The International Journal of Advanced Manufacturing Technology (2024) 134:5431–5447
https://doi.org/10.1007/s00170-024-14468-w

ORIGINAL ARTICLE

A robotic manipulation framework for industrial human–robot collaboration based on continual knowledge graph embedding

Bohan Feng[1] · Xinzhe Juan[1] · Xinyi Gao[1] · Qi Zhou[1] · Youyi Bi[1]

Received: 15 March 2024 / Accepted: 8 September 2024 / Published online: 20 September 2024

Abstract
Hybrid robots can assist human workers in various tasks due to their integration of mobility and manipulability. The rapid diffusion of these robots in factories has significantly elevated the automation and intelligence level of manufacturing, while also brings challenges to human–robot collaboration. Traditionally, human workers need to instruct robots to perform a range of tasks by explicitly demonstrating these operations. However, this process imposes excessive burdens on workers as the tasks and environment for robots become more and more diversified and complex. To alleviate this issue, we propose an innovative robotic manipulation framework based on continual knowledge graph embedding. This framework enables hybrid robots to break free from the constraints of fixed rules set by human demonstrations, instead endowing them with inferring capability. The core idea is to utilize semantic information related to objects (such as category, material, and components) and tasks assigned to infer appropriate operational parameters for robots via a knowledge graph. These operational parameters include the suitable type of gripper, the proper area for object manipulation, and the reasonable force range for effective grasping. We conduct an experimental analysis of the proposed framework with a real-world hybrid robot, which performed 158 different tasks involving 46 objects commonly seen in industry, achieving a success rate of up to 96.8%. Furthermore, our framework can continuously enhance the adaptability of robotic operations and effectively balance the learning of new and old knowledge. This research contributes to the development of advanced robotic manipulation method in the context of industrial human–robot collaboration.

Keywords Hybrid robots · Continual knowledge graph embedding · Human–robot collaboration · Industrial automation

1 Introduction

A hybrid robot in factories usually consists of a robotic arm and a mobile platform, which is often an Automated Guided Vehicle (AGV) (see Fig. 1). This integration of mobility and manipulability allows hybrid robots to flexibly replace or assist workers in a variety of operational tasks in industry, such as retrieving certain tools or inspecting the status of facilities as needed by workers. In addition, the robotic arm on a hybrid robot can be equipped with Auto Tools Change (ATC), which greatly extends the type of end-effectors that a single robot can use and broadens the range of manipulable objects. This advancement significantly improves human–robot collaboration efficiency and spurs the rapid spread of hybrid robots in smart factories [1].

The foundation for these hybrid robots to accomplish various operational tasks lies in stable and reliable robotic grasping. Current hybrid robots often learn to perform tasks by imitating human demonstrations, i.e., the workers manually show robots the operational tasks to provide routine grasping instructions. This approach lacks scalability, particularly in factory environments where there is an abundance of objects with intricate designs and delicate handling requirements. Additionally, due to variations in production, the form and

✉ Youyi Bi
youyi.bi@sjtu.edu.cn

Bohan Feng
bohan.feng@sjtu.edu.cn

Xinzhe Juan
1akman@sjtu.edu.cn

Xinyi Gao
yvonne4399@sjtu.edu.cn

Qi Zhou
zhouqi1998@sjtu.edu.cn

1 University of Michigan – Shanghai Jiao Tong University Joint Institute, Shanghai Jiao Tong University, 800 Dongchuan Road, Minhang Dist., Shanghai, China

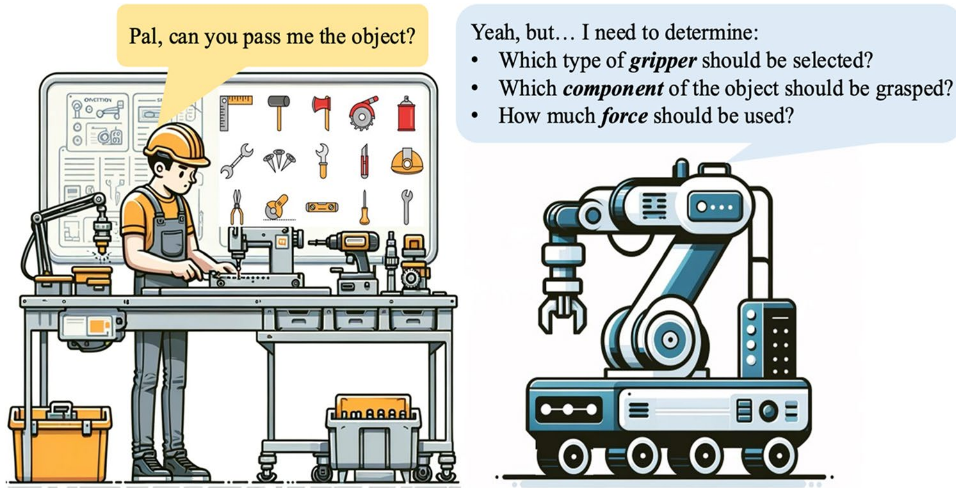

Fig. 1 A typical scenario of human–robot collaboration in industrial environment

materials of operation objects for robots frequently change, necessitating appropriate operational guidance under different grasping tasks. This complexity imposes significant burden on human workers.

To enable robots to grasp objects stably and diminish the need for manual instruction, a variety of grasping strategies have been developed. These include the utilization of dual-arm robots with heterogeneous grippers [2], grasping based on tactile sensing (leveraging tactile feedback for robust grasping under uncertainty) [3], grasping based on visual and language instructions [4], and learning grasp poses from synthetic training examples [5]. Moreover, recent advances in control systems have enhanced the robots' ability to adapt their grasping strategy to the characteristics of the object at hand [6–8]. Current grasping models primarily focus on the physical and geometric properties of objects, proposing feasible areas for robotic grasping without considering the subsequent tasks to complete. However, the selection of grasping area should also be contingent upon these subsequent tasks. Moreover, robots should be capable of inferring the relationship between a given task and the operational object's constituent components, thereby determining the proper grasping area based on the characteristics of the task. This aspect is particularly crucial in human–robot collaboration, as the correct choice of grasping area can ensure efficiency and safety in robot's subsequent operations. For instance, in precision assembly or handling fragile items (e.g., micro sensors, delicate glass containers), the choice of grasp points directly influences the success of the task and the operational safety.

In fact, more intelligent grasping can be achieved by fully utilizing the semantic knowledge extracted from tasks and environment. The statistical correlations or heuristic features learned from historical grasping tasks can effectively support the generalization of grasping methods in new environments [9]. For example, recent research in semantic grasping suggests using semantic segmentation to predict an object's affordances [10], leveraging data and semantic knowledge for task-oriented grasping areas [11], or employing probabilistic logical frameworks to identify the most likely graspable components of an object [12]. Yet, these methods primarily focus on the object's grasping area without considering factors such as the appropriate type of gripper or the optimal force for grasping, which are crucial for many complex robotic operations. Kwak et al. [13] introduced a knowledge graph-based semantic grasping approach, capable of reasoning about the proper grasping area, end effector and grasping force. However, this approach mainly targeted at the grasping of relatively simple objects, such as various types of bottles. It also required extensive annotated data for training, making this approach less applicable for rapid deployment in complex factory settings.

To address these challenges, we propose a robotic manipulation framework for human–robot collaboration in industrial scenarios based on continual knowledge graph embedding. Our research aims to enhance the scalability of hybrid robot operations, fundamentally enabling these robots to become more versatile and adapt to a diverse array of tasks and operational objects. This is achieved by effectively utilizing semantic information regarding the operational objects (such as their categories, materials, and components) and the tasks assigned by workers to infer suitable robot manipulation parameters. Unlike previous research, these parameters include the appropriate type of gripper, the optimal area for object manipulation, and the reasonable force for effective operation, as illustrated in Fig. 1. The continual knowledge graph embedding approach can effectively balance between acquiring old and new knowledge, preventing the robot from forgetting old knowledge while learning new information. To better understand the ambiguous instructions from human workers, our framework employs large pre-trained models to recognize the class, components, and materials of the objects to manipulate, which can rapidly adapt to new environments through fine-tuning without the need of extensive annotated data. It significantly reduces the time and cost associated with data collection and processing. Our framework enhances the intelligent interaction between human workers and hybrid robots, providing an efficient pathway for the practical implementation and swift deployment of human–robot collaboration system in smart factories.

The rest of the paper is structured as follows. Section 2 reviews related work in robotic grasping, knowledge graph embedding, and continual learning. Section 3 introduces our proposed approach and key techniques involved. Section 4 presents the effectiveness of our approach through experiments on measuring inference success, evaluating continual learning, testing real robot performance, and assessing recognition effect. Section 5 discusses the key findings and provides detailed analysis. Section 6 summarizes the work and suggests future research directions.

2 Literature review

2.1 Robotic grasping

Reliable robotic grasping strategies are pivotal for the successful manipulation of various objects. These strategies involve not only the assessment of proper grasping poses (e.g., determining grasping angle and grasping area), but also the selection of suitable grasping tool and reasonable grasping force adaptive to different task scenarios and objectives. Previous studies employing dense supervision strategies [14] or using sampling-evaluation-based methods [15] have achieved high success rates in grasping generic objects by determining gripper poses. Similar research includes dual-arm robot grasping with heterogeneous grippers [2], tactile feedback-based grasping [3], vision and instruction-based grasping [4], and learning grasp poses from synthetic training examples [5]. However, these studies mainly focus on the physical and geometric characteristics of objects to identify feasible grasping poses. These models, being task-unrelated, may not adapt well to complex and variable industrial scenarios, where even the same object requires different manipulation strategies within varying tasks. For instance, in the task of packaging electronic parts, the focus of the manipulation task is operational efficiency and safety. Robots may choose suction cups rather than conventional metal grippers to ensure rapid handling without compromising the safety of these parts. While in the task of assembling electronic parts that involve high-precision operations such as insertion, the manipulation focuses on the operational accuracy. Robots may employ mechanical grippers with fine control capability to take accurate picking, transporting, and placing actions.

Recent research has focused on enabling robots to make reasonable decisions, flexibly respond to diverse tasks and objects by incorporating semantic information. For example, CAGE [16] combines a neural network based on the Wide & Deep model [17] with context-aware semantic representation, achieving a balance in understanding and generalizing context for semantic grasping. Duan et al. [18] used a multi-task semantic grasping convolutional neural network to understand the relationship between objects and grasping in various scenarios, utilizing multimodal information to choose the best grasping area of objects.

In the realm of semantic grasping, the combination of knowledge graphs with grasping strategies has recently attracted significant attention. Murali et al. [11] introduced the GCNGrasp framework, utilizing semantic knowledge encoded in knowledge graphs to extend the TaskGrasp dataset for objects and tasks. Kwak et al. [13] developed roboKG, a hierarchical graph embedding system representing household items in terms of labels, components, and materials, which can predict the grasp parameters of grippers. These methods provide an inspiration for constructing knowledge graphs for robotic operations. However, they primarily focus on home-application scenarios and lack the capability of continual learning, and they also require extensive annotated data for model training. In this study, we expect to address these issues in semantic grasping for robots, enabling the generation of appropriate grasping decisions for different tasks and operational objects.

2.2 Knowledge graph embedding

Knowledge graph embedding (KGE) aims to map each entity and its relations in a knowledge graph into a low-dimensional vector space as embeddings, which can be used to predict missing links between entities and facilitate knowledge reasoning. This technology has been applied to enhance the performance of certain information query tasks such as recommendation systems [19]. For each knowledge triple (h, r, t), a distance-based scoring function $f(h, r, t)$ is used to model the plausibility of the triple. This function measures the distance between the entity embeddings of h and t, specific to the relation r.

A notable KGE method is TransE [20], which embeds entities and relations in a shared vector space of dimension d. Its loss function is defined as $\|h + r - t\|$, aiming to bring h closer to t after translation by r. This can be extended with different geometric transformations, such as TransH [21], which projects the entity embeddings of h and t onto a relation-specific hyperplane, or RotateE [22], which defines the relation as a rotation in the complex vector space from entity h to t. Therefore, their embeddings are expected to satisfy $h \odot r \approx t$, where $\odot$ denotes element-wise multiplication. DistMult [23] proposed constraining the relation matrix to a diagonal matrix, significantly reducing the number of parameters in the bilinear model. HAKE [24] maps entities into the polar coordinate system, with radial coordinates representing entities at different semantic levels and angular coordinates distinguishing entities at the same level. These structured KGE methods provide direct data mapping, suitable for handling well-defined entities and relations, effectively supporting the decision-making process in industrial robot operations.

Recently, description-based KGE methods are emerging [25–27]. Their essence lies in the integration of Large Language Models (LLMs), which can facilitate KGE methods in encoding or generating facts from textual information. However, description-based methods rely on large volumes of high-quality consistent textual data, and complex text processing techniques. In this study, we expect to leverage the capabilities of structured KGE methods to address complex challenges in knowledge representation and reasoning.

2.3 Continual learning for knowledge graph embedding

Many previous knowledge graphs, especially those used for industrial robot grasping, undergo incremental updates as the operational objects in the industrial environment frequently change. Static embedding models usually necessitate retraining the entire knowledge graph after each update [28]. This process is further complicated by data incompleteness in industrial scenarios [29] due to privacy protection, data storage constraints, or limited data acquisition windows. Utilizing all training data to update the entire knowledge graph is time-consuming and sometimes impractical, thereby highlighting the necessity of continual learning methods.

Continual learning, a subset of lifelong machine learning, aims to acquire knowledge of new domains, categories, or tasks without erasing previous learning experiences [30]. Various continual learning methods have been proposed by previous researchers. For example, regularization allows for adjusting shared weights that perform well across both past and recent sessions by enforcing some regularization terms in new learning sessions [31, 32]. Modifying network architectures [33, 34], as well as generative replay, such as deep generative replay, enable learning the distribution of training data from earlier learning sessions [35]. The crux of continual learning of graph embeddings lies in applying the principles of continual learning to the structure of knowledge graph, allowing the system to maintain flexibility and continuity of knowledge in a constantly changing environment. Knowledge graphs provide a rich information network through structured relations of entities, but when confronting dynamic environments, this network necessitates continuous updates and adjustments [36]. As robots increasingly encounter new environments and tasks, continual learning can effectively balance the learning of new and old knowledge, ensuring that robots do not forget old knowledge while learning new information [37]. In this study, we propose to construct a continual knowledge graph embedding approach suitable for robotic manipulations in industrial settings. Compared to previous work, we aim to update the knowledge graph embedding network under the complete knowledge graph of continuously updated manipulation demonstrations, significantly enhancing the transferability of the task planning for robotic manipulations.

3 Method

3.1 Robotic manipulation framework for industrial human–robot collaboration

Figure 2 illustrates the proposed comprehensive robotic manipulation framework to enhance industrial human–robot collaboration. The framework comprises four key processes: Instruction, Recognition, Inference, and Motion Planning. Each process plays a crucial role in achieving seamless and efficient collaboration between human workers and hybrid robots within an industrial setting.

The Instruction process interprets and fulfills factory workers' requirements for assistance in object-related manipulation tasks. The hybrid robot, equipped with ATC (Auto Tools Change), demonstrates versatility by swiftly switching between various end-effectors (e.g., parallel jaw gripper, soft gripper, suction cup, electric screwdriver, etc.) to adapt to the task at hand. As depicted in Fig. 2, a typical process involves a worker indicating the need to grasp an object situated on a table. Upon receiving this instruction, the hybrid robot initiates the recognition of the grasping object.

The Recognition process is pivotal for understanding the physical object through the perception ability of robots. It involves the identification of object class, semantic segmentation, and material detection. Initially, the robot utilizes its arm-mounted camera to scan the object and acquire its 3D point cloud data (as demonstrated with a box cutter in Fig. 2). This data is then rendered into multiple views to facilitate comprehensive analysis of the object. By feeding the images of these views into the CLIP model [38], a multimodal pre-trained model that combines vision and language comprehension, our system can accurately recognize the object's class. After that, the PartSLIP model [39], adept at zero-shot/few-shot semantic segmentation processes the object's different views to accomplish semantic segmentation at the instance level. The segmentation outcomes, coupled with material detection through the DEP model [40],

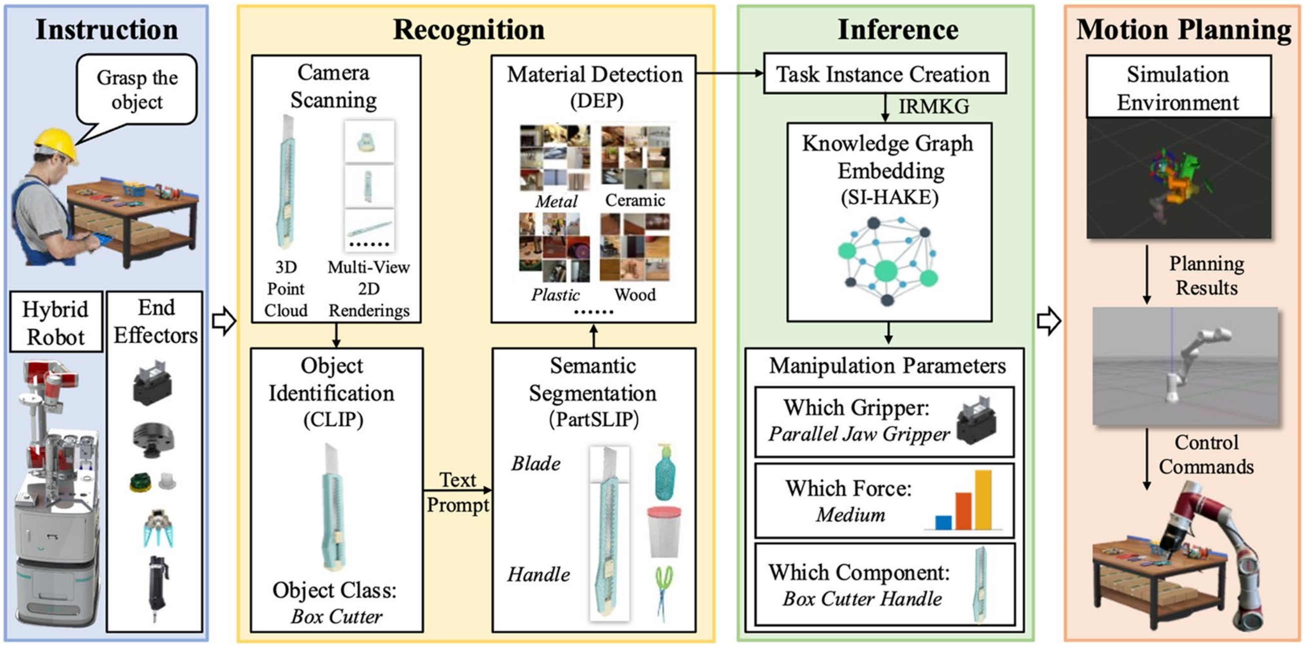

Fig. 2 The proposed robotic manipulation framework for industrial human–robot collaboration. The grasping of a box cutter is used as an illustrative example

enrich our framework's understanding of the object's components with their respective materials.

The Inference process first utilizes the information acquired from the Recognition process to formulate a task instance in the form of knowledge graph triples in accordance with the developed Industrial Robot Manipulation Knowledge Graph (IRMKG). After that, the Synaptic Intelligence—Hierarchy Aware Knowledge Graph Embedding (SI-HAKE) method is designed to infer the optimal manipulation parameters of the task instance considering the object's physical characteristics and the task requirements. These parameters include the choice of gripper, the force magnitude, and the specific component of the object to be grasped. The IRMKG and the SI-HAKE method will be detailed in Sections 3.2 and 3.3.

The Motion Planning process then deals with the generated manipulation parameters. Simulation environments are utilized to generate the motion planning results and refine the control commands. These commands are finally executed by the physical hybrid robot, translating the virtual motions into tangible actions.

By integrating advanced object recognition, knowledge graph-based manipulation inference and efficient motion planning, the proposed robotic manipulation framework sets a new practice for human–robot collaboration. The subsequent Sections 3.2 and 3.3 will provide a detailed explanation of the framework's core mechanisms: the construction of the knowledge graph (IRMKG) and the graph inference method (SI-HAKE).

3.2 IRMKG: industrial robot manipulation knowledge graph

In industrial environments, the completion of specific tasks assigned by workers, such as retrieving a tool from an adjacent table, is a sophisticated decision-making process for hybrid robots. This process involves determining the optimal grasp point, selecting the appropriate end-effector, and applying the correct force to ensure the successful completion of the task. To address this challenge, we develop the Industrial Robot Manipulation Knowledge Graph (IRMKG), a knowledge graph representing the expert-defined knowledge of robotic manipulation required for various tasks on different objects in industrial scenarios. Drawing the inspirations from the construction logic of GCNGrasp [11] and roboKG [13], we redefine and enrich their relational schema in the design of IRMKG. The IRMKG is constructed by ten distinct entity types, ten unique relationships, and ten specific triple configurations as detailed in Tables 1 and 2. These tables provide examples of each entity and relationship with the number of these examples.

We introduce an entity type named Object_Class to represent objects in our application context (e.g., Saw, Plier, Screwdriver, Screw, Container). For each object, the number of Object_Instance entities corresponds to the combination of the object's material and components. The relationship Instantiate_Object connects Object_Class and Object_Instance, as in the triple of *(Saw, Instantiate_Object, Cross_Saw)*. We represent materials with Material_Class. Unlike

Table 1 Entity types in IRMKG

Entity type	Examples	Num. of examples
Object_Class	Saw, Piler, Screwdriver···	96
Object_Instance	Cross_Saw, Forest_Saw, Piler_1···	170
Component_Category	Body, Head, Blade, Handle, Lid	5
Component_Class	Saw_Blade, Saw_Body, Saw_Handle, Screwdriver_Handle···	142
Material_Class	Ceramic, Fabric, Food, Glass, Leather, Metal, Paper, Plastic, Rubber, Stone, Wood	11
Task_Category	Grasp, Lift, Open, Pour, Push, Pull, Rotate, Squeeze, Screw	9
Task_Class	Grasp_Saw, Lift_Saw, Push_Saw, Pull_Saw···	391
Task_Instance	Grasp_Cross_Saw, Lift_Cross_Saw···	624
Gripper_Type	Parallel_Jaw_Gripper, Soft_Gripper, Hard_Suction_Cup, Soft_Suction_Cup, Electric_Screwdriver	5
Grasping_Force	Weak, Medium, Strong	3

Table 2 Relationship types in IRMKG

Relation	Triple type	Num. of examples
Instantiate_Object	(Object_Class, Instantiate_Object, Object_Instance)	170
Specify_Component	(Component_Category, Specify_Component, Component_Class)	142
Is_Component_Of	(Component_Class, Is_Component_Of, Object_Instance)	236
Make	(Material_Class, Make, Component_Class)	142
Specify_Task	(Task_Category, Specify_Task, Task_Class)	391
Instantiate_Task	(Task_Class, Instantiate_Task, Task_Instance)	624
Include	(Task_Instance, Include, Object_Instance)	624
Which_Gripper	(Task_Instance, Which_Gripper, Gripper_Type)	624
Which_Force	(Task_Instance, Which_Force, Grasping_Force)	624
Which_Component	(Task_Instance, Which_Component, Component_Class)	624

roboKG, we associate materials to specific components of objects, as materials can vary across different components of an object in industrial scenarios. For example, a saw's handle can be made of plastic *(Plastic, Make, Saw_Handle)*, while the blade is made of metal *(Metal, Make, Saw_Blade)*. We consider 11 different materials: ceramic, fabric, food, glass, leather, metal, paper, plastic, rubber, stone, and wood. To connect Component_Category and Component_Class, we establish the relationship Specify_Component and triples like *(Blade, Specify_Component, Saw_Blade)*. To represent which component is part of which object instance, we devise the relationship Is_Component_Of and triples like *(Saw_Blade, Is_Component_Of, Cross_Saw)*.

We consider nine different operational tasks: Grasp, Lift, Open, Pour, Push, Pull, Rotate, Squeeze, and Screw, represented by Task_Category. To distinguish each task based on the target object, we use Task_Class to indicate which task is performed on which specific object, such as Grasp_Saw, Open_Container. To narrow Task_Category down to Task_Class, we define the relationship Specify_Task and triples like *(Grasp, Specify_Task, Grasp_Saw)*. We utilize Task_Instance to map each task to a specific object, such as Grasp_Cross_Saw. To narrow Task_Class to Task_Instance, we define the relationship Instantiate_Task and triples like *(Grasp_Saw, Instantiate_Task, Grasp_Cross_Saw)*. Moreover, each Task_Instance should be connected to Object_Instance to indicate a task includes which object. To represent this, we create the relationship Include and triples like *(Grasp_Cross_Saw, Include, Cross_Saw)*.

Finally, we define three special relationships directly related to robotic manipulation: Which_Gripper, Which_Force, and Which_Component. Given a specific object and task, we indicate which gripper the robot should use, the force with which the robot should grasp the object, and which component of the object the robot should grasp. IRMKG incorporates five different types of grippers: parallel jaw gripper, soft gripper, hard suction cup, soft suction cup, and electric screwdriver. Additionally, we categorize the reasonable magnitude of grasping force into three different levels: weak, medium, and strong. For each Task_Instance, we specify an ideal type of gripper, grasping force, and grasping component. For example, when grasping a cross saw, we construct triples like *(Grasp_Cross_Saw, Which_Gripper, Parallel_Jaw_Gripper)*, *(Grasp_Cross_Saw, Which_Force, Medium)*, and *(Grasp_Cross_Saw, Which_Component, Saw_Handle)* to indicate the manipulation parameters of this task.

3.3 Continual knowledge graph embedding: Synaptic Intelligence—Hierarchy Aware Knowledge Graph Embedding (SI-HAKE)

In this subsection, the Synaptic Intelligence—Hierarchy Aware Knowledge Graph Embedding (SI-HAKE), a continual knowledge graph embedding (CKGE) method designed for dynamic graph inference within the IRMKG, is explained. Before digging into the details of SI-HAKE, the necessary technical background is provided.

Knowledge Graph Embedding (KGE) represents knowledge graph $\mathcal{G}$ in vector space, learning a continuous vector representation from a dataset of triples $\mathcal{D} = \{(h, r, t)_i, y_i \mid h_i, t_i \in \mathcal{E}, r_i \in \mathcal{R}, y_i \in \{0,1\}\}$, with $i \in \{1 \ldots |\mathcal{D}|\}$. Here y_i denotes whether relation $r_i \in \mathcal{R}$ holds between entities $h_i, t_i \in \mathcal{E}$. Each entity is encoded as a vector $\mathbf{e} \in \mathbb{R}^{d_{\mathcal{E}}}$, and each relation is encoded as a mapping between vectors $\mathbf{r} \in \mathbb{R}^{d_{\mathcal{R}}}$, where $d_{\mathcal{E}}$ and $d_{\mathcal{R}}$ are the dimensions of vectors and mappings respectively. The embeddings for $\mathcal{E}$ and $\mathcal{R}$ are learned through a scoring function $f(h, r, t)$ that assigns higher values to triples that correctly depict true facts, known as positive triples.

The goal of CKGE is to extend KGE framework to dynamically update and refine the embeddings as new information is integrated. This ensures that the knowledge graph can evolve over time, addressing the challenge of new data's arrival. CKGE involves splitting the dataset of triples $\mathcal{D}$ into multiple subsets $\mathcal{D}^n$ for different learning sessions, where n denotes the session index. Each subset $\mathcal{D}^n$ contains a distinct set of triples, corresponding to a specific set of entities and relationships, which grows as new observations are made (i.e., $|\mathcal{E}^n| \leq \left|\mathcal{E}^{n+1}\right|$).

In our research, a hierarchical structure exists between entities, such as Component_Category → Component_Class and Task_Category → Task_Class. To represent entities and relations in a continuous feature space while maintaining the structure of the knowledge graph, we propose the Synaptic Intelligence—Hierarchy Aware Knowledge Graph Embedding (SI-HAKE) method. This continual knowledge graph embedding method maps entity embeddings to a polar coordinate system, distinguishing entities at different and same levels of semantic hierarchy. The SI-HAKE comprises two parts: the modulus part is used to differentiate entities between different levels of semantic hierarchy, and the phase part is used to distinguish entities within the same level.

To distinguish embeddings in the different parts, we use $\mathbf{e}_{\mathrm{m}}$ ($\mathbf{e}$ can be $\mathbf{h}$ or $\mathbf{t}$) and $\mathbf{r}_{\mathrm{m}}$ to denote the entity embedding and relation embedding in the modulus part, and use $\mathbf{e}_{\mathrm{p}}$ and $\mathbf{r}_{\mathrm{p}}$ to denote the entity embedding and relation embedding in the phase part.

The formulation of modulus part is given by:

$$\mathbf{h_m} \circ \mathbf{r_m} = \mathbf{t_m} \tag{1}$$

where $\circ$ is the Hadamard product, $\mathbf{h_m}, \mathbf{t_m} \in \mathbb{R}^{d_{\mathcal{E}}}$, $\mathbf{r_m} \in \mathbb{R}_+^{d_{\mathcal{R}}}$, allowing recognition of the existence of a relationship between two entities through signs. The corresponding distance function is defined as:

$$d_{r,m}(\mathbf{h_m}, \mathbf{t_m}) = \| \mathbf{h_m} \circ \mathbf{r_m} - \mathbf{t_m} \|_2 \tag{2}$$

Similarly, the formulation of phase part is given by:

$$\left(\mathbf{h_p} + \mathbf{r_p}\right) \mathrm{mod}\ 2\pi = \mathbf{t_p} \tag{3}$$

where $\mathbf{h_p}, \mathbf{t_p}, \mathbf{r_p} \in [0{,}2\pi]^{d_{\mathcal{E}}}$. Since in the polar coordinate system, phases possess periodic characteristics, therefore, the distance function is defined in sin form:

$$d_{r,p}(\mathbf{h_p}, \mathbf{t_p}) = \| \sin(\mathbf{h_p} + \mathbf{r_p} - \mathbf{t_p})/2 \|_1 \tag{4}$$

where sin is an operation that applies the sine function to each element of the input. Combining the modulus parts $d_{r,m}\left(\mathbf{h_m}, \mathbf{t_m}\right)$ and phase part $d_{r,p}\left(\mathbf{h_p}, \mathbf{t_p}\right)$, the distance funciton is defined as:

$$d_r(\mathbf{h}, \mathbf{t}) = d_{r,m}\left(\mathbf{h_m}, \mathbf{t_m}\right) + \gamma d_{r,p}\left(\mathbf{h_p}, \mathbf{t_p}\right) \tag{5}$$

where $\gamma \in \mathbb{R}$ is a parameter learned by the method. For an entity $\mathbf{h}$, SI-HAKE maps it to $\left[\mathbf{h_m}; \mathbf{h_p}\right]$ where $[\cdot;\cdot]$ denotes the concatenation of two vectors. The scoring function $f_r(\mathbf{h}, \mathbf{t})$ for evaluating the likelihood of relationships between pairs of entities is defined as:

$$f_r(\mathbf{h}, \mathbf{t}) = -d_r(\mathbf{h}, \mathbf{t}) = -\| \mathbf{h_m} \circ \mathbf{r_m} - \mathbf{t_m} \|_2 - \gamma \| \frac{\sin\left(\mathbf{h_p} + \mathbf{r_p} - \mathbf{t_p}\right)}{2} \|_1 \tag{6}$$

However, assuming all entities and relations are known prior to training is impractical. Considering the increased learning sessions in continual learning, we use synaptic intelligence [31] to consider the weight-specific contributions to the reduction in loss over a learning session. It encourages trained weights to not deviate from their previous values.

Based on this scoring function, the loss function is defined by self-adversarial training plus the synaptic intelligence part as:

$$L_{\mathcal{D}^n} = -log\sigma(\alpha - d_r(\mathrm{h}, \mathrm{t})) - \sum_{i=1}^{|\mathcal{D}^n|} p(h'_i, r, t'_i) log\sigma(d_r(\mathrm{h}'_\mathrm{i}, \mathrm{t}'_\mathrm{i}) - \alpha) \tag{7}$$

$$L = L_{\mathcal{D}^n} + \lambda \cdot (\| \Omega_e\left(\mathbf{e}^n - \mathbf{e}^{n-1}\right) \|_2^2 + \| \Omega_r\left(\mathbf{r}^n - \mathbf{r}^{n-1}\right) \|_2^2) \tag{8}$$

where α is a fixed margin, σ is a sigmoid function, and $\left(h_i', r, t_i'\right)$ is the i-th negative triple. λ is a regularization scaling term tuned as a hyper-parameter, and Ω is the parameter regularization strength.

Equation (7) is designed to discern between positive triples (true facts) and generated negative triples (false facts), enhancing the method's precision in representing graph relationships. Equation (8) introduces the synaptic intelligence part ($\| \Omega_e\left(\mathbf{e}^n - \mathbf{e}^{n-1}\right) \|_2^2 + \| \Omega_r\left(\mathbf{r}^n - \mathbf{r}^{n-1}\right) \|_2^2$) by incorporating a regularization scaling term (λ). This term penalizes abrupt changes in the embeddings of entities and relations across successive learning sessions. By ensuring gradual evolution of the method's embeddings, it facilitates the seamless integration of new information alongside the preservation of existing knowledge, a cornerstone for successful continual learning.

In the SI-HAKE method, entities' hierarchical relationships and category information are distinctively represented through the modulus and phase parts in a polar coordinate system. Figure 3 visualizes the embeddings of several entity pairs obtained from the SI-HAKE. Each point on this plot represents a mapping of the entity embedding into the 2D space. The mapping process works as follows. Each entity is represented by an embedding with two parts: modulus and phase components. The modulus component undergoes a transformation where the logarithm of its absolute value is multiplied by its sign, enhancing the contrast in magnitudes. The phase component is normalized within the range of $[-\pi, \pi]$ by dividing it by the embedding range and then scaling by π. The transformed modulus and phase components are then used to compute the Cartesian coordinates (x, y) for both the head and tail entities, converting polar coordinates (magnitude and phase) into a 2D space representation. The head and tail entities are plotted in this space, with each entity's position reflecting its semantic hierarchy. From Fig. 3, we can see that the resulting scatter plot displays the entities as points in a 2D space. The distance between the points and the circle center reflects the moduli of the entities at various semantic levels. The clear concentric circles indicate that our method can effectively capture and represent semantic hierarchies among entities. By leveraging semantic hierarchical reasoning for link prediction, the SI-HAKE method facilitates the automatic inference of relevant parameters necessary to complete specific manipulation tasks.

Figure 4 demonstrates an illustrative example of graph inference using SI-HAKE. In this example, the goal is to predict the appropriate manipulation parameters for Grasp_Box_Cutter_1. Note that both Box_Cutter_1 and Scissor_1 possess handle and blade, with the handle being plastic and the blade metallic. Owing to these shared properties, the SI-HAKE method attempts to place the embedding vectors of Box_Cutter_1 and Scissor_1 close together. Furthermore, since Grasp_Box_Cutter_1 and Grasp_Scissor_1 share the Grasp in their entity hierarchy, it can be inferred that the robot could grasp the Box_Cutter in the same manner as grasping the Scissor. Thus, we can use the learned embedding vectors to predict the appropriate gripper type, grasping force, and grasping component for a specific grasping task.

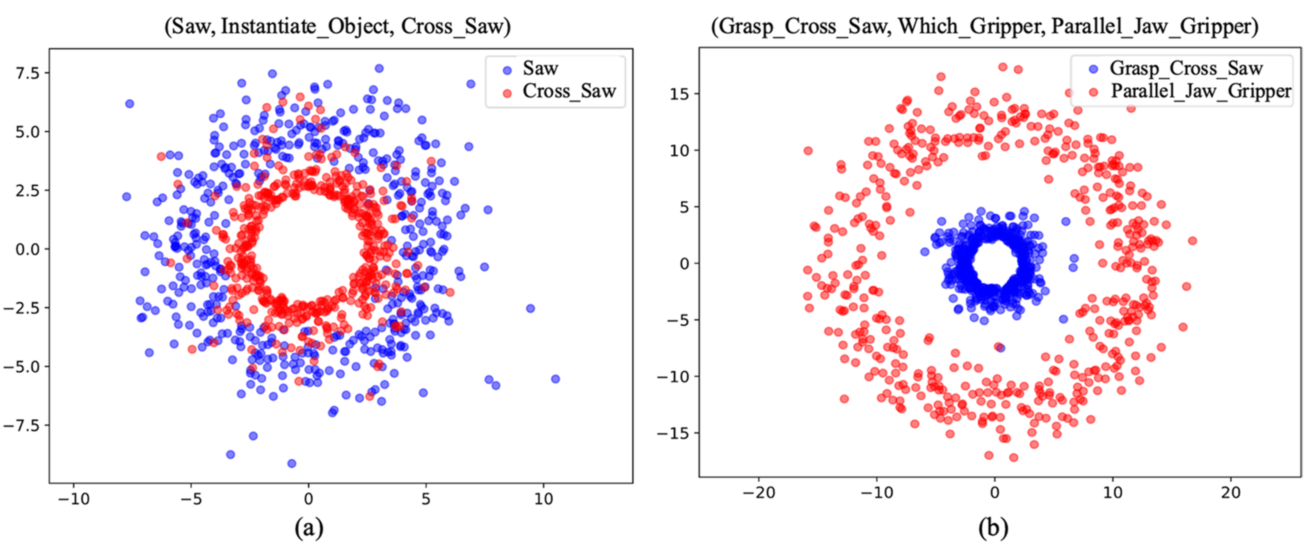

Fig. 3 Visualization of the embeddings of several entity pairs from SI-HAKE. **a** *(Saw, Instantiate_Object, Cross_Saw)*, **b** *(Grasp_Cross_Saw, Which_Gripper, Parallel_Jaw_Gripper)*

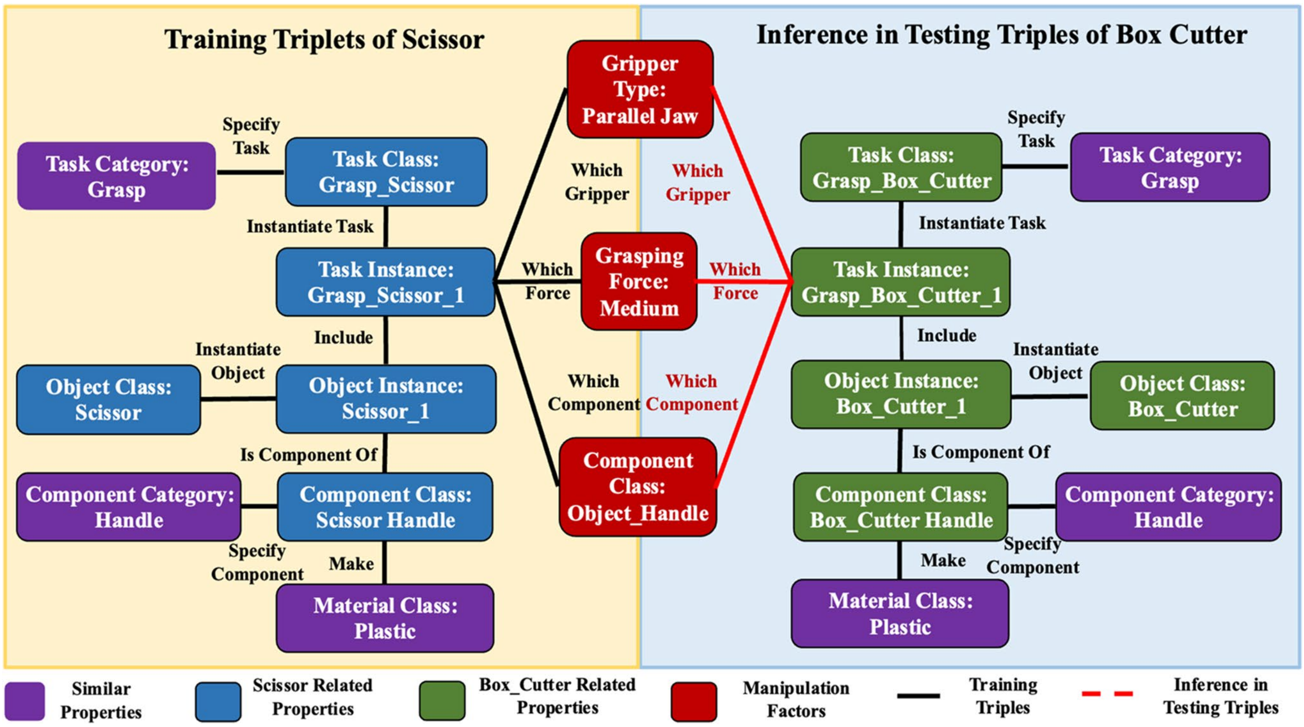

Fig. 4 An illustrative example of graph inference using the SI-HAKE method

4 Experiments and results

Four different experiments were conducted to validate the proposed robotic manipulation framework, especially the effectiveness of the developed IRMKG and the SI-HAKE method with a real hybrid robotic system. We first measured the inference success rate of the knowledge graph embedding method (SI-HAKE) to verify the appropriate construction of IRMKG and the validity of SI-HAKE in predicting missing entities during testing. Then, we evaluated the effectiveness of our method in the context of continual learning. After that, we tested the real robot based on the predictions obtained from IRMKG and SI-HAKE to check the success rate of robotic manipulation tasks. Lastly, we assessed the success rate of the Recognition process proposed in our manipulation framework where objects, components, and materials are automatically identified.

We built a hybrid robot system with a Jaka Zu7 multi-joint robotic arm combined with an AGV platform. To switch between different end-effectors for the robot, we installed Auto Tools Change (ATC) plates on both the robotic arm and end-effectors. Available end-effectors included parallel jaw gripper, soft gripper, hard suction cup, soft suction cup, and electric screwdriver. The robot can manipulate target objects based on predictions for each Task_Instance of Gripper_Type, Component_Class, and Grasping_Force. For each component of the target object, its precise position and posture were obtained by the camera in the robot system. Grasping _Force was controlled at three different levels: weak, medium, and strong. Corresponding to these three different levels of grasping force, we set the gripping force for the parallel jaw gripper at 15 N, 25 N, and 35 N, the surface pressure for pneumatic soft grippers and suction cups at 1.0 MPa, 1.5 MPa, and 2.0 MPa, and the torque for the screwdriver at 3 Nm, 5 Nm, and 8 Nm, respectively. The triples detailed in Section 3.2 served as the foundation for the training dataset. The hyperparameters of our method were as follows: $d_{\mathcal{E}}, d_{\mathcal{R}} = 1000, l_r = 0.0001, \alpha = 8, \lambda, \Omega_e, \Omega_r = 0.9$. For convenience in comparison, we refer to our knowledge graph-related methods (i.e., knowledge graph construction and embedding) as IRMKG in the following paragraphs.

4.1 Knowledge graph embedding method for predicting manipulation parameters

Given specific objects and tasks, we first tested whether the developed knowledge graph embedding method can accurately predict the appropriate gripper type, grasping force, and grasping component without including the continual learning method (we tested the effect of the continual learning part in Section 4.2). The entire training set was utilized in this test. We compared our method with five baseline methods: Random Prediction (RD), Distribution-Based Random Prediction (DRD), Training Set Memory (TM), Naive Bayes Classifier (NBC) [41], and Markov Logic Network (MLN) [42]. A detailed explanation of these compared methods is given below:

(1) Random Prediction (RD): Randomly predict a gripper type, grasping force, and grasping component.
(2) Distribution-Based Random Prediction (DRD): Consider data distribution in the training set to predict a suitable gripper type, grasping force, and grasping component. For instance, if 65% of tasks in the training set use the parallel jaw gripper, the probability of selecting this gripper is set to 0.65.
(3) Training Set Memory (TM): Use the training triples from the training set without KGE (i.e., lacking inference for test triples) to answer queries. As there are no repeated triples in our training set, we relied on the main grasping method of the object class in the Task_Instance in training set for grasping.
(4) Naive Bayes Classifier (NBC): Choose the most likely option for a given object and task characteristics by calculating and comparing the conditional probabilities of each gripper type, grasping force, and component. We employed the smoothing probability estimation technique m-estimate [43] to avoid overly extreme probabilities.
(5) Markov Logic Network (MLN): Answer queries using the pracmln tool [44] to learn statistical correlations related to objects and object properties in the dataset.

The tenfold cross-validation method was utilized to ensure a comprehensive evaluation, randomly dividing the dataset into ten subsets to represent the entire dataset. As illustrated in Table 3, IRMKG consistently outperformed the other five baseline methods. Notably, it achieved success rates of 0.957, 0.981, and 0.949 in predicting the correct gripper type, force range, and grasping component, respectively. This high level of success rate highlights the method's robustness and precision in understanding and analyzing the complex requirements of robotic operations. In contrast, the baseline methods, particularly RD and DRD, exhibited significantly lower success rate. RD's random approach underscored the task's complexity, which cannot be addressed through mere guesswork. DRD, while slightly better, still failed to capture the nuanced decision-making process in determining proper robotic manipulation parameters.

The IRMKG also achieved the highest success rate of simultaneously predicting all three parameters (i.e., gripper type, grasping force, and component) at 0.891. This underlines the method's ability to handle multifaceted decisions in robotic operations. These findings indicate that our method significantly outperforms traditional methods in predicting the necessary parameters for robotic manipulation tasks.

4.2 Continual learning capability

In this subsection, we tested the continual learning capability of our approach. We generated test triples corresponding to 30% of the training dataset's size, ensuring these test triples were produced independently and not extracted from the existing training set. We further divided the training dataset into three distinct datasets, denoted as $\mathcal{D}^1$, $\mathcal{D}^2$, and $\mathcal{D}^3$, corresponding to different learning sessions ($\mathcal{S}^1$, $\mathcal{S}^2$, and $\mathcal{S}^3$). Each $\mathcal{S}$ contains a non-overlapped subset of entities and relations in the training dataset. Similarly, the test dataset was partitioned into three corresponding segments: $\mathcal{D}^1_{\mathrm{T}}$, $\mathcal{D}^2_{\mathrm{T}}$, and $\mathcal{D}^3_{\mathrm{T}}$. To demonstrate the advantage of our approach in facilitating continual learning, we compared our method with a classical knowledge graph embedding algorithm HAKE [24]. Compared to HAKE, our method incorporates the synaptic intelligence part as described in Eq. (8) to further enhance its continual learning capability. Our evaluation leverages four distinct metrics from [37] for their relevance in assessing the unique challenges and performance of continual learning.

(1) Average Accuracy $\mathcal{ACC}$: $\mathcal{ACC}_{\mathcal{S}^2}\left(\mathcal{D}^2_{\mathrm{T}}\right)$ measures the average accuracy of the model prediction performance on $\mathcal{D}^2_{\mathrm{T}}$ after the end of the learning session $\mathcal{S}^2$; $\mathcal{ACC}_{\mathcal{S}^3}\left(\mathcal{D}^3_{\mathrm{T}}\right)$ measures the average accuracy on $\mathcal{D}^3_{\mathrm{T}}$ after the end of the learning session $\mathcal{S}^3$; $\mathcal{ACC}_{\mathcal{S}^3}\left(\mathcal{D}^2_{\mathrm{T}}\right)$ measures the average accuracy on $\mathcal{D}^2_{\mathrm{T}}$ after the end of the learning session $\mathcal{S}^3$. The calculation of $\mathcal{ACC}_{\mathcal{S}^2}\left(\mathcal{D}^2_{\mathrm{T}}\right)$ is as follows:

$$\mathcal{ACC}_{\mathcal{S}^2}(\mathcal{D}^2_{\mathrm{T}}) = \frac{1}{\left|\mathcal{D}^2_{\mathrm{T}}\right|}\sum_{(h,r,t)\in\mathcal{D}^2_{\mathrm{T}}} \mathbb{I}\left[f_{\mathcal{S}^2}(h,r) = t\right] \tag{9}$$

Table 3 Average prediction success rate of six methods using tenfold cross-validation

Manipulation parameter to predict	IRMKG	RD	DRD	TM	NBC	MLN
Gripper type	**0.957**	0.157	0.571	0.675	0.721	0.754
Grasping force	**0.981**	0.288	0.602	0.716	0.747	0.808
Grasping component	**0.949**	0.173	0.536	0.643	0.698	0.727
All above	**0.891**	0.009	0.175	0.332	0.371	0.455

Bolded values indicate the highest prediction success rates among all methods for each manipulation parameter where $f_{\mathcal{S}^2}(h,r)$ is the prediction function and $\mathbb{I}$ is the indicator function, taking the value 1 when $f_{\mathcal{S}^2}(h,r)=t$, otherwise 0. Average Accuracy $\mathcal{ACC}$ is a number between 0 and 1, and the larger the better.

(2) Forward Transfer $\mathcal{FWT}$: $\mathcal{FWT}_{\mathcal{S}^3}=\mathcal{ACC}_{\mathcal{S}^2}\left(\mathcal{D}_{\mathrm{T}}^3\right)$ measures the zero-shot learning effect of session $\mathcal{S}^3$ in $\mathcal{D}_{\mathrm{T}}^3$ by transferring the weights learned in the previous session $\mathcal{S}^2$. $\mathcal{FWT}$ is also a number between 0 and 1, and the larger the better.

(3) Backward Transfer $\mathcal{BWT}$: $\mathcal{BWT}_{\mathcal{S}^3}=\max\left(0,\mathcal{ACC}_{\mathcal{S}^3}\left(\mathcal{D}_{\mathrm{T}}^2\right)-\mathcal{ACC}_{\mathcal{S}^2}\left(\mathcal{D}_{\mathrm{T}}^2\right)\right)$ measures the increase in the expected effectiveness of $\mathcal{D}_{\mathrm{T}}^2$ of the previous learning course $\mathcal{S}^2$ due to learning in session $\mathcal{S}^3$. $\mathcal{BWT}$ is a number between 0 and 1, and a higher value indicates an improvement in model prediction accuracy on previous datasets after learning new information.

(4) Remember $\mathcal{REM}$: $\mathcal{REM}_{\mathcal{S}^3}$ measures how the performance in learning session $\mathcal{S}^2$ declines due to learning in subsequent session $\mathcal{S}^3$. The calculation of $\mathcal{REM}_{\mathcal{S}^3}$ is as follows:

$$\mathcal{REM}_{\mathcal{S}^3}=\frac{\sum_{(h,r,t)\in\mathcal{D}_{\mathrm{T}}^2}\mathbb{I}\left[f_{\mathcal{S}^2}(h,r)=t\wedge f_{\mathcal{S}^3}(h,r)=t\right]}{\sum_{(h,r,t)\in\mathcal{D}_{\mathrm{T}}^2}\mathbb{I}\left[f_{\mathcal{S}^2}(h,r)=t\right]} \tag{10}$$

the numerator represents the number of triples that are correctly predicted in $\mathcal{S}^2$ and remain correctly predicted after $\mathcal{S}^3$. An ideal value of $\mathcal{REM}_{\mathcal{S}^3}$ is 1, meaning that old knowledge is fully preserved without decay during the process of learning new knowledge.

In this experiment, our approach IRMKG demonstrated well performance in a continual learning environment as shown in Fig. 5. In terms of $\mathcal{ACC}$, IRMKG achieved an accuracy of 0.701 on the dataset $\mathcal{D}_{\mathrm{T}}^2$ after the end of the second learning session $\mathcal{S}^2$. After the end of the third learning session $\mathcal{S}^3$, this accuracy increased to 0.941 on dataset $\mathcal{D}_{\mathrm{T}}^3$ and further enhanced to 0.963 on the dataset $\mathcal{D}_{\mathrm{T}}^2$. Additionally, its $\mathcal{FWT}_{\mathcal{S}^3}$ score of 0.684 indicates robust zero-shot learning capability, effectively applying previous insights to new data in $\mathcal{D}_{\mathrm{T}}^3$. Furthermore, IRMKG's $\mathcal{BWT}_{\mathcal{S}^3}$ of 0.262 showcases its exceptional ability to improve upon previous knowledge without detriment from new learning sessions, a critical attribute of continual learning systems. In comparison, the HAKE method exhibited less impressive gains across these metrics, signifying its limitations in both integrating new knowledge and enhancing performance on previously learned datasets. Specifically, its lower $\mathcal{ACC}$ ($\mathcal{ACC}_{\mathcal{S}^2}\left(\mathcal{D}_{\mathrm{T}}^2\right)$, 0.684; $\mathcal{ACC}_{\mathcal{S}^3}\left(\mathcal{D}_{\mathrm{T}}^2\right)$, 0.920, $\mathcal{ACC}_{\mathcal{S}^3}\left(\mathcal{D}_{\mathrm{T}}^3\right)$, 0.914), $\mathcal{FWT}_{\mathcal{S}^3}$ (0.663), and $\mathcal{BWT}_{\mathcal{S}^3}$ (0.263) scores highlight a constrained adaptability compared to IRMKG. Additionally, the Remember ($\mathcal{REM}_{\mathcal{S}^3}$) score of 1.0 signified that IRMKG method successfully retained all previous knowledge without any decay, showcasing its exceptional continual learning capability. In contrast, the HAKE method exhibited the score of 0.977, suggesting a minor loss in previously acquired knowledge.

This comparative analysis clearly demonstrates the efficacy of our proposed IRMKG approach in continual learning. By incorporating synaptic intelligence, which applies a regularization based on each embedding component's historical importance, the IRMKG method efficiently balances learning new information and retaining old with enhanced adaptability.

4.3 Real robot manipulation test

To examine the effectiveness of our method in physical settings, we tested various manipulation tasks with a real hybrid robot based on the predictions generated from the IRMKG. This experiment involved 46 objects of 39 different object classes, as shown in Fig. 6. We considered 10 different tasks and created a total of 158 task instances. Notably, not all these objects were present in the training set of the

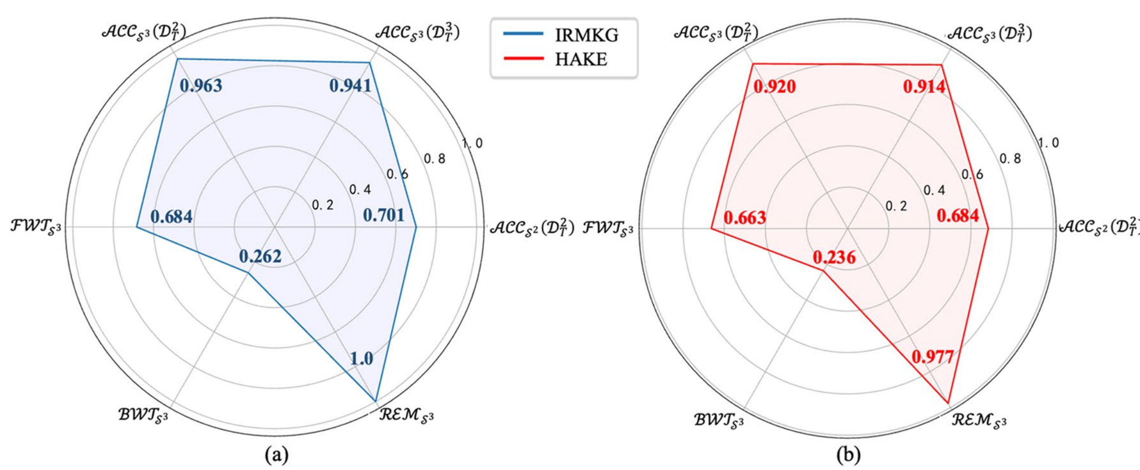

Fig. 5 Performance of continual learning capability of **a** IRMKG and **b** HAKE in six metrics

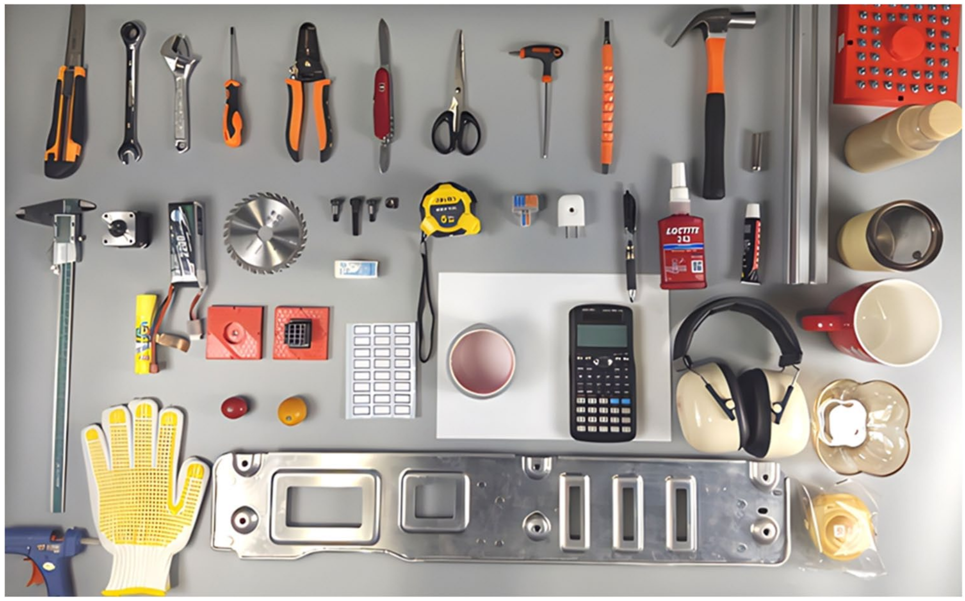

Fig. 6 Variety of experimental objects in real robot manipulation test

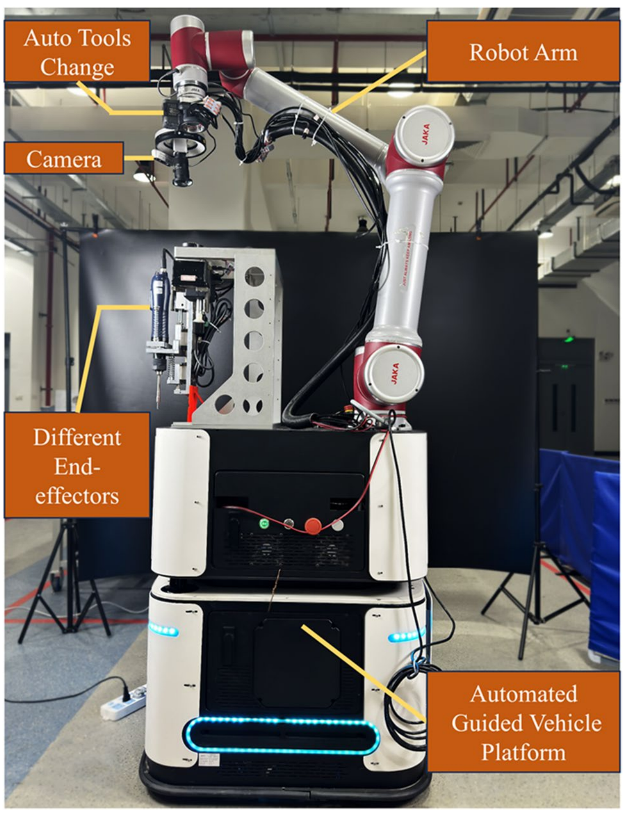

Fig. 7 Hybrid robot in real robot manipulation test

robot's knowledge graph (e.g., the thread-locker glue and the wire stripper). In this experiment, we assumed that accurate object labels, their components, and the materials of these components are provided. Figure 7 shows the hybrid robot used in the experiment to perform real manipulation tasks.

To evaluate the performance of our IRMKG method in selecting robotic manipulation parameters, we developed a rule-based method (denoted as RULE) based on the guidance of expert workers as the comparison benchmark. These rules, while informed by expert judgment, do not always guarantee task success but rather reflect the operational experience of skilled workers. The detailed rules specified by expert workers are as follows:

(1) The robot uses a soft gripper for objects made of ceramics, food, and glass, a parallel jaw for other objects, a soft suction cup for paper, a hard suction cup for metal and glass sheets, and an electric screwdriver for screw task.
(2) For the open task, the robot grasps the lid of the object, the handle is available for lift and pour tasks, and the body or identifiable components for other tasks. For the screw task, the operating point is the screw head.
(3) The grasping force is set to be strong for squeeze tasks and medium for other tasks like push, pull, and rotate. For tasks involving paper or food, the grasping force is set low to avoid damage.

We define the following criteria to evaluate the success of various tasks performed by robots to compare the effectiveness of the two methods in real-world operations. For the grasp task, success is determined to hold a designated component of an object without altering its shape. In the lift task, the robot needs to raise the object at least 10 cm off the ground. The open task requires the robot to lift a component of the object, such as a lid, by a minimum of 10 cm without damaging the object's structure. Pour task is judged by the robot's ability to tilt the object by 30 degrees. In the pull and push tasks, the robot needs to draw the object straightly for at least 10 cm while keeping its orientation stable. The rotate task involves the robot turning the object by 30 degrees. In the squeeze task, the robot is required to grasp and alter the shape of the object. Finally, for the screw task, the robot must rotate and securely fasten a screw or similar object by at least 360°. These criteria provide a comprehensive framework to assess the effectiveness and precision of robotic actions in varied manipulation scenarios.

Table 4 shows the success rates of robot manipulation tasks for both the IRMKG method and the RULE method across various tasks. We find that the IRMKG method outperformed the RULE method with a success rate of 0.968 compared to 0.898 when counting all tasks. On the other hand, the RULE method's failures were mainly attributed to its inability to adapt the force output appropriately, leading to issues like object slippage or inappropriate grasping gripper. For example, as depicted in Fig. 8, the application of medium force by the RULE method led to slippage during the lifting of the stepper motor. Additionally, when performing an opening operation on a cup with a glass lid, the RULE method inappropriately utilized a parallel jaw gripper. A soft gripper would be more suitable in this context to prevent potential damage to the glass lid. Our findings suggest that incorporating inferential capability in robotic manipulation can greatly enhance task success, offering significant improvements over traditional rule-based approaches.

Table 4 Success rates of different methods across various robot manipulation tasks

Manipulation task with number of tasks	IRMKG	RULE
Grasp task (41)	**0.951** (39/41)	0.878 (36/41)
Lift task (35)	**0.943** (33/35)	0.886 (31/35)
Open task (5)	**1.000** (5/5)	0.800 (4/5)
Pour task (5)	**1.000** (5/5)	0.800 (4/5)
Push task (20)	**1.000** (20/20)	0.950 (19/20)
Pull task (20)	**1.000** (20/20)	0.950 (19/20)
Rotate task (26)	**0.962** (25/26)	0.923 (24/26)
Squeeze task (2)	**1.000** (2/2)	1.000 (2/2)
Screw task (4)	**1.000** (4/4)	0.750 (3/4)
Total tasks (158)	**0.968** (153/158)	0.898 (142/158)

4.4 Comparative evaluation of recognition processes

To examine the performance of our proposed Recognition process (named as Combination Recognition I) in the manipulation framework, we compared it with the recognition process in the state-of-the-art semantic grasping method [13] (named as Combination Recognition II). Combination Recognition II integrates Yolo model for object identification, PartNet model for semantic segmentation, and DEP model for material detection. While in our approach Combination Recognition I, we propose to leverage the CLIP model for

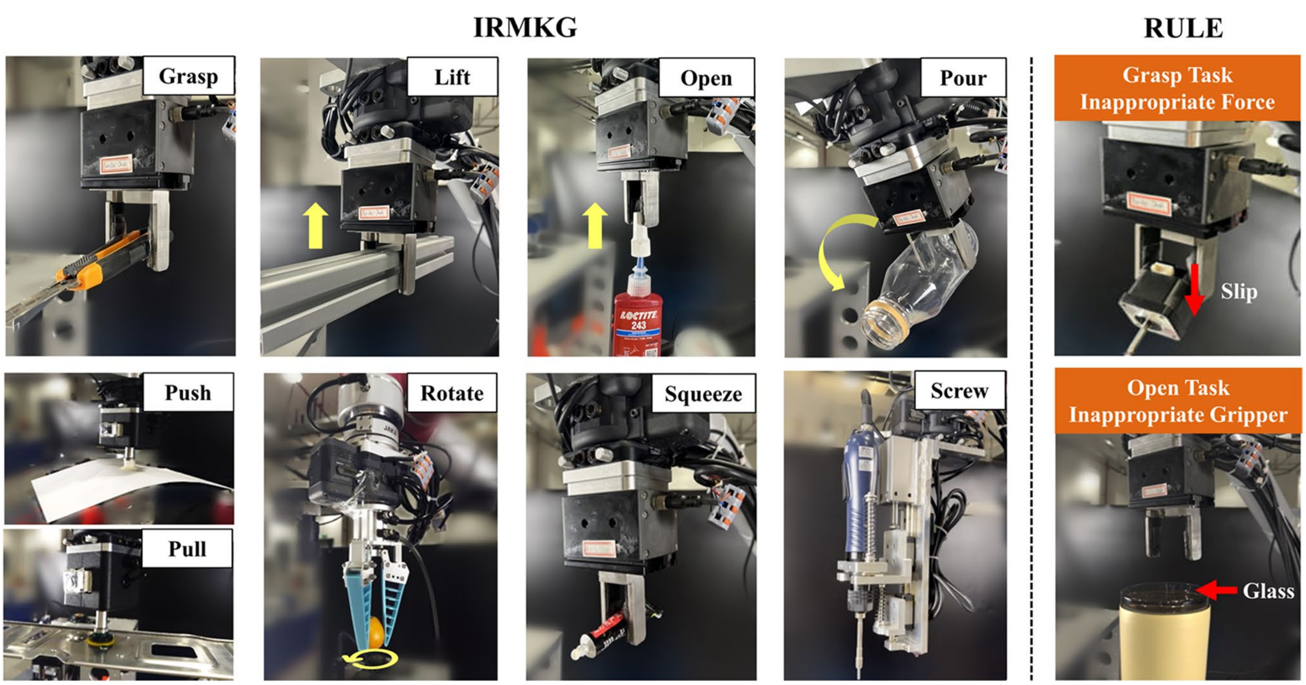

Fig. 8 Demonstration of certain robotic manipulation tasks between IRMKG and RULE

object identification and the PartSLIP model for semantic segmentation to reduce the need of training data.

For model training, we utilized the dataset specified in the literature associated with Combination Recognition II. We used the objects from Section 4.3 as the test case. Among our objects, 41 object instances were expected to be correctly identified by the Recognition process, meaning their object labels, components, and materials could be accurately recognized. Prior to the experiment, we fine-tuned the identification method (CLIP) and semantic segmentation method (PartSLIP) with a few-shot learning approach using objects from our laboratory, while the material detection method (DEP) was trained on the MINC2500 dataset [45].

Table 5 presents the comparison results of the success rates between two recognition processes on difference tasks. Our approach Combination Recognition I achieved a total success rate of 0.902. It returned incorrect answers for 2 objects in object identification, 4 objects in semantic segmentation, and 3 objects in material detection. Notably, even an object is misidentified by the CLIP model in Combination Recognition I, this error may not influence the successful performance of the manipulation task. For example, although CLIP incorrectly recognized a flexible screwdriver as a pipe, it did not affect the subsequent grasping operation, allowing the robot to successfully complete the task.

From Table 5, we also observe that Combination Recognition II only had a success rate of 0.756 when counting all total tasks. Training Yolo and PartNet models require a substantial amount of data, and PartNet needs individual training for each object class to ensure its effectiveness. In contrast, due to the few-shot learning approach, Combination Recognition I does not require extensive training data to adapt CLIP and PartSLIP models to specific industrial context, significantly reducing training time and resource consumption. The DEP model's performance varied between the two recognition processes, with Combination Recognition I demonstrating a higher success rate (0.926) compared to Combination Recognition II (0.829). This improvement is primarily due to Combination Recognition I's component-based detection strategy, which focuses on segmented components of objects for material detection, reducing the interference from non-relevant components. This targeted strategy significantly enhances the detection success rate by isolating object features relevant to the task at hand. In contrast, Combination Recognition II scans the entire object for material properties, proving less effective, particularly for objects with heterogeneous material composition across different components (e.g., the cup in Fig. 8 with a glass lid and a metal body).

Table 5 Success rates of different recognition processes on tasks of object identification, semantic segmentation and material detection

Task	Combination Recognition I	Combination Recognition II
Object identification	**0.951** (39/41)	0.853 (35/41)
Semantic segmentation	**0.902** (37/41)	0.756 (31/41)
Material detection	**0.926** (38/41)	0.829 (34/41)
All above	**0.902** (37/41)	0.756 (31/41)

In summary, our Recognition process harnesses the power of large pre-trained models with a focused, component-based detection strategy, making it a more reasonable choice for object recognition in human–robot collaboration.

5 Discussion

The key findings from our experiments underscore the effectiveness of the IRMKG approach within our industrial robotic manipulation framework. Our approach notably achieved success rates of 95.7% for gripper type selection, 98.1% for grasping force application, and 94.9% for component grasping—substantially surpassing the performance of traditional methods. These results can be primarily attributed to the effective use of advanced knowledge graph embedding techniques, which significantly enhance the system's ability to process and utilize complex relational data. The structured methodology of IRMKG provides a robust framework for encoding detailed knowledge about object properties and task requirements, enabling precise and rapid decision-making capability within the robotic system.

Furthermore, the IRMKG approach exhibited superior continual learning capability, demonstrated by its ability to integrate new data while significantly improving performance on previously learned datasets. Specifically, after successive learning sessions, our method increased accuracy from 70.1% on the second session dataset to 96.3% on the same dataset after the third session, showcasing its effectiveness in retaining and enhancing prior knowledge without degradation. Additionally, our framework achieved a perfect Remember score of 1.0, indicating that it successfully preserved all previously acquired knowledge during the process of learning new information, thus highlighting its exceptional capability in maintaining knowledge consistency. These results demonstrate how the integration of synaptic intelligence has significantly enhanced the framework's continual learning ability, enabling it to dynamically adapt and apply new knowledge without degrading performance on previously learned tasks.

In practical settings, the robustness of our framework was examined through a variety of complex real-world tasks, achieving an overall success rate of 96.8%. It showcases the practical applicability of our system across diverse operational scenarios. Moreover, the comparative evaluation of recognition processes revealed significant advancements in object recognition capability. Utilizing a combination of CLIP, PartSLIP, and DEP models, our recognition process achieved a total success rate of 90.2%. This superior performance is attributed to our few-shot learning approach and a component-based detection strategy that focuses on segmented components of objects, thereby reducing interference and enhancing accuracy. Even in cases of misidentification, such as a flexible screwdriver being mistaken as a pipe, the system's resilience ensured that these errors did not affect the overall task performance.

Our framework significantly enhances human–robot collaboration in industrial environments through its adaptability and continual learning capability. By allowing robots to swiftly adjust to new tasks and changes in operations, it minimizes the burden on human operators and streamlines work flows. This adaptability not only boosts overall productivity but also smooths the integration of robotic systems into diverse factory settings. The continual learning capability of the framework further minimizes the need for frequent manual reprogramming and adjustments, thereby reducing the overall operational costs of smart manufacturing.

6 Conclusion

In this study, we propose a robotic manipulation framework for industrial human–robot collaboration based on continual knowledge graph embedding. This innovative framework enables hybrid robots to perform industrial tasks with less reliance on explicit instructions from human workers. Instead, the robots autonomously decide how to manipulate various objects by utilizing learned semantic information from the operational object and task. It not only alleviates the instructional burden on human workers, but also enhances the precision and efficiency of robotic manipulations. Experimental analysis has shown that our approach achieves up to 96.8% success rate across various tasks involving manipulating commonly seen industrial objects, indicating its feasibility and adaptability in real-world environments. Our research also demonstrates the importance of continual learning in human–robot collaboration. Robots continuously adapt to new environments and tasks through continual learning while retaining their understanding of existing knowledge. This ability enables robots to efficiently meet known task requirements and prepare for potential future challenges. Moreover, the recognition process within our framework does not rely on extensive labeled data, and reduces the time and resources needed for data collection and processing, which is crucial for the rapid and practical deployment of robotic systems in industry.

While our findings are promising, the application of our algorithm in real-time scenarios may face certain constraints. First, the computational demands of processing complex knowledge graphs and maintaining continual learning updates may introduce latency issues, particularly in scenarios where immediate robotic response is critical. Secondly, while the framework has shown high levels of robustness in experimental settings, its performance in highly variable real-world environments still faces significant challenges. Factors such as unanticipated object properties, unexpected environmental conditions, and dynamic changes in task requirements could affect the accuracy and reliability of the system.

In the future, we plan to delve deeper into the complexities and parameter uncertainties of our knowledge graph construction and embedding methods to address these challenges. To mitigate computational complexity and latency issues, we will explore optimization techniques such as graph simplification and parallel processing. By streamlining the knowledge graph structure and leveraging distributed computing resources, we aim to reduce processing times and enhance the system's responsiveness in real-time applications. To tackle parameter uncertainties and enhance the robustness of our framework in diverse real-world environments, we intend to integrate adaptive learning mechanisms and context-aware updates into our continual learning process. This involves developing more sophisticated models capable of dynamically adjusting to new and unforeseen variables, such as changes in object properties and environmental conditions. Additionally, we will incorporate advanced sensor fusion techniques to provide more accurate and comprehensive data inputs, and support the system to better handle dynamic task requirements.

Acknowledgements The authors would like to acknowledge the financial support from the National Key R&D Program of China (2022YFB4702400).

References

1. Yang M, Yu L, Wong C, Mineo C, Yang E, Bomphray I, Huang R (2022) A cooperative mobile robot and manipulator system (Co-MRMS) for transport and lay-up of fibre plies in modern composite material manufacture. Int J Adv Manuf Technol 119:1249–1265. https://doi.org/10.1007/s00170-021-08342-2
2. Mahler J, Matl M, Satish V, Danielczuk M, DeRose B, McKinley S, Goldberg K (2019) Learning ambidextrous robot grasping policies, Sci Robot 4. https://doi.org/10.1126/scirobotics.aau4984
3. Merzić H, Bogdanović M, Kappler D, Righetti L, Bohg J (2019) Leveraging contact forces for learning to grasp, in. Int Conf Robot Autom (ICRA) 2019:3615–3621. https://doi.org/10.1109/ICRA.2019.8793733
4. Lu Y, Fan Y, Deng B, Liu F, Li Y, Wang S (2023) VL-Grasp: a 6-Dof interactive grasp policy for language-oriented objects in cluttered indoor scenes, in. IEEE/RSJ Int Conf Intell Robot Syst (IROS) 2023:976–983. https://doi.org/10.1109/IROS55552.2023.10341379

5. Satish V, Mahler J, Goldberg K (2019) On-policy dataset synthesis for learning robot grasping policies using fully convolutional deep networks. IEEE Robot Autom Lett 4:1357–1364. https://doi.org/10.1109/LRA.2019.2895878
6. Peng Z, Song X, Song S, Stojanovic V (2023) Hysteresis quantified control for switched reaction–diffusion systems and its application. Complex Intell Syst 9:7451–7460. https://doi.org/10.1007/s40747-023-01135-y
7. Wang R, Zhuang Z, Tao H, Paszke W, Stojanovic V (2023) Q-learning based fault estimation and fault tolerant iterative learning control for MIMO systems. ISA Trans 142:123–135. https://doi.org/10.1016/j.isatra.2023.07.043
8. Song X, Wu N, Song S, Zhang Y, Stojanovic V (2023) Bipartite synchronization for cooperative-competitive neural networks with reaction–diffusion terms via dual event-triggered mechanism. Neurocomputing 550:126498. https://doi.org/10.1016/j.neucom.2023.126498
9. Dang H, Allen PK (2012) Semantic grasping: Planning robotic grasps functionally suitable for an object manipulation task, in. IEEE/RSJ Int Conf Intell Robots Syst 2012:1311–1317. https://doi.org/10.1109/IROS.2012.6385563
10. Chu F-J, Xu R, Vela PA (2019) Learning affordance segmentation for real-world robotic manipulation via synthetic images. IEEE Robot Autom Lett 4:1140–1147. https://doi.org/10.1109/LRA.2019.2894439
11. Murali A, Liu W, Marino K, Chernova S, Gupta A (2020) Same object, different grasps: data and semantic knowledge for task-oriented grasping. In: 4th Conference on Robot Learning. Cambridge, pp 1540–1557. https://proceedings.mlr.press/v155/murali21a.html. Accessed 14 Mar 2024
12. Antanas L, Moreno P, Neumann M, Figueiredo RP, Kersting K, Santos-Victor J, Raedt L (2019) Semantic and geometric reasoning for robotic grasping: a probabilistic logic approach. Auton Robots 43:1393–1418. https://doi.org/10.1007/s10514-018-9784-8
13. Kwak JH, Lee J, Whang JJ, Jo S (2022) Semantic grasping via a knowledge graph of robotic manipulation: a graph representation learning approach. IEEE Robot Autom Lett 7:9397–9404. https://doi.org/10.1109/LRA.2022.3191194
14. Fang H-S, Wang C, Fang H, Gou M, Liu J, Yan H, Liu W, Xie Y, Lu C (2023) AnyGrasp: robust and efficient grasp perception in spatial and temporal domains. IEEE Trans Rob 39:3929–3945. https://doi.org/10.1109/TRO.2023.3281153
15. Ten Pas A, Gualtieri M, Saenko K, Platt R (2017) Grasp pose detection in point clouds. Int J Rob Res 36:1455–1473. https://doi.org/10.1177/0278364917735594
16. Liu W, Daruna A, Chernova S (2020) CAGE: context-aware grasping engine, in. IEEE Int Conf Robotics Automation (ICRA) 2020:2550–2556. https://doi.org/10.1109/ICRA40945.2020.9197289
17. Cheng H-T, Koc L, Harmsen J, Shaked T, Chandra T, Aradhye H, Anderson G, Corrado G, Chai W, Ispir M, Anil R, Haque Z, Hong L, Jain V, Liu X, Shah H (2016) Wide & deep learning for recommender systems. In: Proceedings of the 1st Workshop on Deep Learning for Recommender Systems, New York, pp 7–10. https://doi.org/10.1145/2988450.2988454.
18. Duan S, Tian G, Wang Z, Liu S, Feng C (2023) A semantic robotic grasping framework based on multi-task learning in stacking scenes. Eng Appl Artif Intell 121:106059. https://doi.org/10.1016/j.engappai.2023.106059
19. Cai C, Jiang Z, Wu H, Wang J, Liu J, Song L (2024) Research on knowledge graph-driven equipment fault diagnosis method for intelligent manufacturing. Int J Adv Manuf Technol 130:4649–4662. https://doi.org/10.1007/s00170-024-12998-x
20. Bordes A, Usunier N, Garcia-Duran A, Weston J, Yakhnenko O (2013) Translating embeddings for modeling multi-relational data. Adv Neural Inf Process Syst, Curran Associates, Inc. https://proceedings.neurips.cc/paper_files/paper/2013/file/1cecc7a77928ca8133fa24680a88d2f9-Paper.pdf. Accessed 14 Mar 2024
21. Wang Z, Zhang J, Feng J, Chen Z (2014) Knowledge graph embedding by translating on hyperplanes. Proceedings of the AAAI Conference on Artificial Intelligence 28. https://doi.org/10.1609/aaai.v28i1.8870
22. Sun Z, Deng Z-H, Nie J-Y, Tang J (2019) RotatE: knowledge graph embedding by relational rotation in complex space. In: International Conference on Learning Representations. https://openreview.net/forum?id=HkgEQnRqYQ. Accessed 14 Mar 2024
23. Yang B, Yih W, He X, Gao J, Deng L (2015) Embedding entities and relations for learning and inference in knowledge bases. In: 3rd International Conference on Learning Representations, ICLR 2015, San Diego. https://arxiv.org/abs/1412.6575
24. Zhang Z, Cai J, Zhang Y, Wang J (2020) Learning hierarchy-aware knowledge graph embeddings for link prediction. Proceedings of the AAAI Conference on Artificial Intelligence 34:3065–3072. https://doi.org/10.1609/aaai.v34i03.5701
25. Saxena A, Kochsiek A, Gemulla R (2022) Sequence-to-sequence knowledge graph completion and question answering. In: Proceedings of the 60th Annual Meeting of the Association for Computational Linguistics, Stroudsburg. pp 2814–2828. https://doi.org/10.18653/v1/2022.acl-long.201
26. Zhu Y, Wang X, Chen J, Qiao S, Ou Y, Yao Y, Deng S, Chen H, Zhang N (2023) LLMs for knowledge graph construction and reasoning: recent capabilities and future opportunities, ArXiv abs/2305.13168. https://doi.org/10.48550/arXiv.2305.13168
27. Xie X, Zhang N, Li Z, Deng S, Chen H, Xiong F, Chen M, Chen H (2022) From discrimination to generation: knowledge graph completion with generative transformer. In: Companion Proceedings of the Web Conference 2022, ACM, New York, pp 162–165. https://doi.org/10.1145/3487553.3524238
28. Wei Y, Chen W, Li Z, Zhao L (2021) Incremental update of knowledge graph embedding by rotating on hyperplanes, in. IEEE Int Conf Web Services (ICWS) 2021:516–524. https://doi.org/10.1109/ICWS53863.2021.00072
29. Yan H, Wang J, Chen J, Liu Z, Feng Y (2022) Virtual sensor-based imputed graph attention network for anomaly detection of equipment with incomplete data. J Manuf Syst 63:52–63. https://doi.org/10.1016/j.jmsy.2022.03.001
30. Parisi GI, Kemker R, Part JL, Kanan C, Wermter S (2019) Continual lifelong learning with neural networks: a review. Neural Netw 113:54–71. https://doi.org/10.1016/j.neunet.2019.01.012
31. Zenke F, Poole B, Ganguli S (2017) Continual learning through synaptic intelligence. In: Proceedings of the 34th International Conference on Machine Learning - vol. 70, JMLR.org, pp 3987–3995. https://doi.org/10.48550/arXiv.1703.04200
32. Kirkpatrick J, Pascanu R, Rabinowitz N, Veness J, Desjardins G, Rusu AA, Milan K, Quan J, Ramalho T, Grabska-Barwinska A, Hassabis D, Clopath C, Kumaran D, Hadsell R (2017) Overcoming catastrophic forgetting in neural networks. Proc Natl Acad Sci 114:3521–3526. https://doi.org/10.1073/pnas.1611835114
33. Rusu AA, Rabinowitz NC, Desjardins G, Soyer H, Kirkpatrick J, Kavukcuoglu K, Pascanu R, Hadsell R (2016) Progressive neural networks, CoRR abs/1606.04671. http://arxiv.org/abs/1606.04671
34. Lomonaco V, Maltoni D (2017) CORe50: a new dataset and benchmark for continuous object recognition. In: Conference on Robot Learning. https://api.semanticscholar.org/CorpusID:20848702. Accessed 14 Mar 2024
35. Shin H, Lee JK, Kim J, Kim J (2017) Continual learning with deep generative replay. In: Proceedings of the 31st International Conference on Neural Information Processing Systems, Curran Associates Inc., Red Hook, pp 2994–3003. https://doi.org/10.48550/arXiv.1705.08690

36. Wei Y, Chen W, Wen S, Liu A, Zhao L (2023) Knowledge graph incremental embedding for unseen modalities, Knowl. Inf Syst 65:3611–3631. https://doi.org/10.1007/s10115-023-01868-9
37. Daruna AA, Gupta M, Sridharan M, Chernova S (2021) Continual learning of knowledge graph embeddings, IEEE Robot Autom Lett 6:1128–1135. https://doi.org/10.1109/LRA.2021.3056071
38. Radford A, Kim JW, Hallacy C, Ramesh A, Goh G, Agarwal S, Sastry G, Askell A, Mishkin P, Clark J, Krueger G, Sutskever I (2021) Learning transferable visual models from natural language supervision. In: International Conference on Machine Learning. https://api.semanticscholar.org/CorpusID:231591445. Accessed 14 Mar 2024
39. Liu M, Zhu Y, Cai H, Han S, Ling Z, Porikli F, Su H (2023) PartSLIP: low-shot part segmentation for 3D point clouds via pretrained image-language models, in: 2023 IEEE/CVF Conference on Computer Vision and Pattern Recognition (CVPR), IEEE Computer Society, Los Alamitos, CA, USA, 21736–21746. https://doi.org/10.1109/CVPR52729.2023.02082
40. Xue J, Zhang H, Nishino K, Dana KJ (2022) Differential viewpoints for ground terrain material recognition. IEEE Trans Pattern Anal Mach Intell 44:1205–1218. https://doi.org/10.1109/TPAMI.2020.3025121
41. Ng A, Jordan M (2001) On discriminative vs. generative classifiers: a comparison of logistic regression and naive Bayes. Adv Neural Inf Process Syst, MIT Press. https://proceedings.neurips.cc/paper_files/paper/2001/file/7b7a53e239400a13bd6be6c91c4f6c4e-Paper.pdf. Accessed 14 Mar 2024
42. Daruna AA, Liu W, Kira Z, Chernova S (2019) RoboCSE: robot common sense embedding, 2019 International Conference on Robotics and Automation (ICRA), pp 9777–9783. https://api.semanticscholar.org/CorpusID:67856661. Accessed 14 Mar 2024
43. Jiang L, Wang D, Cai Z (2007) Scaling up the accuracy of Bayesian network classifiers by m-estimate. In: Advanced Intelligent Computing Theories and Applications. With Aspects of Artificial Intelligence. ICIC. https://doi.org/10.1007/978-3-540-74205-0_5
44. Nyga D, Picklum M, Beetz M (2019) pracmln: Markov Logic Networks in Python. https://www.pracmln.org. Accessed 14 Mar 2024
45. Bell S, Upchurch P, Snavely N, Bala K (2015) Material recognition in the wild with the materials in context database, in. IEEE Conference on Computer Vision and Pattern Recognition (CVPR) 2015:3479–3487. https://doi.org/10.1109/CVPR.2015.7298970