, so understanding the reasons and logic behind robot behavior is crucial. And existing "one-shot imitation" methods often rely on meta-learning [28], which necessitates extensive data for training and may require retraining or model adjustments for new tasks.

VIMA [29] combines the method of multi-modal prompts and one-shot imitation. It can process multi-modal prompts and predict robot action sequences akin to Palm-E. However, it is based on a simulation environment, and the expert demonstrations used for learning in VIMA are also generated by the simulation environment, so it cannot be easily extended to real-world demonstrations. In addition, its "one-shot imitation" method leans more toward accurate reproduction of the trajectory of the demonstration rather than understanding and analyzing the demonstration. Its powerful multi-modal prompt understanding and processing ability have not been fully utilized here. Therefore, multi-modal learning method (MLM) has been proposed to compensate for this deficiency.

This paper proposes a new method for robot pick-and-place task imitation learning based on human demonstration, which is different from existing robot LfD methods (like meta-learning and end-to-end learning), and is a paradigm of demonstration classification plus pre-training multi-modal robot agent. The method defines six types of pick-and-place tasks and classifies human demonstrations into these six tasks. Meanwhile, the method defines the categories of objects and detects objects within current scene. Using the Realsense camera to have depth information of the objects, the method calculates the height of objects in the scene to enable the robot to pick up objects at the correct height. Then the method generates multi-modal prompts that are proposed in this paper and embeds the information obtained from object detection into the multi-modal prompts. Pre-trained multi-modal robot agent is used in the method to predict robot action sequences. The satisfactory success rates and loss results obtained in both simulation environments and real-world experiments validate the effectiveness and practicality of MLM. This new method has stronger logicality compared to existing methods, and it has a learning approach similar to human action of "understanding what is going on" and then "learning what to do," which also makes MLM more structured and human–robot interactive. This is a new paradigm for robot LfD, which can achieve what existing methods require a huge amount of training with a very small amount of training and may have enormous potential in the future. In practical scenarios, such as in industrial assembly lines, this method can enable the use of the same robot on the same assembly line to handle more different tasks without the need to open another assembly line or design another robot to complete different tasks. This will result in significant cost savings.

This paper also proposes a new dataset that is compatible with MLM. The dataset is divided into two parts: the first part consists of over 300 real-world demonstration videos of pick-and-place tasks, which

are mainly used to train and test the VCM; the second part consists of over 300 annotated images, which are used for training and testing the object detection module. They contain objects of different shapes. Unlike some existing behavioral demonstration datasets, the establishment of this dataset is entirely based on industrial pick-and-place tasks, and this dataset is all captured in real world. Therefore, this dataset is conducive to its rapid deployment for MLM in the real world.

And this paper is organized as follows. Section Related Works (Section 2) first introduces some related works to explain some limitations of existing methods and how MLM overcomes these limitations. Next, the section Method (Section 3) describes how MLM works module by module. Then section Experiment (Section 4) introduces the experimental design and evaluation result to reveal the effectiveness of the method. Finally, the section Conclusions & Discussion (Section 5) summarizes the work and proposes some prospects.

2. Related works

Multi-modal robot agents, combining visual and language capabilities, have garnered significant attention in recent years. For instance, approaches integrating deep learning with multi-modal sensory data have shown promising results in tasks such as object recognition, navigation, and human–robot interaction [30–32]. Palm-E [19], as mentioned previously, stands as a prominent example in this domain. It seamlessly integrates visual understanding with language processing, enabling effective interaction and control in various tasks. However, using multi-modal prompts requires workers to have certain skills, so the "one-shot imitation" approach is used to solve this problem, which only requires human demonstrations without the need to input multi-modal prompts.

One-shot imitation learning has emerged as a promising paradigm for enabling robots to learn tasks from a single demonstration. While traditional approaches often rely on extensive training data, recent advancements have focused on small sample learning techniques [33, 34], which aim to generalize from limited demonstrations. Meta-learning algorithms, in particular, have shown promise in enabling robots to adapt quickly to new tasks with limited data. Techniques such as model-agnostic meta-learning (MAML) [28] and gradient-based meta-learning have been applied to one-shot imitation learning, enabling robots to generalize from a few examples and adapt to novel tasks efficiently. Ref. [35] uses a Transformers attention mechanism and a self-supervised inverse dynamics loss, and its experiment shows better results than the ref. [28].

However, this method still has some shortcomings. First, although less demonstration data can be used in meta-training, for some complex tasks, a large amount of demonstration data is still required to achieve good performance. The solution to this problem is to use a pre-trained multi-modal robot agent. This multi-modal robot agent, after pre-training, can utilize multi-modal prompts to improve the robot's understanding ability in tasks and can learn semantic associations between different modalities such as images and text, so there is no need to use a large amount of demonstration data to complete training. MLM employs pre-trained multi-modal robot agents, which improve the understanding and processing capabilities of robots by utilizing multi-modal inputs, making them more flexible and efficient in generalization to new tasks. Compared with traditional meta-learning methods, MLM not only relies on a small amount of demonstration data but also can better adapt to complex task environments and demonstrate higher generalization ability. In the Transformer method [35], a single pick-and-place task in simulated environment has a $88\% \pm 5.0\%$ success rate with about 100 demonstrations in the training set. But in MLM, a simplest task which is similar to the task has a success rate of 88% with only about 20 demonstration videos in training set, and with the size of the training set grows to 90, the success rate of MLM increases to 96% (as shown in Section 4.4).

Second, this method requires high quality and diversity of demonstrations. If the demonstrations are not diverse enough or of poor quality, it may affect the generalization ability and performance of the model. This is because meta-learning methods may be limited by traditional convolutional neural network architectures when processing visual inputs, while MLM adopts TimeSformer [36], which has

better long-range dependency modeling ability and parameter efficiency and can better capture semantic information in images, thereby improving the performance of robots in visual tasks. How to use the TimeSformer is detailed and described in Section 3.

As for multi-modal robot agents, combining visual and language capabilities has garnered significant attention in recent years. VIMA [29], a robot agent used for multi-modal task specification, whose core idea is to instantiate different task specification paradigms (such as target conditions, video demonstrations, and natural language instructions) into the form of multi-modal prompts. VIMA adopts a multi-task encoder–decoder architecture and an object-oriented design. Specifically, VIMA learns a robot strategy $\pi(a_t|P, H)$, where H represents past interaction history, P represents multi-modal prompts, which are a sequence of text and images used to guide robots in performing specific tasks, and $o_t \in O$ and $a_t \in A$ represent observations and actions for each interaction step, respectively. VIMA encodes multi-modal prompts using a frozen pre-trained language model and decodes the robot's waypoint commands through a cross-attention layer.

Although VIMA has made significant progress in multi-task robot strategies, it also has some shortcomings. VIMA's visual tokenization scheme relies on the accuracy of object detection and may be affected by object detection errors. To address this issue, this study has designed the object detection module and applied the YoLoV8 (You Only Look Once v8) model (in Section 3), and YOLOv8 is an advanced real-time object detection algorithm that can quickly and accurately detect objects in images and locate their positions. By applying the YOLOv8 model, the goal is to improve the accuracy and robustness of object detection, thereby improving the performance of VIMA in handling multi-modal prompts. The high performance and effectiveness of YOLOv8 make it an ideal choice to help VIMA understand and execute tasks more reliably without overly relying on the accuracy of object detection.

In addition, VIMA has the function of one-shot imitation, but as mentioned earlier, it leans more toward precise reproduction of a trajectory rather than imitating how to complete a pick-and-place task as required. VIMA's multi-modal learning ability and generalization ability are excellent, but it has not been well applied in the one-shot imitation task. Therefore, MLM separates the understanding of videos, generates corresponding multi-modal prompts after obtaining the task demonstrated in the video, and then hands them over to the robot action prediction module to leverage VIMA's powerful robot strategy in the one-shot stimulation task.

Furthermore, VIMA's approach is based on a benchmark VIMA-BENCH in a simulation environment, which means that both the understanding of demonstration videos in the real world and the influence of various objective factors in the real environment are obstacles for VIMA to deploy in the real world. For this purpose, a dataset is captured in the real-world environment, which helps to better understand the physical scenes of the real world in conjunction with the VCM and object detection module, making up for VIMA's difficulty in understanding real-world images or demonstrations.

Overall, MLM combines VisionTransformer and pre-trained multi-modal robot agents to improve traditional visual meta-learning methods. By utilizing the ability of VisionTransformer to process image input and combining it with the multi-modal understanding ability of multi-modal agents, MLM can more effectively handle complex visual tasks and improve the performance and efficiency of robots in various tasks and environments. Compared to VIMA, MLM can combine video understanding with multi-modal prompts, enabling robots to learn tasks from a single demonstration and possessing strong generalization ability. Second, although VIMA also uses pre-trained language models, its processing of visual input may be limited by traditional convolutional neural network architectures, while MLM can better capture semantic associations between vision and language. In addition, MLM not only performs well in simulated environments but also can be applied in real world, providing a broader application prospect for applying MLM to actual robot systems. In contrast, although VIMA has achieved good performance in simulated environments, its generalization ability in the real world may pose certain challenges. MLM, a novel robot one-shot optimization paradigm, adopts advanced models and algorithms to handle multi-modal inputs and achieves innovation in practical application feasibility in the real world, significantly improving VIMA to achieve higher efficiency and accuracy compared to existing methods.

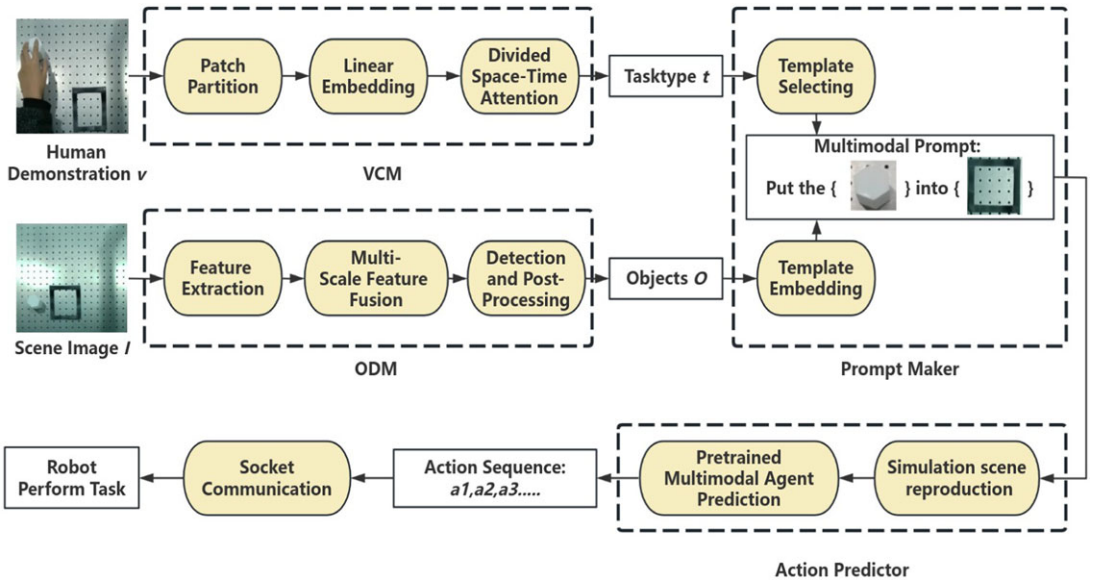

Figure 1. *The structure of MLM.*

3. Method

To combine the method of one-shot imitation and multi-modal prompt, MLM consists of a Video-Classification-Module (VCM), an Object-Detection-Module (ODM), a PromptMaker, and an ActionPredictor. The entry of MLM is a video that needs to be imitated to complete its task, as well as an image of the current scene that needs to be operated on (captured from a fixed camera shooting the scene). The video is input into VCM for classification, and then the PromptMaker selects the corresponding multi-modal prompt template based on the classification. The image is input to ODM to detect the types and positions of objects in the current scene, and the objects' information is then sent to PromptMaker, too. A set of corresponding types of objects are generated at the same coordinate position in the simulation environment. PromptMaker combines object information with multi-modal prompt templates to form a complete multi-modal prompt, which is ultimately handed over to ActionPredictor to predict the robot's action sequence. The overall structure of MLM is shown in Figure 1.

3.1. Human demonstration classification

This paper proposes a VCM to classify human demonstration videos ($\mathbf{V}$) into different tasks ($\mathbf{\Gamma}$). $\mathbf{\Gamma}$ comprises tasks such as simple manipulation, rearrangement, novel noun understanding, follow motion, same shape, and pick in order then restore. Each task can be classified using a human demonstration video v ($v \in \mathbf{V}$) to execute the classification process effectively, and the TimeSFormer architecture proposed by Bertasius et al. [36] is utilized.

The classification process begins by dividing a video v into F frames, each consisting of RGB images with dimensions 640 × 640 pixels. Subsequently, each frame is partitioned into 16 × 16 image patches. Within each patch, attention mechanisms are employed to capture both spatial and temporal dependencies, crucial for understanding the dynamic nature of the video content. Here, the divided space-time attention (DSTA) [36] is employed. DSTA is a crucial component in the TimeSFormer architecture designed to effectively capture attention in both the temporal and spatial dimensions of time series data. The core idea behind this mechanism is to separate the attention mechanism into two aspects: temporal attention and spatial attention, allowing for a better understanding of the underlying structure of time series data. Temporal attention focuses on capturing dependencies and pattern variations along

the temporal dimension, such as identifying periodicity, trends, and abrupt changes in the time series. On the other hand, spatial attention is concerned with relationships between different feature dimensions, capturing the correlation and importance of various features in the time series data. The advantage of DSTA lies in its ability to consider attention separately in temporal and spatial dimensions. By separating the attention mechanism, the network can more flexibly adapt to the diverse characteristics of time series data and more effectively capture patterns and relationships within the time series. This enables TimeSFormer to achieve superior performance in time series modeling tasks such as forecasting, anomaly detection, and other related applications. The TimeSFormer consists of L encoding blocks, within each block l, the attention computation is formalized by the following equation [36]:

$$\alpha_{(p,t)}^{(l,a)} = \mathrm{SM}\left(\frac{q_{(p,t)}^{(l,a)\top}}{\sqrt{D_h}} \cdot \left[k_{(0,0)}^{(l,a)} \left\{k_{(p',t')}^{(l,a)}\right\}_{\substack{p'=1,\ldots,N \\ t'=1,\ldots,F}}\right]\right) \tag{1}$$

In the equation, $\alpha_{(p,t)}^{(l,a)}$ represents the attention weights at head a, position p, and time t. The softmax (SM) activation function is applied to normalize the computed scores. $q_{(p,t)}^{(l,a)}$ denotes the query vector used in the attention weight computation. $k_{(p',t')}^{(l,a)}$ represents the key matrix, encompassing keys at all positions p' and times t' for layer l and head a. The latent dimensionality for each attention head is determined as $D_h = D/A$, where D represents the original dimension of the key and query vectors and A denotes the number of attention heads involved in the computation.

Expanding upon this framework, it is essential to highlight the significance of attention mechanisms in facilitating effective feature extraction across both spatial and temporal dimensions. By dynamically attending to relevant image patches over time, the model can discern meaningful patterns and temporal dependencies inherent in the video data, thus enabling robust classification performance across diverse task categories. Moreover, the modular architecture of Vision Transformers offers scalability and flexibility, allowing for seamless integration of additional modalities or architectural enhancements to further improve performance and adaptability in real-world scenarios.

To train the VCM, a dataset is captured. The videos in this dataset consist of five types of human demonstration videos, corresponding to six common pick-and-place tasks. In MLM, any pick and place task is decomposed into three main steps: understanding the conditions given by the task (understanding the current scene), understanding the goal of the task (understanding multi-modal prompts), and reasoning how to complete the task (predicting action sequences). According to whether each of the three steps of the task is challenging, all tasks are divided into six categories, which were shown in Figure 2. This classification method helps to test which step of MLM is done well and which step is not done well, identify shortcomings, and help to improve MLM targeted in subsequent research. For these six tasks, all objects in the scene are classified as dragged objects or base objects. The dragged objects refer to the objects being picked and placed, while the base object refers to the placement location of the dragged object in the scenes, which is often represented as a component in actual scenes. The descriptions of the six tasks are as follows:

- Task 1: Simple manipulation – The robot picks the dragged object up and puts it into the base object. This is the simplest pick-and-place task. This task can be used to verify whether the robot can complete a pick-and-place task and test the pick-and-place accuracy. It does not impose high inference requirements on the three steps of the task. In real-world scenarios, peg-in-hole assembly is one of the examples of this task.
- Task 2: Rearrange – The robot picks the dragged object up and places it into a defined location, which may be taken by another dragged object. This task poses certain inference requirements for understanding the current scene. This task requires the robot to determine whether the position defined in multi-modal prompts has been occupied by another object in the current scene. If so, the robot needs to first move away from the occupying object and then put the dragged object

shown in prompts into the defined location. In real-world production line, robots may need to rearrange parts to adapt to different vehicle models or production plans. In electronic manufacturing, robots are used to rearrange electronic components and circuit boards to complete different assembly tasks.

- Task 3: Novel noun understanding – In this task, the robot receives an image with both dragged objects and base objects. A novel noun is given to the dragged object's image and another novel noun to the base object's image. The novel nouns are randomly fabricated, which are not trained before. The robot needs to pick up the dragged object defined by the novel noun and place it into the base object. This task poses certain reasoning requirements for understanding the step of multi-modal prompts. It provides object a novel definition (the novel noun) in multi-modal prompts, and the robot needs to understand what this novel definition refers to and pick and place it. In industrial applications, it is possible to encounter situations where a new part is not named before and a new name is created. In this case, the novel noun understanding method in this task can be used to handle new parts.
- Task 4: Follow motion – Different kinds of dragged objects are present in the current scene, but only one kind needs to be manipulated. One of the two objects will be pointed out in multi-modal prompts that it needs to be picked and placed several times by the robot, and another needs to remain stationary, the position of place is demonstrated in the demonstration video. This task requires certain requirements for the predicting action sequence step, as it needs to be replicated in order of the position of the dragged object in the demonstration video. In real-world scenarios, sorting tasks is a good portrayal of this task. Robots need to identify different objects and then pick them up and place them into different target containers according to requirements.
- Task 5: Same shape – In this task, different kinds of dragged objects appear in the scene. In this task, the robot needs to pick up dragged objects with the same shape and place them into a specified base object. This task places a certain requirement both in understanding multi-modal prompts step and understanding the current scene step. The robot needs to understand the meaning of text "same shape" and determine wich two objects in the current scene have the same shape. In real-world scenarios, a robot may need to execute some complex sorting tasks, in which multiple objects appear at once in a scene.
- Task 6: Pick in order then restore – In this task, there are different base objects, and one of these base objects initially has a dragged object in it. The robot needs to pick the dragged object up and place it into the defined base objects sequentially and finally put it back to its initial base object. The multi-modal prompts for this task include the prompt "restore it into its initial container (as shown in Section 3.3 task 6)." Therefore, this requires the robot to first understand the meaning of the prompt "initial container." Second, the robot needs to understand which is its initial container in the current scene. Finally, the robot needs to correctly predict the motion sequence to comply with the placement sequence required by the prompt. So this task requires certain requirements for all three steps. As an example of this task, in electronic manufacturing, a pallet (initial base object) on the conveyor has a PCB board (dragged object) on it. A robot picks up the PCB board and puts it onto different stations (base objects) sequentially for processing. After that, the robot picks the PCB board up and puts it back to the pallet.

These six tasks are categorized into four levels. Task 1 is classified as a level 1 task because it is a simple pick-and-place task. Task 2 and Task 3 are classified as level 2 Tasks because they put certain requirements for robots in terms of understanding steps. Task 4 has certain requirements for the predicting action sequence step (a reasoning step), and in a pick-and-place task, predicting the action sequence is the relatively most difficult of the three steps, because problem-solving requires understanding the problem (the other two steps); Task 5 places requirements for both of the step understanding multi-modal prompts and the step understanding the current scene. So these two tasks have higher requirements

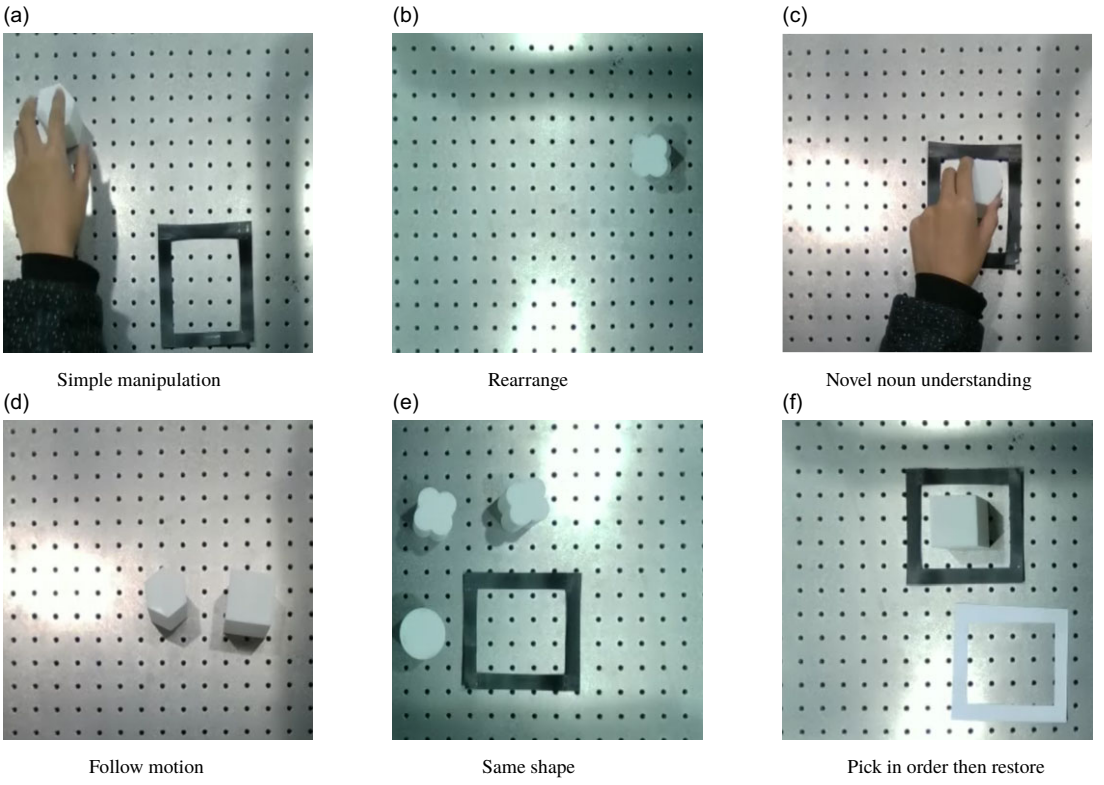

Figure 2. *Demonstrations of different tasks. (a) Simple manipulation – Level 1 Task. In the video, the demonstrator picks an object (dragged object) up and places it into a black or white box (base object). (b). Rearrange – Level 2 task. A video where the positions of different objects are recorded. The demonstrator rearranges the objects. (c) Novel noun understanding – Level 2 Task. The objects in the video is unknown. Novel nouns are given to name the objects. The video is the same as that used Task 1. (d) Follow motion – Level 3 Task. There are two objects in the video, but only one needs to be manipulated. The dragged object is moved sequentially, with brief pauses in the video to record its location. (e). Same shape – Level 3 Task. The demonstration video has two types of objects. The demonstrator picks the same type of objects up and puts them into the base object. (f) Pick in order then restore – Level 4 Task. In the video, the dragged object is in the initial base object and moved to different base objects sequentially. After that, the dragged object is placed back to the initial base object.*

compared to the tasks of the previous level, that's why they are in level 3. Because the Task 6 requires certain requirements for all three steps, so the Task 6 is classified as highest level 4.

Theoretically, each task needs a corresponding demonstration video for the robot to understand and imitate the task; therefore, for the six tasks, there should be six types of videos. Because the same video can be used to demonstrate Task 1 and Task 3, only are five types of videos needed. Task 1 and Task 3 can be specified manually in the human demonstration video. The practical significance of using five types of videos is to improve the efficiency and accuracy of video classification.

3.2. Object detection

In addition to classifying the demonstration videos and obtaining the task γ, detecting objects in the current scene is also needed to distinguish the dragged object and the base object in γ. Besides, VIMA is based on simulation environments and cannot directly detect objects in the real-world scene. Therefore,

Table I. *Hash table of classes to categories.*

Class	Category
Square	Dragged object
Round	Dragged object
Flower	Dragged object
Pentagon	Dragged object
Hexagon	Dragged object
Letter G	Dragged object
Letter E	Dragged object
Letter A	Dragged object
White container	Base object
Black container	Base object

ODM is proposed. A RealSense depth camera system is used for capturing an RGB image and depth information of the scene.

After obtaining a RGB image I_c of the current scene, YoLoV8 is used to detect objects. In ODM, objects are classified into two categories: dragged object and base object. Each object is also classified into a unique class according to its shape. The classes include square, round, flower, pentagon, hexagon, letter L, white container, and black container. These two classifications are connected through a hash table as shown in Table I.

YoLoV8 detects objects in I_c to obtain a box of each object (this box contains the pixel coordinates, the class, etc.), and ODM retrieves the class in the box and maps it to its category based on the hash table. However, the object information obtained in YoLoV8 is only planar, and the heights of the dragged objects are still unknown. This can cause the robot to fail to pick up the dragged object or collide with the dragged object. Therefore, the depth information is used to calculate the heights of the dragged objects. For each box detected from YoLoV8, if its category is the dragged object, its height information H_o will be added to the box. H_o is calculated as:

$$H_o = H_c - D\,(x, y) \tag{2}$$

where $D(x, y)$ is the depth value of the coordinates (x, y) of the box. In this equation, all parameters use the same robot base coordinate system. If the category of the box is base object, its height value will be set to 0 and added to the box.

Flat RGB images only provide the position of the object without its height, this can easily cause instability in picking. If the height is too high, the robot end may not be able to reach the object, and if it is too low, it may compress and damage the robot end or object. Having the depth information to calculate the object height effectively solves this problem.

In summary, the ODM represents an effective framework for object detection. By leveraging the YoLoV8, coupled with RealSense depth camera system to have height information of objects, MLM empowers robots to perceive, interact with, and manipulate objects in their environment.

3.3. Prompt maker

Once human demonstration videos have been processed and categorized into different tasks and the objects are detected; this paper proposes a prompt maker to generate multi-modal prompts that provide actionable instructions for the robot. These prompts include both visual and text information, enabling the robot to comprehend and execute a task effectively.

For each task, using the multi-modal prompt template [29], this paper proposes the following prompts:

- Task 1: Simple Manipulation – Put the {dragged object} into the {base object}.
- Task 2: Rearrange – Rearrange to this {an image of scene}.
- Task 3: Novel Noun Understanding – This is a {dragged object novel noun A}{dragged object}, this is a {base object novel noun B}{base object}, put the {dragged object novel noun A} into {base object novel noun B}.
- Task 4: Follow Motion(Level 3) – Follow the motion of {dragged object}, {several images of scenes}.
- Task 5: Same Shape(Level 3) – Put the same shape objects into {base object}.
- Task 6: Pick in Order then Restore(Level 4) – Put the {dragged object} into {base object}, then {base object}, finally restore it into its initial container.

The placeholders in the prompt templates are all image modalities except the novel noun prompt. Image prompts are divided into scene images or object images. The scene images are all captured directly by Realsence camera. For the object images, the prompt maker will extract the images of the objects from the boxes detected by ODM and use them to fill the placeholders. In MLM, integrating information from multiple modalities is divided into two steps. The first step is done in the prompt maker, where the prompt selects the template corresponding to the task (a string with one or more placeholder), and then corresponds the placeholder to the prompt of the image or text (new new new task) modality. For example, when displaying multi-modal prompts, the text template of the entire multi-modal prompt is obtained first, and for the placeholder, the prompt of the corresponding image or text (novel noun task) modality is displayed instead of displaying a string of placeholder, thus completing the generation of multi-modal prompts. The second step of integrating multi-modal information will be done at the beginning of the action prediction. Multi-modal fusion provides richer and more diverse semantic representations by combining multiple information sources. For example, in Task 1, images intuitively provide visual information to specify the object to be operated on, which is significantly more user-friendly for human–robot interaction, while text descriptions accurately supplement language information to express the actions that need to be taken. The combination of the two enables the model to more accurately understand and describe the image content. If only non-multi-modal approaches are used, it is not possible to achieve both the intuitiveness of object specification and the conciseness and accuracy of text description in a one-time prompt. In addition, fusing information from multiple modalities can make MLM easier to train. As stated in Section 2, compared to the ref. [35] that only uses video modality methods, MLM has a significantly reduced requirement for the training dataset and higher success rate than the ref. [35]. This is because, as mentioned earlier, MLM uses different analysis methods for information from multiple modalities. For the input of video modalities, MLM first uses VCM for video classification to obtain the type of demonstration task, while the detailed details of the task (such as the object being operated on) are analyzed by other modules. And [35] analyzes and considers all details within the same modality, which requires a huge amount of learning.

In summary, prompt maker enables effective communication and learning between humans and robots. Through the proposed multi-modal prompts, the prompt maker empowers robots to perform pick-and-place tasks in real world.

3.4. Robot action sequence prediction

The proposed multi-modal prompt has three kinds of formats – text, image of a single object, and image of a scene. To predict robot action sequence, a pre-trained VIMA [29] is used. The pre-trained VIMA adopts an encoder–decoder architecture. First, encode multi-modal prompts and unify all prompts into token sequences. In this process, text, individual object images, and scene images are all transformed into a unified representation that the model can process. When making predictions, VIMA's attention mechanism focuses on the relationship between token sequences and historical action sequences. Specifically, VIMA captures and utilizes these relationships by calculating key and value sequences in prompts, as

well as querying sequences from robot interaction history. The encoder part is responsible for extracting and compressing input prompt information, while the decoder part generates predicted action sequences based on historical actions and current prompts. Through this approach, MLM can effectively integrate and utilize multi-modal information, improving the accuracy and robustness of task prediction. The integration of multi-modal information is completed.

Since VIMA is based on the simulation environment VIMA-Bench [29] and cannot used on real-world scenes directly, reproducing the real-world scene in VIMA-Bench is needed. This is easy because all information of objects in the scene is already obtained in ODM and the VIMA-Bench can generate the objects according to the information of objects. By setting the coordinate system of VIMA-Bench to be consistent with the robot base coordinate system in the real world, a scene that is consistent with the scene in the real world is generated in VIMA-Bench. In this way, VIMA can be applied to predict the robot action sequence. The action sequence includes the initial status of a robot, robot action commands such as "movej," and logic control commands which are used to control the end to pick up or place dragged objects.

The action sequence of the robot can be used directly in VIMA-Bench, a robot in VIMA-Bench performs pick-and-place tasks according to the action sequence. As for the real-world robot, the socket communication is used to send the action sequence to an ABB robot. The robot executes the sequence using rapid motion functions and ultimately performs the task demonstrated in the real world.

4. Experiment

4.1. Baselines

Given the nascent status of the VIMA-BENCH benchmark and the relatively limited adoption of this methodology, the study adopts VIMA's one-shot imitation task as the baseline for evaluation. The one-shot imitation technique employed by VIMA involves the extraction of select frames from the video to replicate the trajectory exhibited within. To ensure fairness in comparison, a standardized procedure where the initial and final frames of each pick-and-place maneuver within the scene are utilized as inputs for VIMA's one-shot imitation approach. Furthermore, MLM explicitly specifies the object to be manipulated, aligning MLM with the same scene and human demonstrations.

In parallel with the evaluation using VIMA's one-shot imitation, MLM establishes communication with an ABB robot using a similar approach. Subsequently, MLM conveys a sequence of instructions derived from the same set of initial and final frames utilized in the VIMA-based evaluation. This enables a direct comparison between the imitation performance of VIMA and the execution capabilities of the ABB robot, thereby facilitating a comprehensive assessment of the efficacy and applicability of both methodologies within the context of robotic learning tasks.

This comparative analysis serves to illuminate the strengths and limitations of each approach, shedding light on their respective performance metrics and suitability for real-world deployment scenarios. By leveraging both VIMA's one-shot imitation and the ABB robot's execution capabilities, MLM aims to garner insights into the efficacy of imitation-based learning paradigms and their potential implications for robotic automation across diverse application domains.

4.2. Datasets and experiment setup

The dataset is all shot on an optical platform, in normal conditions, existing methods typically use UCF101 [37] as the training dataset for this type of VCM and COCO [38] as the training dataset for the object detection model training. However, since video classification and object detection requirements are specific to different human demonstration videos and objects, a dataset is created.

The dataset is a comprehensive collection consisting of two main components: human demonstration videos and annotated images of objects. The video dataset encompasses approximately 450 videos, showcasing demonstrations of five different tasks, with each task represented by around 45 videos shot

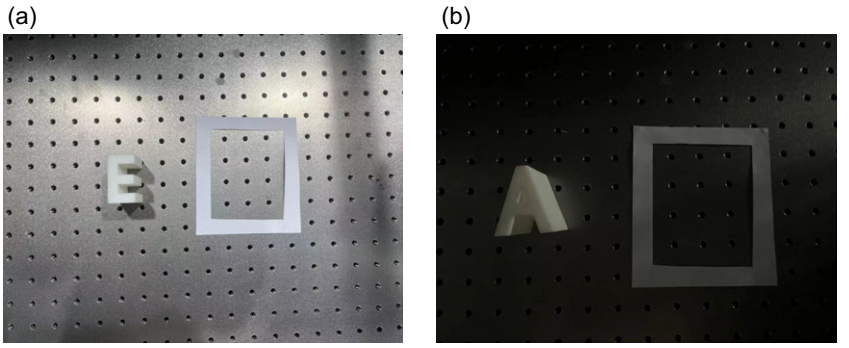

Figure 3. *Videos in different lighting conditions.*

in an environment with lighting and 45 videos shot in an environment with a dim lighting condition. The two kinds of videos with different lighting environments will be used to verify the robustness of MLM in different lighting scenarios. These two environments are shown as Figure 3. The video datasets captured under two different lighting conditions are trained together. These videos vary in duration but maintain a consistent frame size of 640 x 640 x 3 pixels, ensuring compatibility and ease of processing.

In addition to the videos, the dataset includes annotated images of objects, which are categorized into two types of base objects (black square zone and white square zone) and eight types of draggable objects (square, flower, round, pentagon, hexagon, letter G, letter E, letter A). Each category contains about 20 images, totaling around 200 annotated images. These images also maintain a resolution of $640 \times 640 \times 3$ pixels.

The strengths of the dataset lie in its diversity, consistency, and richness of annotations. By showcasing demonstrations of five different tasks, the video dataset offers a wide range of actions and scenarios, making it suitable for various applications in computer vision and robotics. Moreover, the consistent frame size of the videos simplifies preprocessing tasks and ensures uniformity across the dataset, facilitating training and evaluation processes. The annotated images provide detailed annotations for base objects and draggable objects, enabling precise object recognition and tracking.

Overall, the dataset serves as a valuable resource for researchers and developers working in fields such as action recognition, object detection, and human–computer interaction. Its diverse content, consistency, and rich annotations make it well suited for training and testing MLM in pick-and-place tasks.

For the setting of noisy data in the experiment, first, as described in Section 3.1, there are two objects in Task 4, and only one of them is designated as the object that needs to be operated on. In Task 5, in addition to two objects with the same shape, there are also objects with different shapes; Task 6 not only specifies the various base objects in the specified order but also includes some base objects that will not be used. These are all noisy dataset in the experiment. In addition, two different light conditions were set up in the experiment. The above factors are all used to test the robustness of MLM.

The total experimental setup is shown in Figure 4. MLM runs in Ubuntu 20.04 desktop with Intel Core i9 CPU and NVIDIA 3060Ti GPU. The experimental configuration includes an Intel Realsense d435 RGBD depth camera, several instances of dragged objects and base objects, ABB IRB120 robot and an optical platform.

D435 depth camera is selected for its ability to capture both RGB (color) and depth information simultaneously. Its RGBD capabilities enable precise object recognition and tracking, essential for tasks such as object manipulation and scene understanding. Depth information enriches the data captured, allowing for more accurate perception and spatial understanding in various applications. As mentioned

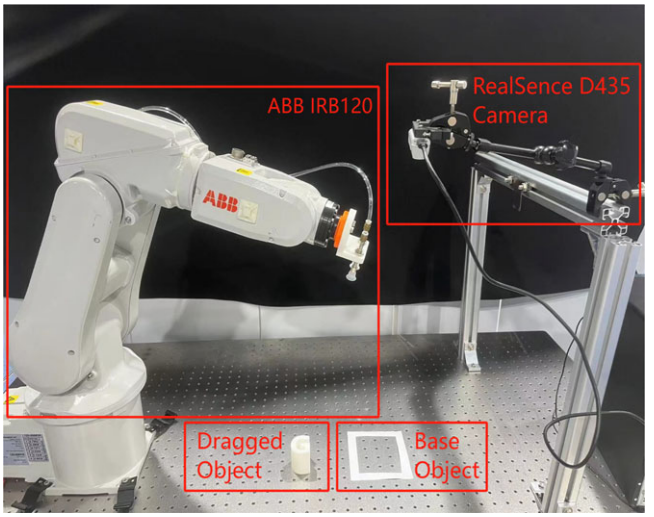

Figure 4. *Experimental setup.*

earlier, having only RGB images is not enough. RGB images can provide the position of objects appearing in the scene through object detection, but because they are 2D, they cannot represent the height of objects. Picking without height information may result in situations where the end does not successfully contact and connect to the object, or excessive compression with the object can cause damage to the robot end. Therefore, it is important to use a D435 depth camera to obtain depth information and then use the depth information to obtain the height of the object, thereby obtaining the pick height at the end of the robot. In addition, the camera shooting range is set to 370 cm x 370 cm and the camera height to 45 cm during the data collection process of all experiments.

The ABB IRB120 robot is integrated into the experimental setup to enable physical interaction with the environment. As a versatile and compact industrial robot, it can perform a wide range of manipulation tasks with precision and efficiency. By incorporating the robot into the setup, researchers can validate and evaluate their algorithms in real-world settings, bridging the gap between simulation and practical deployment. Additionally, the use of a robot facilitates automation and repeatability of experiments, streamlining the research process.

The choice of hardware platform is crucial for ensuring computational power and efficiency in processing data and running algorithms. The Intel Core i9 CPU provides high-performance computing capabilities, suitable for tasks such as data preprocessing, feature extraction, and algorithm execution. The NVIDIA 3060Ti GPU accelerates parallel processing tasks, particularly in deep learning applications, enabling faster training and inference of machine learning models. Ubuntu 20.04 is selected as the operating system for its stability, compatibility with research tools and libraries, and robust support for development environments.

In addition to these, the experimenters who are the task demonstrators are required to maintain the same attire as during the training set shooting during the demonstration, as this may bring some errors to the judgment of the demonstration. They are required to perform stable and smooth demonstrations with one hand, without unnecessary body parts or movements, which will effectively help improve the accuracy of the demonstration and thus enhance the accuracy of the experiment. Maintaining a positive and positive mindset at the same time will greatly contribute to the efficiency of the team collaboration.

By assembling this experimental configuration and the dataset, researchers can conduct comprehensive experiments to test the reliability and efficiency of MLM. The integration of diverse components facilitates experimentation across multiple domains, from perception and manipulation to control and automation, ultimately driving innovation and progress in intelligent systems.

Table II. *Pseudocode for timesformer training.*

Step	Description
1	Initialize Timesformer model components: $\text{Timesformer}_{\text{backbone}}$, $\text{Timesformer}_{\text{head}}$.
2	Prepare training dataset with data augmentation ($\text{Dataset}_{\text{train}}$).
3	Prepare optional validation dataset ($\text{Dataset}_{\text{val}}$).
4	Initialize optimizer (Opt) with scheduler (LR_Scheduler) and regularization.
5	Load pre-trained weights if available (pre-trained_Weights).
6	During each training epoch (epoch): a. Set the model to training mode. b. Iterate through training batches (batch): i. Compute predictions ($\hat{y}$) and gradients. ii. Update model parameters (θ). c. Update the scheduler and log progress. d. Optionally evaluate on validation set.
7	Save best model and optimizer state.
8	Print training completion message.

Table III. *Pseudocode for YoLoV8 training.*

Step	Description
1	Initialize YoLoV8 model YOLOv8 with parameters.
2	Prepare training dataset Dataset with augmentation and preprocessing.
3	Define loss function $\mathcal{L}$ for YoLoV8.
4	Initialize optimizer Opt and learning rate scheduler LR_Scheduler.
5	For each training epoch: a. Train model on batches. b. Validate the model if possible. c. Update learning rate. d. Print progress and metrics.
6	Save trained model YOLOv8 and optimizer state Opt.

4.3. Training

In MLM, the VCM and ODM need to be trained. For both datasets of human demonstration videos and image annotations, the training-to-test ratio is 9:1. The VCM is trained on the PaddlePaddle AIStudio platform, using a Tesla V100 32G GPU with an initial learning rate of 0.0005 and a batch size of 32, and the training process is shown as Table II.

The ODM is trained on NVIDIA 3060Ti, with an initial learning rate of 0.0002 and a batch size of 4, and the training process is shown in Table III.

4.4. Evaluation and results

In order, the experiment first evaluates whether ODM and VCM worked successfully in the case of small data samples. For six different tasks, the accuracy of the multi-modal prompts generated by MLM is recorded after receiving input in both conventional samples' sizes (approximately 45 samples for both bright condition and dim condition, totally 90 samples) and small data samples. MLM(S) represents Multi-modal Learning Method with a very small sample size (give $\leq$10 video samples per task for both bright condition and dim condition, totally $\leq$20 samples to VCM's training dataset and $\leq$5 image samples per object to ODM's training dataset), MLM(N) represents Multi-modal Learning Method with a normal sample size (give about 40 video samples per task for both bright condition and dim condition,

Table IV. *Success rates of pick-and-place of different tasks in VIMA-BENCH.*

	VIMA	**MLM(S)**	**MLM(N)**
Simple manipulation	68%	88%	96%
Rearrange	68%	80%	96%
Novel noun	64%	68%	92%
Follow motion	72%	72%	84%
Same shape	-(no success)	68%	80%
Pick in order then restore	36%	64%	76%

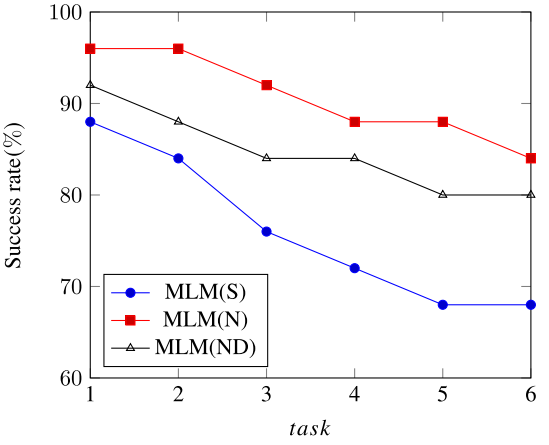

Figure 5. *Success rate of prompt generating.*

totally 80 samples to VCM's training dataset and about 20 image samples per object to ODM's training dataset). Both MLM(S) and MLM(N) are experimented within a bright light condition. MLM(ND) represents Multi-modal Learning Method with the same sample size as MLM(N), but within a dim light condition. For the data collection method of this experimental result, the success rate of the experimental results is collected by manually judging whether the generated multi-modal prompts met the expectations. If the experimental result is successful, then the text modality in the multi-modal prompt should be the same as the template, and the image modality in the prompt should be consistent with the image type represented by a placeholder in its template. The result is shown in Figure 5. It can be seen that when the sample size reaches a very small value, and MLM still has a high accuracy in generating multi-modal prompts. In addition, the success rate of prompt generating still maintaining a relatively high level when light condition becomes challenging. This is because the trained dataset contains sample data under dim lighting conditions, so MLM can still maintain a relatively accurate level under the challenging light conditions.

Then, the experiment tests the success rate of robots learning from demonstration and imitating to complete the pick-and-place tasks in VIMA-BENCH and the physical world. In two environments, experimental results data are recorded through human observation. Whether the object is placed into the target base object (each edge and vertex of the object is within the range represented by the base object) is the standard to determine if the task is successful, and the success rate is calculated by calculating the proportion of successful attempts to the total number of experiments. In addition, experiments in the real world also recorded the success rate under dim lighting conditions in order to test if MLM is robust while facing challenging light conditions. The results are shown in Tables IV and V.

It is not difficult to see that VIMA's one-shot imitation method does not achieve satisfactory results in real-world physical scenes. For the four levels of the tasks in VIMA-BENCH, the success rates (Level 2 and Level 3 are the average success rates of Task 2,3 and Task 4,5) are shown in Figure 6.

Table V. *Success rates of pick-and-place of different tasks in physical world.*

	VIMA	MLM(S)	MLM(N)	MLM(ND)
Simple manipulation	60%	88%	96%	92%
Rearrange	60%	80%	96%	92%
Novel noun	56%	68%	92%	84%
Follow motion	72%	68%	80%	80%
Same shape	-	52%	68%	64%
Pick in order then restore	28%	60%	68%	68%

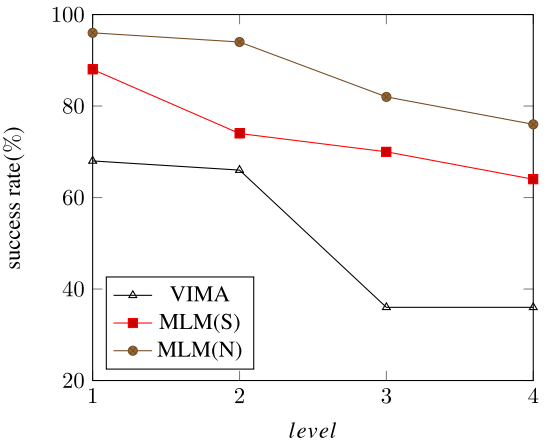

Figure 6. *Success rates of pick-and-place of different levels' tasks.*

Because the one-shot imitation method of VIMA reproduces the trajectory of an object, when performing the demonstrated task, if the object in the real scene remains unchanged (with the same pose as the demonstration scene), VIMA's method is likely to complete the pick-and-place task. However, if the object's position changes, VIMA's method can also find how to pick the target, but it cannot be placed in the target position according to the task requirements. When there are two or more objects to be picked, VIMA's method can only track the trajectory of one object and cannot perform the picking and placement of multiple objects in tasks such as "same shape." As shown in Figure 5, MLM's success rate of prompt generating is of a high tone; it accurately obtains and reproduces the types and poses of objects through the ODM and classifies the tasks performed in human demonstration videos through VCM. When reducing the sample size to a minimum, MLM is only affected by some errors in detecting scene objects and classifying human demonstrations. This is because in some tasks that have a high demand for accurate object recognition (such as tasks "novel noun" and "same shape"), imprecise classification of objects or tasks can affect object classification and picking pose. However, there is still a high success rate for tasks at level 1. In addition, comparing the data of MLM (N) and MLM (ND) in Table V, we can find that MLM still maintains a high success rate of the overall task even under poor lighting conditions, which means that MLM is robust to the factor of challenging lighting.

MLM's superiority over existing techniques highlights its transformative potential in real-world robotics applications. While VIMA excels in replicating object trajectories, its limitations become apparent in scenarios involving dynamic object positions or multiple objects. In contrast, MLM integrates cutting-edge object detection and task classification modules, enabling precise identification of objects and tasks essential for successful pick-and-place operations. What sets MLM apart is its remarkable adaptability even with minimal training data. This resilience to errors in object detection and task classification is particularly valuable in industries where tasks demand precise execution despite varying

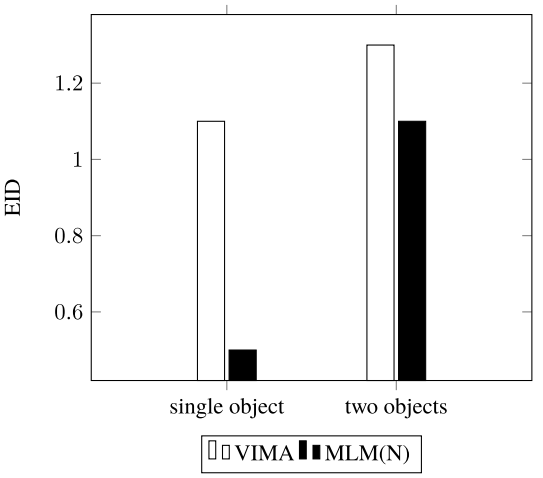

Figure 7. *EIDs of pick-and-place.*

object configurations. By overcoming these challenges, MLM promises to revolutionize robotics by enhancing performance and efficiency in complex pick-and-place tasks. Ultimately, MLM's practical significance lies in its ability to address real-world challenges, driving the advancement of automation solutions that are not only reliable but also adaptable to the ever-evolving demands of modern industries.

The experiment also measures and compares the *EID* (Error In Distance) of MLM and VIMA's one-shot imitation method in placing objects in the physical world. This part of the experiment is used to evaluate the accuracy of the method's placement position when placing a single or two objects. Since this part of the experiment is used to evaluate the accuracy of placement, the part of robot reasoning has been minimized as much as possible. Therefore, the simplest tasks 1 and 5 are adopted here to evaluate the accuracy of method placement. The experiment condition is the same as mentioned above, and the experiment obtains the center-point coordinate data of the object in the test results by capturing the current scene again and then using ODM to detect the object again. The *EID* in the simplest task 1 for a single dragged object and Task 5 for two dragged objects are defined as:

$$EID = \frac{\sum_{i=1}^{N} \sqrt{(x_i - x_i')^2 + (y_i - y_i')^2}}{N} \tag{3}$$

where N is the number of calculated samples, (x_i', y_i') stands for each sample's center-point coordinate of the base object, and (x_i, y_i) stands for the coordinate of the dragged object. When there are two dragged objects, the final EID is calculated as the average of each single object. The results obtained are shown in Figure 7. It can be seen that MLM's accuracy of placing position is better than VIMA's trajectory reproduction method, but when two objects need to be placed, it will obtain some collision problem, so there is some incensement in EID for both VIMA and MLM.

This experiment's significance extends beyond the laboratory setting, offering insights into the practical challenges of robotic pick-and-place tasks. By demonstrating MLM's ability to achieve higher accuracy in object placement, especially in scenarios involving single objects, its potential is underscored to enhance efficiency and precision in real-world applications. Moreover, the observed increase in EID when handling multiple objects underscores the importance of further research to address collision avoidance and improve the robustness of robotic manipulation techniques. Ultimately, these findings contribute to advancing the field of robotics by refining algorithms and methodologies to meet the demands of increasingly complex automation tasks in various industries.

Table VI. *The computational complexity of VCM and ODM.*

	VCM	ODM
FLOPs(B)	590	8.1
Params(M)	121.4	3.0

Table VII. *The inference duration of modules in different tasks.*

Inference duration(ms)	VCM	ODM	Action predictor	VIMA (one-shot imitation)
Simple manipulation	369.1	53.6	433.6	879.6
Rearrange	352.2	52.7	485.4	847.3
Novel noun	431.6	54.2	660.0	861.7
Follow motion	449.6	61.3	835.0	868.5
Same shape	590.0	65.6	849.8	903.2
Pick in order then restore	643.4	68.1	867.7	885.4

In addition, the computational complexity evaluation of VCM and ODM in MLM is shown in Table VI. This table is used to display the parameter quantity and FLOPs (floating point operations) evaluation of VCM and ODM in the hardware environment described in Section 4.2. This evaluation data is collected and output through programming.

Due to the role of VCM in video classification, which involves video processing, it is not difficult to see that both the number of parameters and FLOPs will be much larger than ODM. For the robot action predictor module, as it also involves trajectory planning and torque calculation of the robot, and changes in conditions between different tasks can affect computational complexity, the inference time of the module in each task is used to represent its computational complexity. Since this part of experiment focus on the computational complexity and processing speed of the method, the experiment does not pay attention to the success of the task in this evaluation. Due to the fact that the entire MLM method involves the coordination of several modules and contains different models, it is difficult to measure the processing speed of MLM on a unified scale. So when processing a task, the inference duration of each module is recorded as an indicator to measure processing speed. As shown in Table VII, it is shown that as the difficulty of the task increases, the inference duration of VCM, ODM, and VIMA's one-shot inference methods remains stable and almost unchanged, while the inference duration of MLM's action predictor shows a slow upward trend. This is because the action predictor needs to generate prompts based on different tasks, and as tasks become more complex, to correctly complete more complex tasks, the robot's action sequence should also become more complex.

Overall, this part of the experiment highlights the efficacy and practicality of imitation-based learning paradigms, particularly in robotic automation tasks. MLM shows promise for real-world deployment across diverse application domains, offering improved performance and reliability compared to existing techniques.

And the computational complexity of two modules is evaluated, VCM and ODM, within the MLM. Table VI presents their parameters quantity and FLOPs analysis. VCM exhibits significantly higher FLOPs and parameters compared to ODM due to its role in video classification. For the robot action predictor module, computational complexity is represented by inference time in each task, influenced by trajectory planning and torque calculation, as well as task conditions. Table VII displays the inference duration of various modules in different tasks. While VCM and ODM maintain stable inference durations as task difficulty increases, MLM's action predictor shows a slight upward trend due to the need for prompt generation and complex action sequences for more challenging tasks.

5. Conclusions and discussion

A novel multi-modal pick-and-place task learning method grounded in human demonstrations is proposed in this paper. By leveraging categorization techniques for human demonstration videos and integrating real-world scenarios with the VIMA-BENCH multi-modal prompt framework, the method successfully achieved the prediction and control of robot action sequences. The experimental findings demonstrate the effectiveness and practicality of MLM, both in simulated environments and real-world robot experiments.

VCM utilizes TimeSFormer's DSTA mechanism (mentioned in Section 3.1) to effectively capture spatiotemporal dependencies in videos, making regression tasks more efficient. And through task classification, it ensures that MLM can find corresponding strategies to generate multi-modal prompts and further understand prompts to complete the task. In ODM, the height information of object picked in the scene is obtained through depth cameras, and after object detection, the height information is added to the box of the object (Section 3.2). This enables all objects in the current scene to reproduce their size and position relationships well in the simulation environment, which helps predict the subsequent robot action sequence. And ODM divides all objects into dragged objects and base objects, which enables the specified position of the placeholder in the multi-modal prompt template (Section 3.3) to accurately and efficiently obtain multi-modal prompts of the specified type (dragged objects or base objects). As all the necessary conditions for prediction (scene, object information, multi-modal cues) have been obtained earlier, the final robot action sequence is obtained through a pre-trained VIMA (Section 3.4).

MLM exhibited promising results across different task complexities, as evidenced by the relatively slow decrease in success rates on VIMA-BENCH with increasing task difficulty levels. This suggests a certain degree of robustness in MLM, enabling it to handle tasks of varying complexities with reasonable success rates. The ability to generalize across tasks of different difficulty levels is crucial for the practical deployment of robotic learning systems in diverse real-world scenarios.

A dataset for industry pick-and-place tasks is created since existing methods typically use UCF101 for VCM training and COCO for object detection model training, which does not meet specific scenario requirements. The created dataset consists of two components: human demonstration videos and annotated images of objects. This dataset plays a crucial role, as it connects it to real-world scenarios.

Moreover, the experiments in real physical environments further validate the efficacy of MLM. The satisfactory success rates and EID results obtained in real-world robot experiments underscore the reliability and accuracy of MLM in real-world settings. The computational complexity is also evaluated in the experiment to demonstrate the efficiency of MLM. These findings highlight the potential of MLM for seamless integration into industrial and domestic robotic automation applications.

However, MLM still has certain limitations. On the one hand, MLM relies on pre-trained multi-modal prompting robot agents (VIMA [29]) for predicting robot action sequences. This will limit the performance of MLM to some extent by pre-trained VIMA. For example, MLM divides the pickup and place tasks that may occur in the industry into six categories which are included in the 17 tasks VIMA-Bench supported, but for other types of tasks, additional training of VIMA is required. Therefore, a future research direction is to make better pick-and-place task classifications and pre-train them in VIMA. This will significantly improve the adaptability of MLM to various tasks in the future.

On the other hand, the variety of objects is limited in current MLM. That's because some features of objects like shape and material that current VIMA-Bench supports are limited. If objects in VIMA-Bench are used to represent objects in real scenes that are not supported by them, it may cause significant errors, because for different shapes or materials of an object, pick position of the object may vary. Therefore, in the current methods, the types of objects in the dataset still need to be enriched. So as the background supported in the VIMA-Bench. One possible research direction for this limitation is to expand the objects that VIMA-Bench supports as much as possible, enriching its shapes, materials, etc. Another is to use 3D reconstruction to reconstruct objects in the scene in VIMA-Bench, to construct the shape of any object or background that appears in the scene, and the material needs to be analyzed through a visual model in this way.

By addressing these research directions, researchers can continue to advance the most advanced technology of imitation-based robot learning and pave the way for more autonomous and intelligent robot systems.

In conclusion, this paper proposes a new MLM for robot pick-and-place tasks. This method aims to enhance the intelligence of robot pick-and-place through one-shot stimulation and reduce the training cost required for one-shot stimulation, thereby enhancing its generality in industry. By combining human demonstrations with multi-modal learning techniques, this paper has demonstrated the feasibility and effectiveness of MLM in both simulated and real-world environments. With further advancements and refinements, MLM holds great promise for comprehensively enhancing the intelligence and generality of robots.

Author contributions. Diqing Yu conceived and designed the study, Xinggang Fan and Yuao Jin conducted data gathering. Yaonan Li and Han Li performed the experiment and statistical analyses. Diqing Yu, Xinggang Fan, Yaonan Li, and Heping Chen wrote the article.

Financial support. This research is supported by the Basic Research Program of Shenzhen (JCYJ20180504170303184 and JCYJ20190806172007629) and Guangdong Basic and Applied Basic Research Foundation (2021A1515011423).

Competing interest. The authors declare no conflicts of interest exist.

Ethical approval. Not applicable.

References

[1] A. Vaswani, N. Shazeer, N. Parmar, J. Uszkoreit, L. Jones, A. N. Gomez, Ł. Kaiser and I. Polosukhin, "Attention is all you need," *Adv. Neur. Inf. Process. Syst.* **30**, 5998–6008 (2017).

[2] D. M. Katz, M. J. Bommarito, S. Gao and P. Arredondo, "GPT-4 passes the bar exam," *Philos. Trans. R. Soc. A: Math. Phys. Eng. Sci.* **382**(2270), 20230254 (2024).

[3] E. Waisberg, J. Ong, M. Masalkhi, S. A. Kamran, N. Zaman, P. Sarker, A. G. Lee and A. Tavakkoli, "GPT-4: A new era of artificial intelligence in medicine," *Irish J. Med. Sci. (1971-)* **192**(6), 3197–3200 (2023).

[4] S. Wang, Z. Zhou, B. Li, Z. Li and Z. Kan, "Multi-modal interaction with transformers: Bridging robots and human with natural language," *Robotica* **42**(2), 415–434 (2024).

[5] D. Narayanan, M. Shoeybi, J. Casper, P. LeGresley, M. Patwary, V. Korthikanti, D. Vainbrand, P. Kashinkunti, J. Bernauer, B. Catanzaro, A. Phanishayee and M. Zaharia, "Efficient Large-scale Language Model Training on GPU Clusters Using Megatron-LM," **In:** *Proceedings of the International Conference for High-Performance Computing, Networking, Storage and Analysis* (2021) pp. 1–15.

[6] S. Yuan, H. Zhao, Z. Du, M. Ding, X. Liu, Y. Cen, X. Zou, Z. Yang and J. Tang, "Wudaocorpora: A super large-scale Chinese corpora for pre-training language models," *AI Open* **2**, 65–68 (2021).

[7] A. Chowdhery, S. Narang, J. Devlin, M. Bosma, G. Mishra, A. Roberts, P. Barham, H. W. Chung, C. Sutton, S. Gehrmann, P. Schuh, K. Shi, S. Tsvyashchenko, J. Maynez, A. Rao, P. Barnes, Y. Tay, N. Shazeer, V. Prabhakaran, E, Reif, N. Du, B. Hutchinson, R. Pope, J. Bradbury, J. Austin, M. Isard, G. Gur-Ari, P. Yin, T. Duke, A. Levskaya, S. Ghemawat, S. Dev, H. Michalewski, X. Garcia, V. Misra, K. Robinson, L. Fedus, D. Zhou, D. Ippolito, D. Luan, H. Lim, B. Zoph, A. Spiridonov, R. Sepassi, D. Dohan, S. Agrawal, M. Omernick, A. M. Dai, T. S. Pillai, M. Pellat, A. Lewkowycz, E. Moreira, R. Child, O. Polozov, K. Lee, Z. Zhou, X. Wang, B. Saeta, M. Diaz, O. Firat, M. Catasta, J. Wei, K. Meier-Hellstern, D. Eck, J. Dean, S. Petrov and N. Fiedel, "Palm: Scaling language modeling with pathways," *J. Mach. Learn. Res.* **24**(240), 1–113 (2023).

[8] S. Kodagoda, S. Sehestedt and G. Dissanayake, "Socially aware path planning for mobile robots," *Robotica* **34**(3), 513–526 (2016).

[9] Z. Feng, B. Xue, C. Wang and F. Zhou, "Safe and socially compliant robot navigation in crowds with fast-moving pedestrians via deep reinforcement learning," *Robotica* **42**(4), 1212–1230 (2024).

[10] E. Park and J. Lee, "I am a warm robot: The effects of temperature in physical human–robot interaction," *Robotica* **32**(1), 133–142 (2014).

[11] S. Kansal and S. Mukherjee, "Vision-based kinematic analysis of the Delta robot for object catching," *Robotica* **40**(6), 2010–2030 (2022).

[12] M. Zubair, S. Kansal and S. Mukherjee, "Vision-based pose estimation of craniocervical region: Experimental setup and saw bone-based study," *Robotica* **40**(6), 2031–2046 (2022).

[13] S. Jana, L. A. Tony, A. A. Bhise, V. P. V. and D. Ghose, "Interception of an aerial manoeuvring target using monocular vision," *Robotica* **40**(12), 4535–4554 (2022).

[14] Q. M. Marwan, S. C. Chua and L. C. Kwek, "Comprehensive review on reaching and grasping of objects in Robotics," *Robotica* **39**(10), 1849–1882 (2021).

[15] S. S. Gujran and M. M. Jung, "multi-modal prompts effectively elicit robot-initiated social touch interactions," **In:** *Companion Publication of the 25th International Conference on multi-modal Interaction* (2023) pp. 159–163.

[16] V. Lin, H. Yeh, H. Huang and N. Chen, "Enhancing EFL vocabulary learning with multi-modal cues supported by an educational robot and an IoT-based 3D book," *System* **104**, 102691 (2022).

[17] Y. Lee, Y. Tsai, W. Chiu and C. Lee, "Multi-Modal Prompting With Missing Modalities for Visual Recognition," **In:** *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition* (2023) pp. 14943–14952.

[18] H. Kaindl, J. Falb and C. Bogdan, "multi-modal Communication Involving Movements of a Robot," **In:** *CHI'08 Extended Abstracts on Human Factors in Computing Systems*. Association for Computing Machinery (2008) pp. 3213–3218.

[19] D. Danny, F. Xia, M. S. M. Sajjadi, C. Lynch, A. Chowdhery, B. Ichter, A. Wahid, J. Tompson, Q. Vuong, T. Yu, W. Huang, Y. Chebotar, P. Sermanet, D. Duckworth, S. Levine, V. Vanhoucke, K. Hausman, M. Toussaint, K. Greff, A. Zeng, I. Mordatch and P. Florence, "PaLM-E: An embodied multimodal language model," *Int. Conf. Mach. Learn*. PMLR, 8469–8488 (2023).

[20] C. Lynch, A. Wahid, J. Tompson, T. Ding, J. Betker, R. Baruch, T. Armstrong and P. Florence, "Interactive language: Talking to robots in real time," *IEEE Robot. Autom. Lett*. **8**, 1659–1666 (2023).

[21] Y. Duan, M. Andrychowicz, B. Stadie, J. Ho, J. Schneider, I. Sutskever, P. Abbeel and W. Zaremba, "One-shot imitation learning," *Adv. Neur. Inf. process. syst.* **30**, 1087–1098 (2017).

[22] A. Bonardi, S. James and A. J. Davison, "Learning one-shot imitation from humans without humans," *IEEE Robot. Autom. Lett.* **5**(2), 3533–3539 (2020).

[23] D. A. Huang, D. Xu, Y. Zhu, A. Garg, S. Savarese, L. Fei-Fei and J. C. Niebles, "Continuous Relaxation of Symbolic Planner for One-Shot Imitation Learning," **In:** *2019 IEEE/RSJ International Conference on Intelligent Robots and Systems (IROS)* (2019) pp. 2635–2642.

[24] Z. Mandi, F. Liu, K. Lee and P. Abbeel, "Towards More Generalizable One-Shot Visual Imitation Learning," **In:** *International Conference on Robotics and Automation (ICRA)* (2022) pp. 2434–2444.

[25] R. Rahmatizadeh, P. Abolghasemi, L. Bölöni and S. Levine. "Vision-Based Multi-Task Manipulation for Inexpensive Robots Using End-to-End Learning from Demonstration," **In:** *2018 IEEE International Conference on Robotics and Automation (ICRA)* (2018) pp. 3758–3765.

[26] M. G. Tamizi, H. Honari, A. Nozdryn-Plotnicki and H. Najjaran, "End-to-end deep learning-based framework for path planning and collision checking: Bin-picking application," *Robotica* **42**(4), 1094–1112 (2024).

[27] M. A. Post, "Probabilistic robotic logic programming with hybrid Boolean and Bayesian inference," *Robotica* **42**(1), 40–71 (2024).

[28] C. Finn, T. Yu, T. Zhang, P. Abbeel and S. Levine, "One-Shot Visual Imitation Learning Via Meta-Learning," **In:** *Conference on Robot Learning* (2017) pp. 357–368.

[29] Y. Jiang, A. Gupta and Z. Zhang, "Vima: General Robot Manipulation with multi-modal Prompts," **In:** NeurIPS. 2022 Foundation Models for Decision Making Workshop (2022).

[30] A. Nguyen, N. Nguyen, K. Tran, E. Tjiputra and Q. D. Tran, "Autonomous Navigation in Complex Environments with Deep Multi-Modal Fusion Network," **In:** *2020 IEEE/RSJ International Conference on Intelligent Robots and Systems (IROS)* (2020) pp. 5824–5830.

[31] K. Noda, H. Arie, Y. Suga and T. Ogata, "multimodal integration learning of robot behavior using deep neural networks," *Robot. Auton. Syst.* **62**(6), 721–736 (2014).

[32] T. Xue, W. Wang, J. Ma, W. Liu, Z. Pan and M. Han, "Progress and prospects of multi-modal fusion methods in physical human–robot interaction: A review," *IEEE Sens. J.* **20**(18), 10355–10370 (2020).

[33] H. Ravichandar, A. S. Polydoros, S. Chernova and S. Billard, "Recent advances in robot learning from demonstration," *Annu. Rev. Control Robot. Auton. Syst.* **3**(1), 297–330 (2020).

[34] M. N. Nicolescu and M. J. Mataric, "Natural Methods for Robot Task Learning: Instructive Demonstrations, Generalization and Practice," **In:** *Proceedings of the second international joint conference on Autonomous agents and multiagent systems* (2003) pp. 241–248.

[35] S. Dasari and A. Gupta, "Transformers for One-Shot Visual Imitation," **In:** *Conference on Robot Learning. PMLR* (2021) pp. 2071–2084.

[36] G. Bertasius, H. Wang and L. Torresani, "Is space-time attention all you need for video understanding?," *Int. Conf. Mach. Learn.* **2**(3), 4 (2021).

[37] K. Soomro, A. R. Zamir and M. Shah, "UCF101: A dataset of 101 human actions classes from videos in the wild (2012). arXiv preprint arXiv: 1212.0402.

[38] T. Lin, M. Maire, S. Belongie, J. Hays, P. Perona, D. Ramanan, P. Dollár and C. L. Zitnick, "Microsoft COCO: Common Objects in Context," **In:** *Computer Vision–ECCV 2014: 13th European Conference*, Zurich, Switzerland (2014) pp. 740–755. Proceedings, Part V 13.

[39] A. Dosovitskiy, L. Beyer, A. Kolesnikov, D. Weissenborn, X. Zhai, T. Unterthiner, M. Dehghani, M. Minderer, G. Heigold, S. Gelly, J. Uszkoreit and N. Houlsby. An image is worth 16x16 words: Transformers for image recognition at scale[J] (2020). arXiv preprint arXiv: 2010.11929.

[40] S. Carpin, M. Lewis, J. Wang, J. Balakirsky and C. Scrapper, "Bridging the Gap Between Simulation and Reality in Urban Search and Rescue," **In:** *RoboCup 2006: Robot Soccer World Cup X. 10* (2007) pp. 1–12.

[41] J. Hua, L. Zeng, G. Li and Z. Ju, "Learning for a robot: Deep reinforcement learning, imitation learning, transfer learning," *Sensor*, **21**(4), 1278 (2021).
[42] B. Singh, R. Kumar and V. P. Singh, "Reinforcement learning in robotic applications: A comprehensive survey," *Artif. Intell. Rev.* **55**(2), 945–990 (2022).
[43] F. M. Talaat and H. ZainEldin, "An improved fire detection approach based on YOLO-v8 for smart cities," *Neural Comput. Appl.* **35**(4), 20939–20954 (2023).

Cite this article: D. Yu, X. Fan, Y. Li, H. Chen, H. Li and Y. Jin, "A multi-modal learning method for pick-and-place task based on human demonstration", Robotica. https://doi.org/10.1017/S0263574724001322