Journal of Manufacturing Systems 68 (2023) 304–312

Contents lists available at ScienceDirect

Journal of Manufacturing Systems

journal homepage: www.elsevier.com/locate/jmansys

Self-organising multiple human–robot collaboration: A temporal subgraph reasoning-based method

Shufei Li [a,b], Pai Zheng [a,*], Shibao Pang [b,c], Xi Vincent Wang [b], Lihui Wang [b,*]

[a] Department of Industrial and Systems Engineering, The Hong Kong Polytechnic University, Hong Kong Special Administrative Region of China
[b] Department of Production Engineering, KTH Royal Institute of Technology, Sweden
[c] School of Mechanical and Electronic Engineering, Wuhan University of Technology, Wuhan, China

ARTICLE INFO

Keywords:
Human–robot collaboration
Self-organising manufacturing
Knowledge graph
Task allocation
Assembly

ABSTRACT

Multiple Human–Robot Collaboration (HRC) requires self-organising task allocation to adapt to varying operation goals and workspace changes. However, nowadays an HRC system relies on predefined task arrangements for human and robot agents, which fails to accomplish complicated manufacturing tasks consisting of various operation sequences and different mechanical parts. To overcome the bottleneck, this paper proposes a temporal subgraph reasoning-based method for self-organising HRC task planning between multiple agents. Firstly, a tri-layer Knowledge Graph (KG) is defined to depict task-agent-operation relations in HRC tasks. Then, a subgraph mechanism is introduced to learn node embeddings from subregions of the HRC KG, which distills implicit information from local object sets. Thirdly, a temporal reasoning module is leveraged to integrate features from previous records and update the HRC KG for forecasting humans' and robots' subsequent operations. Finally, a car engine assembly task is demonstrated to evaluate the effectiveness of the proposed method, which outperforms other benchmarks in experimental results.

1. Introduction

Conventional automation systems fail to replace manual work in various manufacturing processes, such as the assembly of complicated and fine-fabricated mechanical components [1]. To improve human wellbeings and achieve a high degree of flexible production, Human–Robot Collaboration (HRC) is nowadays of particular interest in dexterous manufacturing tasks [2,3]. HRC is characterised by the combination of complementing competencies, i.e., the strength of robots and the flexibility of humans [4,5]. Human and robotic agents in HRC systems can work side by side in close proximity and share resources and task goals for complex task fulfilment.

Towards human-centric smart manufacturing, numerous research efforts on HRC have been sparked to ensure human safety, enable adaptive robot control and make reasonable task planning in the execution loop. Human pose tracking methods were introduced to improve context awareness capabilities in HRC, which provided a prerequisite for collision-free collaboration [6]. Then, a human action prediction approach was proposed, based on which a robot could learn about human operators' intentions and make proactive motion planning decisions [7]. With the perceptual results among human–robot and the surrounding environment, HRC Knowledge Graph (KG) was introduced to generate adaptive task planning for human and robotic manipulation [8]. The advanced computer vision and KG technologies allow HRC systems to embrace holistic perception capabilities and cognitive intelligence.

Nevertheless, most exploration of HRC settings only involves one single human operator and one robot, which may fail to tackle some complex tasks relying on hybrid multiple human–robot agents' work. For example, the assembly of large-scale complex products was completed with co-working between operators acting in specific roles and robots with different payloads [9]. Two challenges still lie in achieving multi-agent (i.e., humans or robots) teamwork in HRC. Firstly, a notable problem is how to make and assign global optimal operation arrangements to human and robotic agents considering their qualified capabilities and task policies. Furthermore, the knowledge representation of multi-agent task assignments needs to dynamically update with temporal execution results and optimise co-working decisions along task progression in different stages.

Aiming to bridge this research gap, this paper proposes a self-organising HRC architecture, which introduces temporal subgraph neural networks to forecast task allocation for hybrid multi-agent collaboration in complex manufacturing tasks. The rest of this article is organised as follows. Section 2 introduces related works to improve cognitive

* Corresponding authors.
E-mail addresses: pai.zheng@polyu.edu.hk (P. Zheng), lihuiw@kth.se (L. Wang).

https://doi.org/10.1016/j.jmsy.2023.03.013
Received 5 January 2023; Received in revised form 12 February 2023; Accepted 26 March 2023
Available online 21 April 2023
0278-6125/

and self-organising intelligence in HRC, including visual reasoning and KG techniques, and dynamic task planning methods. Section 3 depicts a framework of self-organising HRC, followed by a temporal subgraph method for knowledge representation learning of task allocation and forecasting human–robot operation arrangements. Section 4 shows a demonstration of the self-organising HRC in a car engine assembly task, which evaluates the performance of task planning for multiple human and robotic agents. Section 5 further discusses the connotation of self-organising HRC driven by KG and experimental results. Finally, Section 6 concludes the study and highlights future research directions.

2. Related work

This section summarises the current development of HRC and its implementations in industrial settings, followed by a comprehensive review of cutting-edge Artificial Intelligence (AI) technologies and task planning methods for enhanced intelligence capabilities in HRC.

2.1. Human-Robot collaboration

HRC is vital for flexible automation in modern factories, like the production of medium and small-batch products with a large variety. Nowadays, numerous approaches are emerging to refine the connotation of HRC systems. Proactive HRC was proposed, which immerses mutual-cognitive, predictable, and self-organising intelligence into the human–robot execution loop [10]. In this context, a scene graph-based method was proposed to learn human–robot relations in task processes and generate co-work strategies, which provided guidance to human manual operation and adaptive instructions to robot motions [11]. For predictable intelligence, a robot could predict human motion sequences and respond the human action in advance in HRC tasks [12]. The self-organising HRC plans multiple agents' operations and integrates their best competencies by learning knowledge of human–robot capable capabilities and task structures. HRC systems with these intelligence abilities present widespread application potential.

In the last decade, various technologies were explored for the implementation of HRC in manufacturing. A Virtual Reality (VR) based interface was developed to allow operators to demonstrate robot trajectory in belt grinding tasks [13]. To improve user experience in HRC, Hietanen et al. [14] showed safe zones, robot control buttons, and robot status in an Augmented Reality (AR) environment. For human–robot co-manipulation, Kana et al. [15] combined impedance control and haptic rendering methods to let a user guide robot motions. Buerkle et al. [16] used a mobile electroencephalogram (EEG) to detect potential emergencies in HRC, which prevent a robot from dropping a workpiece or crushing the assembly piece. These explorations on human safety, robot teleoperation, and task guidance facilitate HRC applications for complex tasks containing different operation stages.

2.2. Visual reasoning and knowledge graph

Visual reasoning aims at learning the latent meaning of visual cues, such as relationships between different objects from visual observations [17]. Zeng et al. [18] proposed semantic linking maps to help a robot to infer locations of previously unseen objects, by exploiting spatial relations between prior knowledge and targets. On the other side, KG can utilise graphical structures to semantically extract knowledge from interconnected objects, events, and concepts [19]. KG with the subgraph [20] and temporal reasoning layers [21] can extract implicit connectivity information from internal topology and temporal dynamics, respectively. For example, Wang et al. [22] leveraged KG to extract knowledge of user requirements from large volumes and heterogeneous contextual data of products.

Based on these research efforts, the visual reasoning approach combining KG techniques is arising for knowledge inference in manufacturing scenarios. With extracted visual features, the KG algorithm was utilised to integrate contextual information from object pairs and graphically describe their relations [23]. Sun et al. [24] developed HRC task KG and used recognised human motions to retrieve robot action sequences in the KG. Zhou et al. [25] built an assembly process KG by integrating CAD model information of complex components and their assembly sequences for the interference analysis task of assembly operations. The above crucial research has shown the potential of making reasonable resource allocation decisions in HRC via visual reasoning and KG methods.

2.3. Multi-agent interaction and task planning

Hybrid collaboration between multiple humans and robots is essential for many manufacturing tasks, like the assembly of large-scale products in unstructured spaces. For role allocation in HRC, Lamon et al. [26] calculated performance indices of task complexity, agent dexterity, and agent effort to assign optimal tasks to different agents. Antakli et al. [27] developed a 3D simulation environment for production tasks with hybrid team members, where workers tightly cooperate with robots and receive task assignments. Patnayak et al. [28] designed a wearable super-computing platform to provide networking capabilities and computational processing resources for multi-agent systems, which promoted realistic applications of multiple human–robot collaborative systems.

Besides, some AI algorithms are embedded into multi-agent HRC systems for optimal task planning and resource allocation. Galin et al. [29] described different human–robot interaction types for effective information exchange and leveraged a particle swarm algorithm for task distribution. Liau et al. [30] decomposed assembly operations into actions and analysed the characteristics of these actions to obtain the agent assignment preference. With the analytic process, a genetic algorithm was applied for task allocation in HRC systems with two robots. Karami et al. [31] used an AND-OR graph to describe task characteristics, operator capabilities, and robot functionalities for the sequential allocation of tasks between a human operator, a mobile manipulator, and a dual-arm manipulator for defect inspection tasks.

From the literature, one can find that task decomposition is conducted in advance to generate sequential operations, which remain static during the multi-agent cooperation stage. Multiple human–robot collaborative working processes lack self-organising capabilities for dynamic task assignments and source allocation when facing workspace changes. Motivated by this, a novel method should be proposed for self-organising collaborative task planning between multiple humans and robots.

3. Methodology

The overall architecture of the proposed self-organising HRC between multiple human–robot agents is shown in Fig. 1. Real-time visual data of HRC workspaces is captured by a depth camera for the perception of the task execution over time. With the visual input, a graph R-CNN algorithm [23] is leveraged to detect human(s), robots, and components as KG nodes (entities) from the task execution loop, and then to dynamically identify relationships between these nodes for Scene Graph (SG) construction in different timestamps. The constructed temporal SG is a part of HRC KG and connected with other nodes in the HRC KG for temporal subgraph reasoning. The HRC KG contains human, robot, component, and action nodes. Meanwhile, prior nodes of these elements from previous HRC execution stages are recorded in the graph. On one side, the subgraph reasoning module aggregate various subregions of node sets in KG to learns HRC task planning knowledge among task-agent-operation levels. On the other side, the temporal reasoning module extracts temporal information in different task stages by tracing back historical KG and prior nodes for HRC KG updates. Lastly, multi-agent task planning is generated with the update of HRC KG, which infers and forecasts human and robotic operations for fluent HRC task fulfilment and efficient co-working. The task planning strategies are delivered to human and robotic agents with communication technologies such as in an AR environment [32].

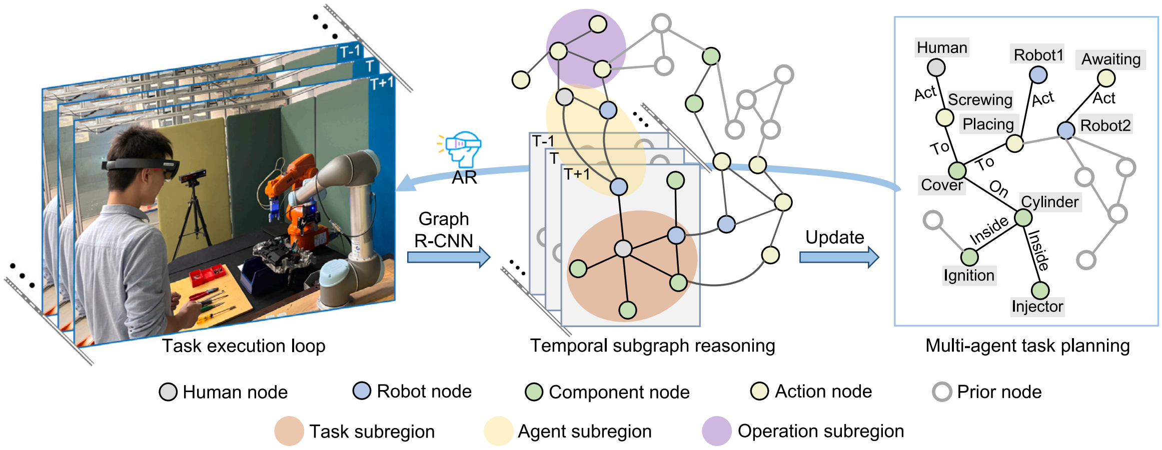

Fig. 1. The framework of self-organising HRC based on a temporal subgraph reasoning method.

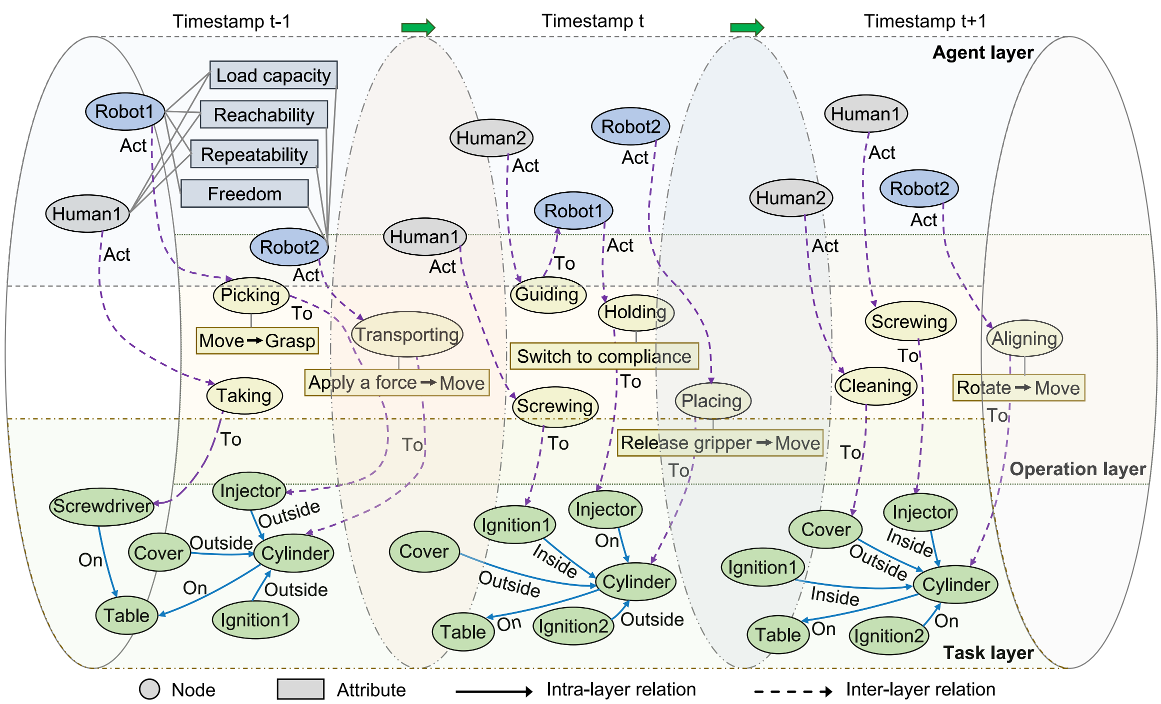

Fig. 2. The construction of temporal tri-layer HRC KG for multi-agent task allocation.

3.1. Tri-layer HRC KG for time-changing task allocation

A temporal HRC KG across task-agent-operation layers is built to describe dynamic task allocation for self-organising HRC involving multiple agents, as presented in Fig. 2. Different nodes containing various attributes are connected by intra-layer relations and inter-layer relations, formulating the HRC KG. In the task process, the HRC KG stores memory of historical operation records and updates the human–robot-environment status for time-changing task decomposition and assignment.

The task layer describes relationships between various components and contains semantic knowledge of task fulfilment situations. The output of the graph R-CNN module, i.e., SG, represents the graph architecture of the task layer and is connected to other subgraphs of the HRC KG. At different timestamps, the task layer can find the changing relations between these component pairs, which denotes different HRC task stages.

The agent layer depicts different task accomplishing capabilities and working roles between HRC heterogeneous teams composed of multiple humans and robots. The priority degree to operate a task by a human operator is evaluated by three metrics: load capacity, reachability, and repeatability. Together with freedom degrees, these kinematic and dynamic characteristics of a robot identify whether it is capable of accomplishing a task. Along the time, the agent layer calculates different reachability levels for humans and robots based on the varying distance between these agents and target components. Meanwhile, the load capacity and repeatability of human operation are fine-adjusted with the human operator continuously working in different task stages. On the other hand, active and supportive roles are on-demand assigned to these agents in different stages to ease each other's work.

The operation layer decomposes potential human–robot operations into a sequence of atomic actions and associates these actions for different task execution. The atomic actions can be assigned to and directly executed by these human and robotic agents. For example, to assemble a fuel injection tube into an engine cylinder head, a robot is required to pick up the tube and transport it above the cylinder head. Then, a human operator can assist the robot to align the tube

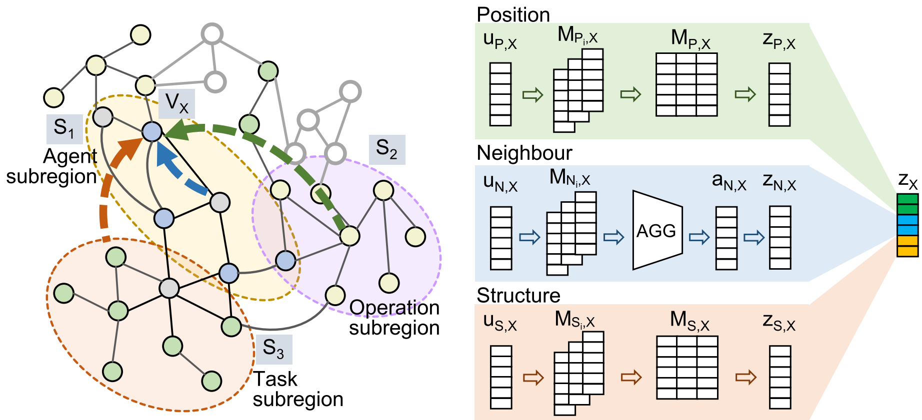

Fig. 3. The procedure of subgraph message passing and encoding.

to the holes of the cylinder head and place the tube into the cylinder head. The picking operation includes atomic actions of moving and grasping, whereas the transporting procedure involves applying a force and moving action. Similarly, the aligning operation contains atomic actions of rotating and moving, while the sequential placing operation requires releasing the gripper and the moving step. The link from the operation layer to the task layer points to target components for an action, which gives a robot trajectory terminus position. Thus, across different task stages, the operation layer arranges suitable operation sequences for task execution.

With task progress, the tri-layer HRC KG is temporally constructed by connecting human nodes, robot nodes, action nodes, and component nodes together. The HRC KG indicates the human and robotic operation arrangement, target manipulated components and task goals, achieving self-organising task planning between multi-agents. For the HRC KG construction, two reasoning processes, i.e., the subgraph and temporal mechanism, are introduced as follows.

3.2. Subgraph message passing and encoding

$G = (V, E)$ represents a temporal HRC KG in a timestamp t. The node set V consists of humans, robots, components, and action entities, whereas the edge set E describes their relations. The subgraph $S = (V', E')$ is used to denote a set of nodes and their relations in task, operation and agent layers. The subgraph encoding approach is utilised to learn the embedding representation of each node for link prediction by capturing graph topology between connected components, like different subgraphs $S = (S_1, S_2, \dots, S_n)$. The subgraph representation is distilled from three stepwise procedures. Firstly, a message is propagated to a node v_i from one of its neighbours $v_j \in \mathcal{N}_{v_i}$. Then, these messages passed from all neighbourhoods are aggregated with different weights. Lastly, the aggregated message is encoded with an activation function and defined as a hidden representation for the layer-wise update.

The subgraph topology is embodied in internal and border connections from position, neighbour, and structure levels, as shown in the left part of Fig. 3. For a node in one subgraph, the internal position calculates the distance between nodes within the same subgraph. In contrast, the border position denotes the distance reaching other nodes outside this subgraph. The internal neighbour identifies directly connected nodes in a subgraph, whereas the border neighbour denotes a set of nodes in the entire KG with the most relatedness ranking the top k scores. The internal structure depicts the connectivity of node sets for each subgraph, while the border structure connects a node inside a subgraph with its border neighbour. In this way, the temporal HRC KG information is captured from multiple distinct subgraphs.

For subgraph S, the message channels from the aforementioned position, neighbour and structure to a node V_X are defined as $U_{P,X}$, $U_{N,X}$, and $U_{S,X}$, respectively, as presented in the left part of Fig. 3. In detail, a message propagated to the node V_X from each subgraph S_i is calculated by,

$$MSG_i^{S \to V_X} = W_i \cdot (V_X, S_i) \tag{1}$$

where W_i is learnable weight parameters for different subgraphs. The message-passing process results representation of $M_{P_i,X}$, $M_{N_i,X}$, and $M_{S_i,X}$ for the three channels of subgraphs. Then, an aggregation function is applied to these messages over all subgraphs. At the layer l, the non-linear activation function σ is leveraged to transform the aggregated message on position, neighbour, and structure channels as a hidden representation for the next layer iteration. The process is depicted as,

$$\begin{aligned} a_{i,X} &= AGG(MSG_i^{S \to V_X}\ \forall S_i \in S) \\ Z_{i,X}^{l+1} &\leftarrow \sigma(a_{i,X}; Z_{i,X}^{l}) \end{aligned} \tag{2}$$

With the layer-wise message passing, the internal and border properties of subgraphs are encoded from three channels as property-aware output representations, denoted as $\{Z_{P_I,X}, Z_{P_B,X}\}$, $\{Z_{N_I,X}, Z_{N_B,X}\}$, and $\{Z_{S_I,X}, Z_{S_B,X}\}$. Finally, these representations are concatenated to produce the node embedding Z_X, which contains positional and structural information of subgraphs.

3.3. Temporal reasoning process and update mechanism

A temporal KG is represented by $G = (V, E, T)$, where T is a timestamp. At time t, a quadruple $\mathcal{I} = (v_s, p_j, v_o, t)$ depicts a connection between a subject entity v_s and an object entity v_o with an edge (predicate) p_j. With these properties, the temporal KG in HRC tasks assigns dynamic operation arrangements to human–robot teams for adaptive task accomplishment based on observed scenario changes. The temporal KG construction process is presented in Fig. 4, comprised of node embedding, edge attention propagation, and graph update.

The node embedding module integrates information from temporal sampling and subgraph sampling. Observed quadruples denoting prior human–robot-component-action relations are used to develop temporal node embedding, as shown in the upper left corner in Fig. 4. For node v_s, its prior fact $\mathcal{Q} = (v_s, p_k, v_p, t' | t' < t)$ is collected to share knowledge of HRC accomplished operations across timestamps. The temporal embedding of node v_s is calculated by a message passing from the set of its prior neighbours $\hat{\mathcal{N}}_{v_s}$, as denoted in Eqs. (1) and (2). On the other hand, the node subgraph embedding is obtained from the aforementioned subgraph encoding process. By concatenating the temporal embedding and subgraph embedding, the entity features evolve over time and exhibit subgraph patterns. The time-aware and subgraph-based node embedding is denoted as $Z_s(x,t) = [\bar{Z}_x \| Z_t]^T \in \mathbb{R}^{d_X + d_T}$. In this context, $\bar{Z}_x \in \mathbb{R}^{d_X}$ is the subgraph embedding to let an

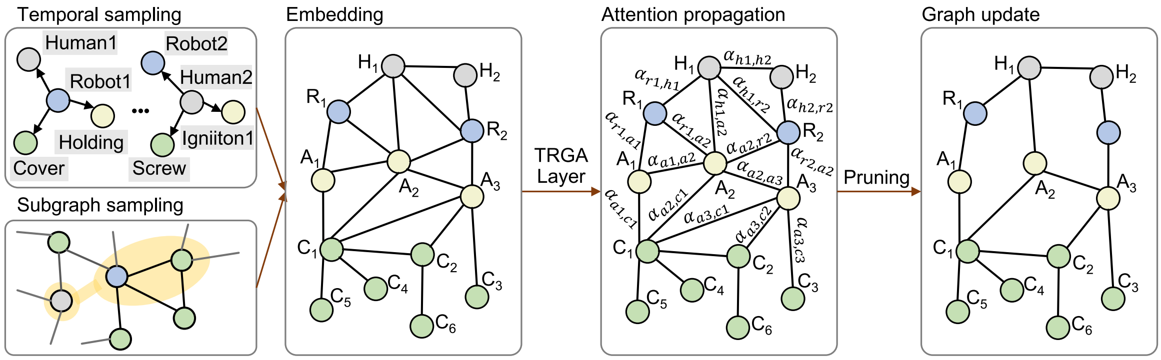

Fig. 4. The procedure of temporal graph reasoning and update.

entity access subregion information in HRC KG, where d_X is the feature dimensionality. Meanwhile, with d_T embedding dimensionality, $Z_t \in \mathbb{R}^{d_T}$ captures time-changing information from prior facts $\mathcal{Q}$ and learns HRC task progressing situations with a broad view of temporal graph structure. Lastly, a stationary embedding vector p_k is learned from the updated graph structure for predicate features between different entities.

After the embedding process, the Temporal SubGraph Attention (TSGA) layer is proposed to distill a graph representation aware of temporal and subgraph relations with the input of entity embedding and predicate embedding. An edge attention score from a subject entity to an object entity is produced by the TSGA layer, followed by a new output of hidden representations for node sets. Varying importance levels are assigned to different edges with the edge attention score, which is calculated by following,

$$e^l_{so}(z_{so}, p_j) = W^l_s (h^{l-1}_s \| p^{l-1}_j) W^l_o (h^{l-1}_o \| p^{l-1}_j) \tag{3}$$

where e^l_{so} is the edge attention score for the quadruple $\mathcal{I}$. z_{so} is the node pair of z_s and z_o. h^{l-1}_s and h^{l-1}_o are hidden representations at layer $l-1$ for node embeddings $Z_s(x,t)$ and $Z_o(x,t)$, respectively. p_j is the predicate embedding. W_s and W_o are two learnable weight matrices. The edge attention mechanism fuses predicate features and entity features to capture relation information in the graph. Then, the attention score is normalised by,

$$\alpha^l_{so}(z_{so}, p_j) = \frac{\exp(e^l_{so}(z_{so}, p_j))}{\sum_{v_c \in \mathcal{N}_{v_s}} \sum_{p_i \in \mathcal{P}_{sc}} \exp(e^l_{sc}(v_s, v_c))} \tag{4}$$

where $\mathcal{N}_{v_s}$ is the set of neighbours of entity v_s. $\mathcal{P}_{sc}$ denotes the edge set linking node v_s and node v_c. The next step is to aggregate hidden representations of these neighbours with different weight scores, as denoted by the following,

$$\hat{h}^l_s(z_s) = \sum_{v_c \in \mathcal{N}_{v_s}} \sum_{p_i \in \mathcal{P}_{sc}} \alpha^l_{sc}(z_{sc}, p_i) h^{l-1}_c(v_c) \tag{5}$$

Then, the activation function LeakyReLU is leveraged to update node embedding of z_s by activating its layer-wise hidden representation and aggregated features from neighbours. Meanwhile, the predicate embedding is updated for next-layer representation aggregation. The update function in the TSGA layer is denoted as,

$$\begin{aligned} h^l_s(z_s) &= \sigma(W^l_h(\beta h^{l-1}_s(z_s) + (1-\beta)\hat{h}^l_s(z_s) + b^l_n)) \\ p^l_i &= W^l_h p^{l-1}_i + b^l_h, \forall p_i \in \mathcal{P}_{sc} \end{aligned} \tag{6}$$

Where β is the hyperparameter to balance the influence between hidden and aggregated features for embedding updates. W^l_h and b^l_h are the weight matrix and bias vector in activating function, respectively.

Various attention scores are assigned to different graph edges after the HRC KG passes the TSGA layer. Based on propagated edge attention score, the attention value of node v_s at the l layer is calculated by the following,

$$c^l_{v_s} = \sum_{v_c \in \mathcal{N}_{v_s}} \sum_{p_i \in \mathcal{P}_{sc}} \alpha^l_{sc}(z_{sc}, p_i) c^{l-1}_{v_c} \tag{7}$$

Then, the edge attention score and node attention value are integrated to get the edge contribution for predicate p_j between entity v_s and entity v_o. The edge contribution score is denoted as $a_{so}(p_j) = \alpha^l_{so}(z_{so}, p_j) c^l_{v_s}$. Finally, the top K edges are retained after ranking these edge contribution scores in the pruning stage. The HRC KG is updated and constructed with these edges connecting different nodes.

For the supervision training process, binary cross-entropy is utilised to learn node embeddings and optimise model parameters in messages passing iterations. The loss function is denoted as,

$$\begin{aligned} L = -\frac{1}{|\mathcal{P}|} \sum_{p_i \in \mathcal{P}_{si}} \frac{1}{|\mathcal{P}_{si}|} \sum_{v_i \in \mathcal{N}_{v_s}} & (y_{v_i, p_i} \log(\frac{c_{v_i}}{\sum_{v_j \in \mathcal{N}_{v_s}} c_{v_j}}) \\ & + (1 - y_{v_i, p_i}) \log(1 - \frac{c_{v_i}}{\sum_{v_j \in \mathcal{N}_{v_s}} c_{v_j}})) \end{aligned} \tag{8}$$

where $\mathcal{P}$ is the set of quadruples in the graph. $\mathcal{P}_{si}$ is the set of node pairs between entity v_s and its neighbours $v_i \in \mathcal{N}_{v_s}$, and p_i is the predicate connecting the node pair. c_{v_i} is the node attention value for entity v_i, and y_{v_i, p_i} is the label that denotes whether v_i is the object entity for v_s linked by p_i.

4. Case study and experimental results

In this section, a case study of self-organising task allocation between multiple human and robotic agents is conducted to collaboratively assemble a partial car engine. Comparison experiments are carried out to evaluate the performance of the proposed temporal subgraph reasoning method.

4.1. Temporal subgraph explanation for HRC assembly

HRC can improve human working conditions in the car engine assembly. In the assembly process, a robot can manipulate high-payload and repetitive operations, since these actions are leading causes of musculoskeletal disorders for human operators. Meanwhile, a human operator is essential to be involved and performs dexterous operations such as screwing and organising wires. The HRC workspace setup for car engine assembly is presented in Fig. 5, where an ABB robot, a UR robot, and a human operator are located around partial engine parts and tools in close proximity. The human operator receives on-site information and procedural guidance via HoloLens2, while robot controllers obtain transmitted instructions through Ethernet. Parts and tools in the assembly process consist of a valve cover, a cylinder head, four ignition coils, a fuel injector tube, several screws, a screwdriver, and a hex hey. An Azure Kinect is used to capture RGB-D visual data at 30 Hz as input for scenario parsing and task planning generation.

The assembly process consists of three subtasks, as shown in Fig. 6. Accordingly, temporal subgraphs can be leveraged to explain operation arrangements in these subtasks. An action entity linked to a robot entity denotes robotic motion types, such as moving or grasping; meanwhile, the subsequent pointing from the action node to a component node

Fig. 5. HRC workspace setup for car engine assembly.

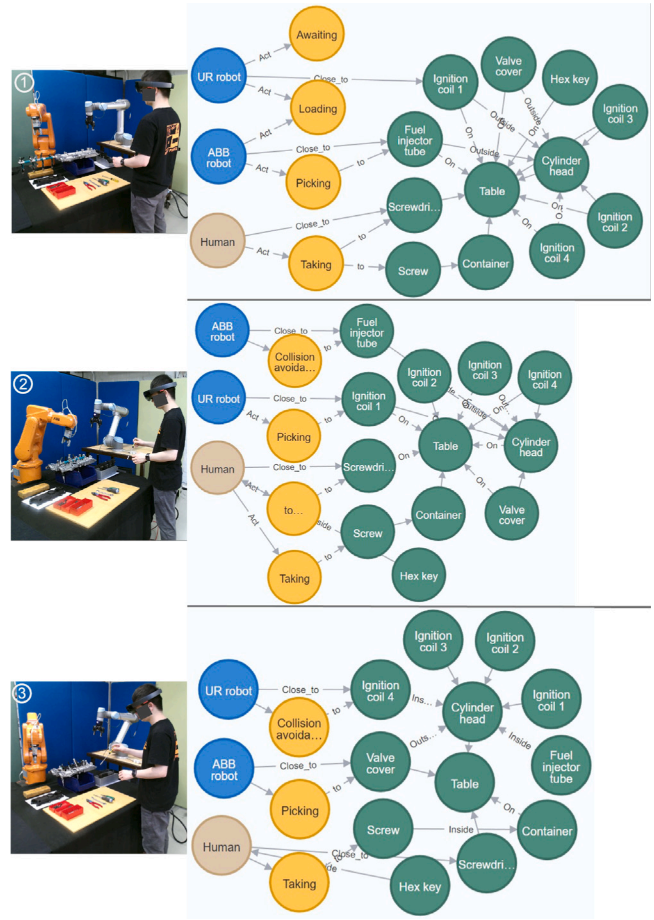

Fig. 6. Temporal subgraph explanation for assembly sequences.

gives the endpoint of the robot trajectory. A robot can plan and execute continuous manipulations following the graphic structure. Similarly, the connection from a human entity to an action entity and followed component nodes gives operation instructions to the human. The node pairs between different component entities describe their geometrical relations in workspaces.

For task fulfilment, the first subtask aims to insert the fuel injector tube into the cylinder head and fasten screws for these parts. A temporal subgraph evolving with four timestamps is utilised to represent assembly operation sequences, which are denoted as following:

(1) A robot node links to the *picking* entity which includes moving and grasping atomic actions, followed by pointing to the component entity of the *fuel injector tube*. A human node connects to the *taking* entity, which sequentially points to the *screw* entity and the *screwdriver* entity.
(2) A robot node links to the *transporting* entity which includes applying a force and moving atomic actions, followed by pointing to the *cylinder head* entity.
(3) A robot node links to the *aligning* entity which includes rotating and moving atomic actions, followed by pointing to the *cylinder head* entity. This robot manipulation aligns four pipe pins of the fuel injector tube to the holes of the cylinder head.
(4) A robot node links to the *placing* entity which includes releasing gripper and moving backward atomic actions, followed by pointing to the entity of the *fuel injector tube*. A human node connects to the *screwing* entity, and then points to the *fuel injector tube* entity.

The second subtask inserts four ignition coils into the cylinder head, followed by screw mounting operations. Assembly sequences in this stage are: ❶ a robot picks up an ignition coil, and the other robot avoids collision with the fuel injector tube while the human takes screws and changes tools to the hex key; ❷ The robot transports the ignition coil to the cylinder head; and ❸ the robot places the ignition coil into a hole in the cylinder head, whereas the human installs the ignition coil via screws. The subtask process can be depicted as an aforementioned temporal subgraph across three timestamps. The *collision avoidance* entity in the first timestamp controls a robot moving backward and keeping a distance from the target object. In this way, four ignition coils are separately mounted on the cylinder head.

The last subtask assembles the valve cover on the cylinder head, which is described as follows: ❶ two robots pick up the valve cover and perform collision avoidance from the ignition coils, respectively, while the human takes screws; ❷ the robot transports the valve cover above the cylinder head; ❸ the robot performs the holding action, whereas the human guides the robot to align bolt holes between the valve cover and the cylinder head; and ❹ the robot places the valve cover while the human fixes it on the cylinder head via screws. Similarly, these sequential operations can be transmitted to temporal subgraph-based graphic structures with four timestamps. In the third step, the *holding* entity means a robot switches to compliance mode for cooperation with human(s).

4.2. Self-organising task allocation result

Based on the three stages of car engine assembly, an Assembly Sequence Dataset (ASD) containing 2202 RGB-D images is developed. Among them, 1531 images and their labels are used as training data, while the remaining data is the test set. In a specific timestamp and subtask of ASD, various images are captured from different geometrical locations of components. For example, the fuel injector tube may be close to the ABB robot in one image, while it may locate beside the UR robot in other images. Images in each timestamp contain four different geometrical relations between components. It means different HRC task allocation strategies are required for changing situations and embodied in the annotation. Thus, the dataset can be utilised to test self-organising task allocation results for multiple human and robot teamwork.

The label for each image in the ASD contains relationships between human–robot-component-action nodes for KG construction and task allocation. For the annotation of node attributes, each component entity is recorded with a tuple of (X, Y, W, H, L), of which the former four elements are coordinates of a bounding box and the last one is the part category. The human node is annotated with attributes of

Table 1
Results of link prediction on the first subtask of car engine assembly. Evaluation metrics are MRR (%) and Hits@1/3 (%).

Subtask1	Timestamp 1			Timestamp 2			Timestamp 3			Timestamp 4		
Model	MRR	Hits@1	Hits@3	MRR	Hits@1	Hits@3	MRR	Hits@1	Hits@3	MRR	Hits@1	Hits@3
GCN	89.10	82.51	90.43	87.79	81.96	86.89	88.01	81.15	87.70	88.99	82.79	89.75
GAT	92.46	86.89	95.08	91.47	86.88	91.81	91.15	86.89	89.34	91.41	86.48	92.21
RGAT	93.47	88.52	96.45	92.09	87.70	92.62	91.01	86.89	89.34	91.94	87.30	92.62
Ours	94.67	90.44	97.27	95.28	92.63	95.08	95.04	92.62	95.08	95.55	92.62	96.72

Table 2
Results of link prediction on the second subtask of car engine assembly. Evaluation metrics are MRR (%) and Hits@1/3 (%).

Subtask2	Timestamp 1			Timestamp 2			Timestamp 3		
Model	MRR	Hits@1	Hits@3	MRR	Hits@1	Hits@3	MRR	Hits@1	Hits@3
GCN	88.34	82.17	87.91	88.18	80.33	90.16	88.51	81.97	88.52
GAT	90.95	86.27	90.57	90.57	84.42	91.80	91.04	86.07	91.39
RGAT	92.61	88.73	92.62	92.90	88.52	93.44	91.96	87.30	92.62
Ours	94.39	90.98	94.88	95.70	92.62	96.72	94.84	91.80	95.49

Table 3
Results of link prediction on the third subtask of car engine assembly. Evaluation metrics are MRR (%) and Hits@1/3 (%).

Subtask3	Timestamp 1			Timestamp 2			Timestamp 3			Timestamp 4		
Model	MRR	Hits@1	Hits@3	MRR	Hits@1	Hits@3	MRR	Hits@1	Hits@3	MRR	Hits@1	Hits@3
GCN	88.92	82.51	88.52	87.99	80.33	88.52	90.05	83.61	90.98	88.36	80.73	90.16
GAT	91.18	85.79	91.80	90.71	84.42	91.80	92.00	86.89	92.62	91.84	86.48	93.85
RGAT	92.21	87.70	92.08	92.26	87.70	93.44	93.23	88.93	93.44	93.06	88.11	95.49
Ours	93.88	90.16	94.54	94.88	91.80	94.26	95.70	92.65	96.73	95.90	93.03	97.13

(N, P, R, L), which means load capacity, reachability, repeatability, and node category. The robot node has the attribute labels of (N, P, R, F, L), where F is the freedom value. In detail, the maximum payloads of human, ABB robot, and UR robot are defined as *2* kg, *4* kg, and *5* kg, respectively. The reachability of humans is defined as the minimum distance from human hands to target objects. Accordingly, robot reachability is calculated by the distance from the robot gripper to the manipulated component. As the assembly task can be completed within ten steps, the repeatability of human and robot agents is defined as the same truth value of 1. The freedom of the ABB robot and UR robot are both 6 DoF.

In the training process for the temporal subgraph model, the agent subregion is composed of human and robot nodes. The sets of component nodes and action nodes stand for the task subregion and operation subregion, respectively. The number of hidden representation layers is set to 3. The model is trained with a learning rate of 0.0002 on a Tesla V100 GPU (16 G), which tackle the data in 32 batch size. The Stochastic Gradient Descent (SGD) is leveraged as the optimiser. For TSGA layer update, the hyperparameter β and parameter K are set to 0.5 and 3, respectively. Besides, the edge with a contribution score over 0.6 threshold value is retained in the pruning stage. In the testing stage, the SG constructed by the Graph R-CNN model is the prior fact for the first timestamp of each assembly subtask. Based on this observed fact, the trained temporal subgraph model is applied to complement the entire KG for HRC task allocation, which forecast correct action nodes linked to human and robot nodes, and subsequent target component nodes. For the following timestamps in an assembly subtask, the above completion graph is the prior fact. Then, the HRC KG is temporally updated with the trained model learning node embeddings from the timely observed scene graph and the prior fact.

Widespread measure metrics, including Mean Reciprocal Ranking (MRR), Hits@1, and Hits@3, are introduced to compare the performance of the proposed temporal subgraph model with other benchmarks. The evaluation process considers triples from agents to action nodes and from actions to component nodes in all testing images. The varieties of geometrical locations and time-changing relations between components in the testing data are suitable to verify the self-organising task allocation performance of the proposed model, which timely forecasts human–robot sequential actions. For three subtasks in car engine assembly, the comparison results between ours and Graph Neural Network (GCN), GAT [33], and RGAT [34] are presented in Tables 1, 2, and 3, respectively. The quantitative results show that the proposed model significantly improves link prediction accuracy for humans, robots, and component nodes. It means the proposed model can extract the implicit information from subregions and temporal procedures. In three assembly subtasks, subsequent timestamps' accuracy is higher than the performance in the first timestamp, as the proposed model can observe more prior facts with the task progressing.

Table 4
Ablation study results on three subtasks. Evaluation metrics are MRR (%).

Model	Subtask 1	Subtask 2	Subtask 3
Proposed (K = 3)	95.06	94.71	94.97
Proposed (K = 5)	93.35	92.94	93.72
Proposed (K = 1)	89.84	90.36	89.67
w/o subgraph sampling	92.13	92.86	92.45
w/o temporal sampling	91.68	92.38	91.94

To further evaluate the effectiveness of the method, an ablation study is conducted in Table 4. As most nodes in the graph have more than one edge, the reduction of graph pruning thresholds will dramatically damage the model performance. Besides, the link prediction accuracy of the proposed model is dropped when individually removing the subgraph sampling and temporal sampling process, which proves the positive effect of these mechanisms.

5. Discussions

Self-organising HRC driven by temporal subgraph allows multiple human and robotic agents to obtain dynamic task planning strategies from task structure understanding and geometric interpretation of workspaces. From the case study, HRC KG is a graphical explanation for operation sequences in different task stages. An action node indicates atomic motion types for robots, and a component node supplies the motion trajectory endpoint. Hence, a robot can execute automatic manipulation with instructions from KG. Meanwhile, human operators can know the changing relationships of objects in the workspace and receive appropriate operation suggestions from time-updated KG.

Although the experimental setup only involves one human and two robots, the HRC KG shows excellent promise to represent task allocation for more agents by expanding its graphical structure.

Despite those advantages, more specific node attributes need further consideration for self-organising HRC. For example, task allocation for a complex manufacturing task, including heavy parts in unstructured spaces, should give attention to human and robot agents' reachability, repeatability, and load capacity. The temporal subgraphs constructed in the case study should add more detailed attributes for action and component nodes in realistic self-organising HRC implementation. For instance, if there are 3D poses and geometric information in component nodes, a robot can obtain the locations of different holes in the cylinder head as end motion points.

6. Conclusions

Self-organising HRC is a foreseeable manufacturing paradigm in which multiple human and robotic agents perform operations qualified for their capabilities to accomplish complex tasks. This article proposed a temporal subgraph method to infer human–robot-action-component relationships and construct HRC KG for self-organising task allocation. To summarise, the principal scientific contributions of this research lies in two aspects.

(1) The temporal subgraph model forecasts human and robot agents' next operations by distilling knowledge representations from subregions in graphs and observed prior records. The self-organising task planning between multiple humans and robots is temporally updated to adapt to changes in task stages and spatial locations of components.
(2) The tri-layer HRC KG is proposed to serve as an explainable representation of operation arrangements between humans and robots. Robots can directly obtain motion types and destinations from various node pairs through the intuitive graphic structure, whereas humans can seamlessly learn about robot manipulation goals and manual operation suggestions.

Besides, comparative experiments are conducted to validate the performance of the proposed method on a car engine assembly task. The accuracy of our method shows a noticeable improvement in several evaluation criteria. Apart from these achievements, some potential future research directions are also highlighted here: (1) extending comprehensive node attributes and leveraging them to trigger ergonomics and self-organising task allocation strategies; (2) leveraging collaborative intelligence to tackle various industrial uncertainties in HRC scenarios; and (3) developing collision avoidance algorithms between multiple robot trajectories for real-world implementation of self-organising HRC.

Declaration of competing interest

The authors declare that they have no known competing financial interests or personal relationships that could have appeared to influence the work reported in this paper.

Acknowledgements

The work described in this paper was partially supported by the grants from the National Natural Science Foundation of China (No. 52005424), Research Grants Council of the Hong Kong Special Administrative Region (Project No. PolyU 15210222), Endowed Young Scholar in Smart Robotics (Project No. 1-84CA), Research Committee of The Hong Kong Polytechnic University under Research Student Attachment Programme 2021/22 and Collaborative Departmental General Research Fund (G-UAMS) from the Hong Kong Polytechnic University, Hong Kong Special Administrative Region, China.

References

[1] Wang L. A futuristic perspective on human-centric assembly. J Manuf Syst 2022;62:199–201.
[2] Lu Y, Zheng H, Chand S, Xia W, Liu Z, Xu X, et al. Outlook on human-centric manufacturing towards industry 5.0. J Manuf Syst 2022;62:612–27.
[3] Zheng P, Xia L, Li C, Li X, Liu B. Towards self-X cognitive manufacturing network: An industrial knowledge graph-based multi-agent reinforcement learning approach. J Manuf Syst 2021;61:16–26.
[4] Wang L, Gao R, Váncza J, Krüger J, Wang XV, Makris S, et al. Symbiotic human-robot collaborative assembly. CIRP Ann 2019;68(2):701–26.
[5] Li S, Zheng P, Liu S, Wang Z, Wang XV, Wang L. Proactive human–robot collaboration: Mutual-cognitive, predictable, and self-organising perspectives. Robot Comput-Integr Manuf 2023;81:102510.
[6] Liu H, Wang L. Collision-free human-robot collaboration based on context awareness. Robot Comput-Integr Manuf 2021;67:101997.
[7] Li S, Zheng P, Fan J, Wang L. Toward proactive human–robot collaborative assembly: A multimodal transfer-learning-enabled action prediction approach. IEEE Trans Ind Electron 2021;69(8):8579–88.
[8] Zheng P, Li S, Xia L, Wang L, Nassehi A. A visual reasoning-based approach for mutual-cognitive human-robot collaboration. CIRP Ann 2022.
[9] Liu X, Zheng L, Shuai J, Zhang R, Li Y. Data-driven and AR assisted intelligent collaborative assembly system for large-scale complex products. Proc CIRP 2020;93:1049–54.
[10] Li S, Wang R, Zheng P, Wang L. Towards proactive human–robot collaboration: A foreseeable cognitive manufacturing paradigm. J Manuf Syst 2021;60:547–52.
[11] Li S, Zheng P, Wang Z, Fan J, Wang L. Dynamic scene graph for mutual-cognition generation in proactive human-robot collaboration. Proc CIRP 2022;107:943–8.
[12] Wang P, Liu H, Wang L, Gao RX. Deep learning-based human motion recognition for predictive context-aware human-robot collaboration. CIRP Ann 2018;67(1):17–20.
[13] Xie H-L, Wang Q-H, Ong S, Li J-R, Chi Z-P. Adaptive human-robot collaboration for robotic grinding of complex workpieces. CIRP Ann 2022.
[14] Hietanen A, Pieters R, Lanz M, Latokartano J, Kämäräinen J-K. AR-based interaction for human-robot collaborative manufacturing. Robot Comput-Integr Manuf 2020;63:101891.
[15] Kana S, Tee K-P, Campolo D. Human–robot co-manipulation during surface tooling: A general framework based on impedance control, haptic rendering and discrete geometry. Robot Comput-Integr Manuf 2021;67:102033.
[16] Buerkle A, Bamber T, Lohse N, Ferreira P. Feasibility of detecting potential emergencies in symbiotic human-robot collaboration with a mobile EEG. Robot Comput-Integr Manuf 2021;72:102179.
[17] Fan J, Zheng P, Li S. Vision-based holistic scene understanding towards proactive human–robot collaboration. Robot Comput-Integr Manuf 2022;75:102304.
[18] Zeng Z, Röfer A, Jenkins OC. Semantic linking maps for active visual object search. In: 2020 ieee international conference on robotics and automation. ICRA, IEEE; 2020, p. 1984–90.
[19] Xia L, Zheng P, Li X, Gao RX, Wang L. Toward cognitive predictive maintenance: A survey of graph-based approaches. J Manuf Syst 2022;64:107–20.
[20] Alsentzer E, Finlayson S, Li M, Zitnik M. Subgraph neural networks. Adv Neural Inf Process Syst 2020;33:8017–29.
[21] Han Z, Chen P, Ma Y, Tresp V. Explainable subgraph reasoning for forecasting on temporal knowledge graphs. In: International conference on learning representations. 2021.
[22] Wang Z, Chen C-H, Zheng P, Li X, Khoo LP. A graph-based context-aware requirement elicitation approach in smart product-service systems. Int J Prod Res 2021;59(2):635–51.
[23] Yang J, Lu J, Lee S, Batra D, Parikh D. Graph r-cnn for scene graph generation. In: Proceedings of the european conference on computer vision. ECCV, 2018, p. 670–85.
[24] Sun X, Zhang R, Liu S, Lv Q, Bao J, Li J. A digital twin-driven human–robot collaborative assembly-commissioning method for complex products. Int J Adv Manuf Technol 2022;118(9):3389–402.
[25] Zhou B, Bao J, Chen Z, Liu Y. KGAssembly: Knowledge graph-driven assembly process generation and evaluation for complex components. Int J Comput Integr Manuf 2021;1–21.
[26] Lamon E, De Franco A, Peternel L, Ajoudani A. A capability-aware role allocation approach to industrial assembly tasks. IEEE Robot Autom Lett 2019;4(4):3378–85.
[27] Antakli A, Spieldenner T, Rubinstein D, Spieldenner D, Herrmann E, Sprenger J, et al. Agent-based web supported simulation of human-robot collaboration. In: WEBIST. 2019, p. 88–99.
[28] Patnayak C, McClure JE, Williams RK. WASP: A wearable super-computing platform for distributed intelligence in multi-agent systems. In: 2021 IEEE high performance extreme computing conference. HPEC, IEEE; 2021, p. 1–7.
[29] Galin R, Meshcheryakov R, Kamesheva S. Distributing tasks in multi-agent robotic system for human-robot interaction applications. In: International conference on interactive collaborative robotics. Springer; 2020, p. 99–106.

[30] Liau YY, Ryu K. Genetic algorithm-based task allocation in multiple modes of human–robot collaboration systems with two cobots. Int J Adv Manuf Technol 2022;119(11):7291–309.

[31] Karami H, Darvish K, Mastrogiovanni F. A task allocation approach for human-robot collaboration in product defects inspection scenarios. In: 2020 29th IEEE international conference on robot and human interactive communication. IEEE; 2020, p. 1127–34.

[32] Li C, Zheng P, Li S, Pang Y, Lee CK. AR-assisted digital twin-enabled robot collaborative manufacturing system with human-in-the-loop. Robot Comput-Integr Manuf 2022;76:102321.

[33] Veličković P, Cucurull G, Casanova A, Romero A, Lio P, Bengio Y. Graph attention networks. 2017, arXiv preprint arXiv:1710.10903.

[34] Busbridge D, Sherburn D, Cavallo P, Hammerla NY. Relational graph attention networks. 2019, arXiv preprint arXiv:1904.05811.