Vol. 16 (2026): Proceedings of the 2026 International Conference on AI and Technological Innovation Applications (AITIA 2026)
E-ISSN: 3105-0522 P-ISSN: 3105-0514
DOI: https://doi.org/10.70267/aitia.2026102118

A Review of Multimodal Perception and Closed-Loop Task Execution for Human-Robot Collaboration Robots Oriented Toward Intelligent Assembly

Zhuoyuan Li*

Faculty of Mechanical and Electrical Engineering, Kunming University of Science and Technology, Kunming 650500, China

**Corresponding author: Zhuoyuan Li.*

Abstract

With the increasing demand for flexible manufacturing, especially for multi-variety and small-batch production, human-robot collaboration has become more widely used in intelligent assembly. Many studies have investigated multimodal perception, task allocation, safety control, and quality inspection, and considerable progress has been made in these areas. Even so, several problems are still difficult to address. Information exchange between different technical modules is often not well connected, task understanding is still limited, and feedback from the assembly process is not fully incorporated into decision making. This review discusses recent research from the perspective of a complete "perception-decision-execution-feedback" process. The main topics include operator action and intention recognition, perception of part states, task allocation under human-related constraints, safe trajectory planning, and quality feedback. Rather than considering these techniques separately, attention is given to how they interact during collaborative assembly. Current studies suggest that the main challenge is no longer improving the accuracy of a single perception or control module. A more important issue is how to connect multi-source perception, task understanding, safety constraints, and quality feedback into a stable closed-loop framework. Future work is expected to pay more attention to cross-station adaptation with limited training samples, practical and verifiable use of vision-language-action models, digital twin-based safety validation, and skill updating based on assembly deviations. It is hoped that this review can provide useful ideas for future research and system development in intelligent human-robot collaborative assembly.

Keywords

human-robot collaboration robots, intelligent assembly, multimodal perception, task allocation, quality feedback, digital twin, embodied intelligence

1. Introduction

The assembly stage is still one of the hardest parts to automate in modern manufacturing. Compared with relatively structured processes such as machining and welding, assembly usually involves greater uncertainty. Parts may vary from one product to another, production processes often need to be adjusted, dimensional errors can accumulate during assembly, and flexible components are difficult to model accurately because their deformation is not always predictable. These problems have become even more obvious as manufacturers increasingly adopt high-mix, low-volume (HMLV) production and customized products, where production lines need to be reconfigured more frequently. Under these conditions, conventional automation based on fixed rules is often not flexible enough, and in many cases it is also expensive to maintain. Human-robot collaboration offers another option by combining the adaptability of human operators with the consistency of robotic systems, and has therefore attracted growing attention in intelligent assembly [1]. At the same time, the move toward more human-centered manufacturing means that collaborative robots are expected to do more than execute predefined motions. They also need better abilities in intention recognition, situation awareness, and coordinated interaction with human workers [2,3].

The role of collaborative robots in intelligent assembly has also changed over time. Instead of simply assisting workers, they are now expected to complete many key operations together with humans. Tasks such as inserting flexible parts, precision fastening, or correcting assembly errors still depend heavily on touch, experience, and quick judgment, which remain difficult for conventional industrial robots. Relying only on manual work is not an ideal solution either, since long working hours can increase fatigue, cause fluctuations in production rhythm, and raise the risk of work-related injuries. For this reason, intelligent assembly is generally viewed as a process of sharing work between humans and robots rather than replacing one with the other. Robots are more suitable for repetitive and high-precision operations, while human operators are still better at handling unexpected situations, checking assembly quality, and making decisions when conditions change. However, this type of collaboration also places higher demands on robotic systems. In semi-structured assembly environments, robots need to recognize operator intentions, understand the current state of parts, and respond without compromising safety or production efficiency. Although many perception and decision-making methods have been proposed, they are still not fully able to meet these practical requirements [4–6]. In other words, improving individual perception or control modules alone is usually not enough. The connection between perception, task allocation, execution, and safety remains a major issue in intelligent assembly.

Research on intelligent human-robot collaborative assembly has gradually developed around four closely related aspects: multimodal perception, collaborative decision-making, task execution, and quality feedback. Early perception methods mainly relied on visual information, whereas more recent studies tend to combine multiple sources of information, including skeletal motion, image features, and process-related text, to obtain richer descriptions of assembly activities. Task allocation has also changed from fixed offline planning to more adaptive strategies that consider operator fatigue, ergonomic conditions, and changes in human trust. In task execution, attention has shifted from simple replay of predefined motions toward demonstration learning, force-position hybrid control, and safer motion planning. Quality assessment is also changing. Instead of only deciding whether a product passes inspection, many recent methods attempt to identify geometric deviations and use this information to improve later assembly operations. More recently, vision-language models, digital twins, and embodied intelligence have opened up new possibilities for assembly understanding and robot control. Even so, their performance in real production environments, especially in terms of reliability, real-time response, and safety, still needs further validation before large-scale industrial deployment [7,8].

A considerable amount of work has been published on human-robot collaborative assembly in recent years. Even so, reviews that specifically discuss intelligent assembly from a system perspective are still relatively limited. Most existing reviews concentrate on one topic, such as collaborative robot applications, human-robot safety, or perception methods, while the complete assembly process is discussed much less often. The links between different stages of the workflow are also not always clearly explained. Multimodal perception, task allocation, adaptive execution, and quality feedback are usually introduced as separate topics, although in practice they continuously influence one another. As a result, the relationships between different technical modules, including how information is transferred, where constraints are introduced, and under what conditions failures occur, are often left unclear [9]. In addition, new technologies such as vision-language models, digital twins, and embodied intelligence are developing quickly. Their potential has attracted considerable attention, but how they can be combined with existing perception, planning, and control methods is still an open question. Because of these gaps, it is still difficult to explain why improving the performance of an individual module does not always result in better collaboration at the system level.

This review is organized around the information flow of "perception-decision-execution-feedback", with multimodal perception and task execution serving as the two main threads. Instead of introducing studies only according to technical categories, the discussion follows the assembly process itself. More attention is given to how different methods are connected, what information they require from previous stages, and what limitations they may introduce to later stages. The intention is not simply to compare algorithms, but to show how perception, task allocation, and execution depend on one another during collaborative assembly.

The literature discussed in this review mainly comes from representative studies published over the last five years in journals including *Robotics and Computer-Integrated Manufacturing*, *Journal of Manufacturing Systems*, *Journal of Intelligent Manufacturing*, *Advanced Engineering Informatics*, and *Computers & Industrial Engineering*, together with related conference papers. Several earlier studies that have had an important influence on collaborative assembly, task allocation, and industrial safety are also included where necessary. The literature search covers topics such as human-robot collaboration, collaborative robots, intelligent assembly, multimodal perception, task allocation, quality feedback, digital twins, and embodied intelligence. This is intended as a thematic review. The main purpose is to summarize the technical development of the field, discuss how different components are connected, and point out research gaps, rather than provide a complete bibliometric survey.

Three points are emphasized in this review. The first is a closed-loop framework that connects multimodal perception, collaborative decision-making, task execution, and quality feedback within the same information flow. The second is a discussion of why better perception accuracy does not necessarily improve the performance of the whole collaborative system, with particular attention paid to semantic representation, information transfer, and feedback updating between different stages. The third is a summary of several research directions that may become more important for practical deployment, including cross-scenario adaptation, task semantic representation, safety-aware planning, and closed-loop quality feedback.

2. Intelligent Assembly Human-Robot Collaboration Robot System Framework

2.1 Characteristics of Intelligent Assembly Tasks

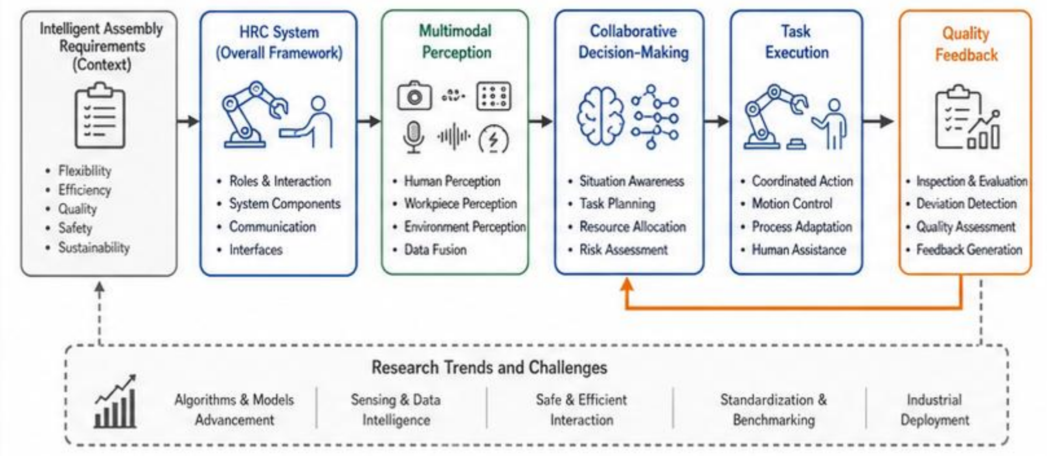
Intelligent assembly places different requirements on a human-robot collaborative system from many other manufacturing processes, and these requirements largely influence how the system is designed. One of the main reasons is that assembly combines precise physical operations with continuous on-site decision-making under conditions that are often uncertain. The widespread use of high-mix, low-volume (HMLV) production has made flexibility even more important [2]. In practice, several difficulties usually appear at the same time. Production lines often need to be adjusted for different product types. Assembly operations such as shaft-hole insertion and flexible part installation require high precision, but the working conditions are not always predictable. In addition, manufacturing errors accumulate during production, so the starting conditions are rarely exactly the same from one product to the next. Because of this, assembly systems need to rely on real-time perception and online adjustment to maintain stable product quality [10].

For these reasons, intelligent assembly cannot depend on a single algorithm or one individual technology. A more practical solution is to distribute work according to the strengths of humans and robots. Robots are generally responsible for repetitive and standardized operations, while human operators remain important for dealing with unexpected situations, making judgments, and confirming assembly quality. Collaboration becomes effective only when the adaptability of human workers and the stable execution of robots can support each other throughout the assembly process [11].

Considering these characteristics, the following discussion uses the requirements of intelligent assembly as the starting point and examines the human-robot collaborative system through a unified framework. The framework connects multimodal perception, collaborative decision-making, task execution, and quality feedback instead of treating them as separate parts. In other words, these functions are viewed as different stages of the same information flow, where the output of one stage becomes the input for the next.

This framework is intended to show how different parts of the collaborative system are related to one another. Assembly requirements, system configuration, perception, task planning, physical execution, and quality evaluation are all considered within the same process rather than discussed independently. From this perspective, improving only a sensor, an algorithm, or a controller is unlikely to solve the main problems in intelligent assembly. More important is how information can move through the whole process, from task requirements and state perception to task assignment, execution, and quality feedback, so that the system can operate as a complete closed loop.

Figure 1: Intelligent Assembly Human-Robot Collaboration Research Framework



2.2 Composition of the Human-Robot Collaboration System

From the previous discussion, it becomes clear that the key issue is not only how individual components work, but how the system deals with uncertainty in a coordinated way. In most cases, a human-robot collaboration (HRC) system is built around the interaction between information perception and decision-making. Operators, robots, tasks, and the surrounding environment are all involved, and they are connected through continuous data exchange rather than isolated modules. Whether these parts can actually work together depends largely on whether the information between them can stay consistent during operation.

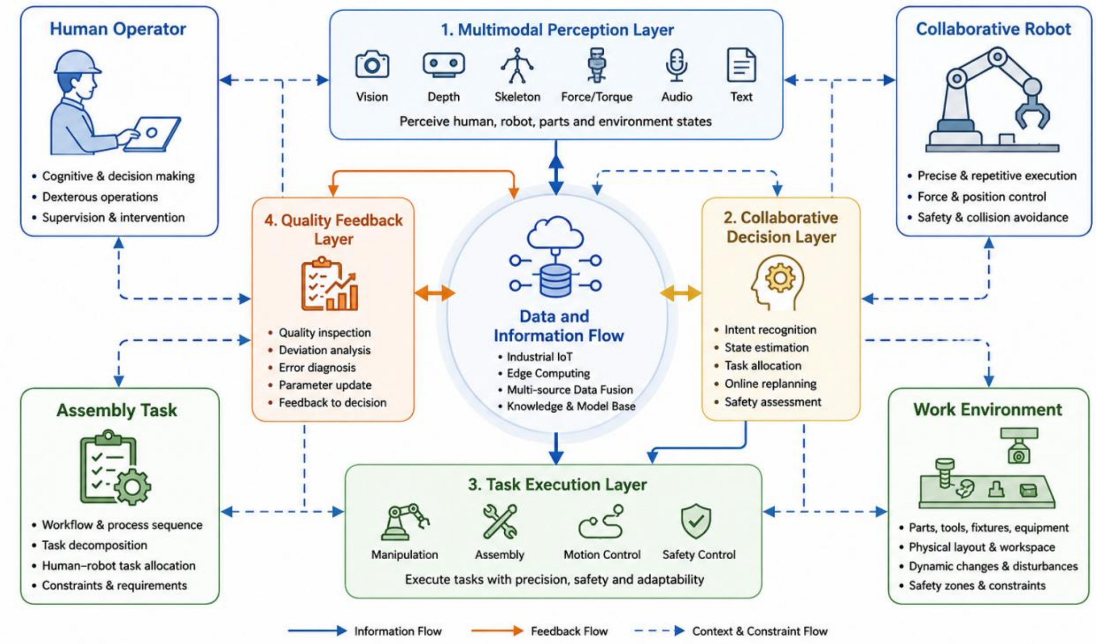
Table 1: Composition Elements of Human-Robot Collaboration Systems Oriented Toward Intelligent Assembly

Component	Function Description	Key Technologies
Operator	Provides high-level decision-making, cognitive flexibility, and dexterous operation capabilities; handles unstructured tasks and abnormal conditions.	Cognitive state assessment, multimodal interaction, action and intention recognition
Collaborative Robot	Undertakes standardized, high-precision, and highly repetitive physical output tasks; serves as the primary carrier of the execution layer.	Compliant control, force-position hybrid control, safe obstacle avoidance and trajectory planning, quality feedback
Assembly Task	Defines collaboration content and operational constraints; influences the formulation of perception, decision-making, and execution strategies.	Task complexity modeling, task process decomposition and sequence planning, human-robot task allocation strategies and algorithms
Operating Environment	Contains the physical layout of workpieces, tools, and shared space; its dynamic changes require real-time perception.	Workpiece identification and localization, scene reconstruction, obstacle detection
Data and Information Flow	Acts as the core enabling facility that permeates the system, enabling real-time transmission and processing of heterogeneous data.	Industrial Internet of Things, edge computing, spatiotemporal synchronization and fusion architecture

In practice, operator behavior, robot status, task requirements, and environmental conditions are not independent. They influence each other all the time. For example, operator actions and intentions are usually reflected at the perception level, but they are not always easy to describe in a fixed format. Task complexity and how tasks are assigned are more related to decision-making, while robot execution and quality checking are closer to the operation stage. The environment is present in almost every step, especially when conditions change during assembly. On top of this, data flow plays a supporting role, mainly for keeping different states synchronized and allowing information to move between modules when needed.

For intelligent assembly, one of the practical requirements is that the system should be able to align human intention, robot state, part position, and environmental changes in something close to real time. This information is then used for decision-making and later adjustments. In this sense, the system is less about having a clear layered architecture, and more about whether these layers can actually communicate without breaking down at the interfaces.

Figure 2: Intelligent Assembly Human-Robot Collaboration System Architecture and Information Flow Relationships



This structure further illustrates that the human-robot collaboration system is not a simple superposition of operators, robots, tasks, and environments. What truly determines system capability is whether data and information flows can be continuously transmitted among all entities: operators provide intentions and experiential judgments, robots undertake stable execution, assembly tasks provide processes and constraints, the operating environment supplies dynamic boundaries, and the data layer transforms these states into information usable for decision-making and feedback.

2.3 Perception–Decision–Execution–Feedback Closed-Loop Framework

Intelligent assembly with human-robot collaboration is often described as a kind of continuous loop, where information keeps moving rather than staying within fixed stages. In a simplified way, the process can be seen as involving task input, perception of states, decision-making, execution of actions, and feedback after quality checking. In real systems, these stages are not always strictly separated, but they still help to explain how the whole process is organized.

Task input usually means breaking a production goal into smaller assembly steps with certain constraints. After that, perception collects information about the operator, the parts, and the working environment, often using different sensing channels together. The decision process tries to combine current states with task requirements and then decides how work should be assigned or adjusted. Execution is where robots actually perform physical operations, and at the same time some process data is recorded. Feedback comes after that, mainly from checking quality through vision, force signals, or pose information, and the results are used to adjust later decisions or actions.

These stages are sometimes summarized as a “perception–decision–execution–feedback” loop [12]. The perception part is mainly about building a basic understanding of what is happening with humans, objects, and the environment. Decision-making is more about turning this information into task arrangements. Execution is about carrying out the physical work safely and accurately. Feedback closes the loop by showing whether the previous steps worked properly or not. In practice, this framework is not only used to describe the system structure, but also becomes a rough reference when different methods are compared in later sections.

For the discussion in the following parts of this paper, technologies are considered from a few practical angles rather than only their individual accuracy. One aspect is what role they play in the loop, such as recognizing states, understanding tasks, generating actions, or correcting deviations. Another aspect is what kind of information they receive and what they pass on to the next step. A third aspect is how they behave when things go wrong, for example when objects are occluded, when production lines change, or when unexpected disturbances appear. This kind of perspective helps avoid treating high accuracy in isolated experiments as equivalent to real industrial performance.

3. Multimodal Perception Technology

Multimodal perception is usually considered the starting point for giving human-robot collaboration systems some ability to understand what is happening in intelligent assembly. In this section, the discussion is divided into three parts: operator action and intention recognition, part state and assembly quality perception, and finally multi-source fusion together with cross-scenario generalization.

3.1 Operator Action and Intention Recognition

To understand operators in assembly tasks, the first step is to recognize what they are doing. Assembly actions are usually very fine-grained, short in duration, and strongly dependent on context. The same hand movement can mean different things depending on which part or process is involved. Because of this, using only a single camera view is often not stable enough to tell what is really happening on the shop floor. Earlier methods mostly relied on single-frame or short-video classification, but later work moved toward combining different types of information, such as skeleton sequences, RGB images, depth data, and sometimes even text descriptions. More attention has also been given to few-shot learning, especially for cases where new assembly tasks appear with very limited data [13,14].

In general, methods in this area can be roughly grouped into RGB-based approaches and skeleton-based approaches. RGB methods usually capture overall appearance and scene context better, while skeleton-based methods are less sensitive to lighting changes and background noise. But neither of them works well in all situations. RGB models often fail when tools or hands are blocked by other objects, while skeleton-based methods miss important interaction details between hands, parts, and tools. Because of this, combining multiple modalities has become more common. Different signals like skeleton data, RGB-D, and inertial sensors are usually fused through attention-based models or multi-stage fusion structures [12,15]. For instance, Matin et al. [16] proposed an RGB-D two-stream model (C3D-AssNet) and reported 98.84% accuracy on a gearbox assembly dataset, which is higher than the 92.28% obtained with RGB alone. This shows that multimodal fusion can work well under controlled experimental conditions. However, this kind of result is also quite dependent on how the dataset is built, including its size, labeling, and recording environment. Some studies also try to add process-related text information to help distinguish similar actions [17].

In order to reduce the cost of collecting labeled industrial data, few-shot learning methods have been used. These methods try to generalize from very few samples using metric learning or meta-learning frameworks. In practice, they can reduce the burden of data collection to some extent, but they still rely quite a lot on the similarity between training and testing environments. When workstation layout, lighting, or camera viewpoint changes a lot, performance usually drops, and it is still not at a level suitable for direct industrial deployment [13].

Even with these improvements, there are still some obvious limitations. Occlusion happens very often in real assembly scenes, especially self-occlusion of hands and occlusion caused by tools or parts. Many assembly actions are also symmetrical in movement, which makes fine-grained classification difficult. More importantly, most current methods still focus on recognizing what action is happening, but they do not really explain why the action is happening. Because of this, research has gradually started to move from pure action recognition toward intention reasoning.

Intention reasoning tries to answer questions like “why is the operator doing this” and “what will happen next.” In real assembly environments, actions such as picking a part or placing a tool may look almost the same, but they can correspond to completely different process states. So intention reasoning is not just classification, but more like a temporal reasoning problem that depends on task context and process order [18].

From the existing literature, there are roughly three directions that are being explored. At the data level, some studies try to include part state information directly into action sequences, so that the relationship between human motion and object state becomes clearer. At the interaction level, some work uses large language models to translate natural instructions into structured assembly steps, which helps connect language and action in a loose way. At the decision level, digital twin techniques are used to keep track of process status and context, and then support prediction or replanning [18–20].

It should be emphasized that these three directions are not a unified or mature framework. They are more like separate attempts that come from different research communities. There is still a long way to go before they can be combined into a stable and deployable intention reasoning system in real factories.

3.2 Part State and Assembly Quality Perception

For assembly objects, the perception system actually has to deal with both people and physical parts at the same time. Part category, spatial pose, process status, and assembly quality all affect whether the robot can act correctly during operation. If one of these pieces of information is missing or wrong, the whole coordination process can become unstable, although in real systems this does not always fail in a clean or obvious way.

These four types of information are often related to each other in practice, but in most existing studies they are still treated separately. In part recognition, some work modifies YOLOv7 with attention modules or improved loss functions, mainly trying to balance speed and accuracy. For pose estimation, a common idea is to first narrow down the possible region and then solve the 6D pose, which seems to work reasonably well in cluttered or stacked situations. Process state recognition usually tries to infer what step the operator is currently doing based on hand motion sequences, and then uses that as a signal for when the robot should assist. For quality perception, many studies no longer rely only on pass/fail results. Instead, they compare 3D point clouds with CAD models to estimate deviation in a more continuous way [21–23]. Each of these directions works to some extent, but they are still mostly tested in separate setups, not really in a combined system.

If we look at it from a more practical angle, these functions are actually linked. Part recognition and pose estimation are mainly used to support grasping and alignment. Process state recognition is more about deciding when the robot should intervene. Quality feedback is closer to what happens after execution, and it can be used to adjust later actions if something goes wrong. In some cases, deviation information is also fed back into learning-based controllers so that the robot can slowly adjust its behavior across batches or mixed production lines. This forms something like a “detect–correct–adjust” loop, although in real applications it is not always as stable as it sounds.

Together with intention understanding, these perception functions provide a basic foundation for more adaptive assembly behavior. It should also be said that this kind of loop description is more of an analytical summary based on existing studies, and it is not a fully integrated system that has been realized in practice.

3.3 Multi-Source Fusion and Cross-Scenario Generalization

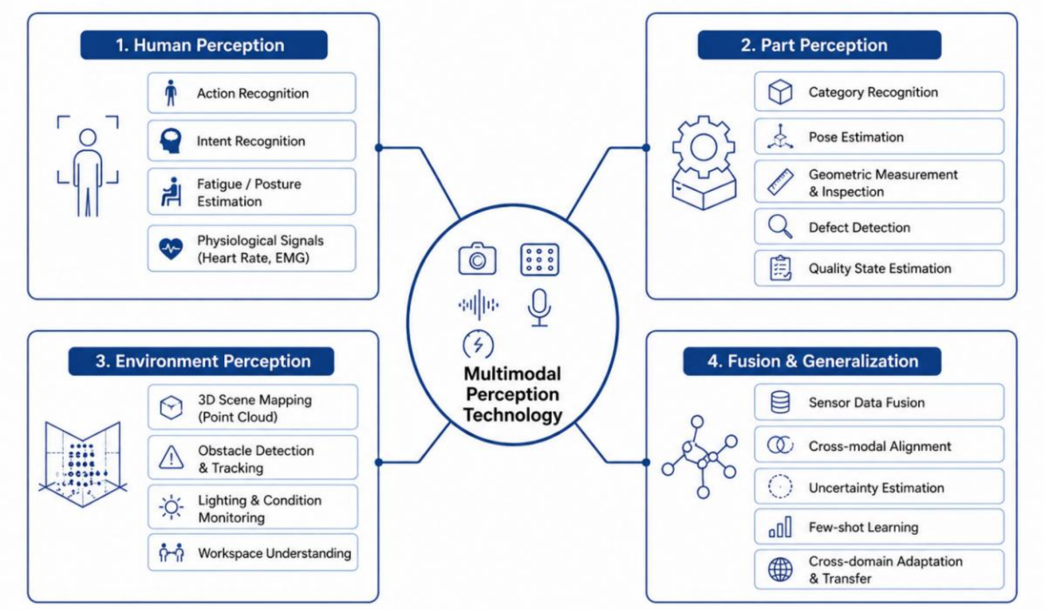
In real assembly lines, it is hard for any single sensor to cover everything that is going on at the same time. Operators, parts, and the surrounding environment all change continuously, so perception systems usually need multiple types of signals together. Each modality gives a different kind of information, but none of them is complete on its own.

RGB cameras can capture appearance and context, but they are easily affected by lighting and shadows. Depth images and point clouds describe geometry better, but in industrial scenes they often become noisy, especially when there are shiny metal surfaces or stacked parts. Skeleton data is relatively stable under different lighting conditions, but it does not really show how hands interact with objects. Force and inertial sensors can capture fine physical interaction, but they are usually wearable, which limits how freely operators can move [15]. Text and language information can bring in process knowledge, but matching language with what is actually happening in the scene is still not straightforward.

Because of these limitations, different modalities are usually combined. Earlier work mostly used simple feature merging or voting at the decision level. Later, attention-based models were introduced to give different weights to different signals depending on the situation. More recent studies try to project different types of data into a shared representation space so that they can be compared or aligned more directly. For example, Gao et al. [17] reported that when skeleton data was used alone, accuracy was 87.6%, but after adding RGB information and text features, it increased to 98.4%. This suggests that combining modalities can help, although the improvement also depends a lot on how controlled the experimental setup is.

At the same time, few-shot learning has been used to reduce the need for large labeled datasets. In practice, it assumes that the new environment is not too different from the training environment. When the layout, camera view, or background changes too much, performance usually drops. This is still one of the main problems for real deployment. Methods such as domain adaptation and data augmentation help to some extent, and some models try to learn features that are less sensitive to specific scenes, but the results are still not very stable across different workstations.

Figure 3: Multimodal Perception Technology Classification Framework



From a more simplified view, multimodal perception can be thought of as involving four types of elements: human behavior, part state, environment, and the fusion mechanism itself. These are not independent in practice, but they are often studied separately in the literature. What makes the problem difficult is that the final perception output is not just about recognizing objects or actions. It needs to reflect the actual process state in a way that can be used later for planning and execution.

In many cases, current systems still output relatively simple results such as action labels or object positions. This is usually not enough for task replanning in real assembly. A more useful form of output would include things like current process stage, operator intention (if it can be inferred), part condition, and possible risks in the operation. But turning raw sensor data into this kind of structured information is still not well solved.

So, rather than only asking what the system can see, a more practical question is how this visual and sensor information can be turned into something that is actually usable for decision-making and task control. This shift is becoming more important in recent work, especially for systems that aim to be used in real industrial environments.

4. Collaborative Decision-Making and Task Allocation Technology

Differences between humans and robots in physical strength, precision, stability, and cognitive flexibility make task allocation a decision-making problem that requires the integration of multi-source information. Unlike experience-based manual allocation, task allocation in intelligent assembly needs to answer three questions: who should perform the task, when it should be performed, and how to adjust it when working conditions change. This chapter expands on these questions across three levels: task classification, human-factor constraints, and dynamic replanning.

4.1 Assembly Task Classification and Human-Robot Capability Matching

Task allocation in human-robot assembly is mainly about dealing with differences between humans and robots. Humans are flexible and good at judgment, while robots are more stable in precision and repeatability, but less adaptable when things change. Because of this, assigning tasks is not just a planning problem, but also depends on information from different sources, including task conditions and human status. In practice, task allocation usually has to consider three basic issues: who does the task, when it is done, and what happens if conditions change during execution. These questions sound simple, but in real production they are not always easy to separate.

Most existing studies try to solve this by matching task characteristics with human or robot capabilities. One common idea is to describe both sides using several dimensions, such as safety, physical difficulty, cognitive demand, and cost or quality requirements. Some indicators are used in practice, for example ergonomic risk can be estimated with the NIOSH equation, visual difficulty can be linked to how easy it is for a recognition system to work, and cognitive load is sometimes measured using indices like CLAM [24]. These values are then combined in different ways to decide whether a task should go to a human, a robot, or both. In some cases, safety rules are set as hard constraints, so if a risk level is too high, the task is directly assigned to the robot without further calculation.

When a task is assigned to both human and robot, another issue appears: who leads and who follows. Vianello et al. [4] showed in experiments that a flexible mode, where humans and robots can switch roles depending on real-time conditions, works better than fixed role assignment in terms of workload balance and adaptability. But this result is still mostly based on controlled experimental setups, and it is not clear how well it works in more complex assembly environments.

More recent studies also try to separate “task complexity” from “task difficulty.” Complexity is usually related to the task itself, while difficulty depends more on who is doing it and under what conditions [25]. This means the same task can be easy for one operator but difficult for another, depending on experience or physical state. Because of this, task allocation cannot rely only on fixed task descriptions. It also needs to consider the condition of the operator at runtime.

Looking at production line balancing studies, human-robot collaborative assembly has been studied more in recent years, but many models still focus mainly on reducing cycle time. Issues like fatigue, workload balance, and safety are often included only in a simplified way or not fully optimized together [11].

Overall, most current methods still assume that both task features and human capability are fixed during execution. In real situations, this is not always true. People get tired, attention changes, and working conditions shift. Because of this, a more practical system should also allow updates during execution, instead of only making decisions at the beginning.

4.2 Task Allocation under Human-Factor Constraints

After deciding whether a task can be done by a human, a robot, or both, the next problem is how to handle multiple tasks together when safety and human conditions also matter. In real systems, safety is usually the first thing to consider. If there is a possible collision risk, the system will still prioritize operator safety, even if it affects efficiency [6]. This part is relatively strict and usually not negotiated.

Beyond safety, human-related factors are often divided into physical and psychological aspects, but in practice the boundary between them is not always clear. On the physical side, some studies use posture evaluation methods like REBA, and also try to detect fatigue through EEG signals, and then include these results in task planning so that workload can be adjusted during operation. On the psychological side, trust becomes an important but unstable factor. Operators may trust the robot more or less depending on how it performs, how often it interrupts the task, and whether it finishes work on time. This trust is sometimes modeled using Bayesian networks, mainly to give a numerical value that can be used in allocation decisions.

However, most of these approaches are still tested in relatively controlled environments. Fatigue detection systems, for example, may reach around 87.7% accuracy, but misjudgment still happens. Some multi-agent systems that rely on large language models also introduce extra delay and computational cost. Trust estimation is also often based on past data, rather than real-time interaction in changing conditions [5]. Petzoldt et al. [9]

also noted that although dynamic allocation is theoretically attractive, most real implementations still rely heavily on simple time-based rules, while factors like ergonomics or quality are not fully included.

Once human factors are added, task allocation is no longer only about efficiency. It becomes a trade-off problem between efficiency and human-related constraints. Some simulation studies on workload balancing show that keeping a fixed human-robot ratio may reduce overall throughput to some extent [26], but this does not always happen in every scenario. The result depends a lot on task type, system setup, and parameter choices, so it should not be treated as a general rule. In practice, different systems may choose very different balances depending on what they prioritize.

Even with these efforts, there are still some unresolved issues. One is that physiological load, trust, and task complexity are clearly related, but there is still no stable way to optimize them together in a single framework. Another issue is that safety constraints are often handled separately from task allocation. When a risk appears, systems usually react at the motion or trajectory level, instead of adjusting task decisions earlier [27]. In many cases, the decision is also based on a single moment of system state, which makes it hard to respond smoothly when conditions change during execution.

4.3 Dynamic Replanning and Intelligent Decision-Making

In real assembly environments, operators do not always follow a fixed procedure. Their actions can drift over time because of habits, small mistakes, or fatigue. As work continues, the execution time may also slowly deviate from the original plan. On top of that, problems such as missing tools, defective parts, or quality issues can suddenly make the original task plan no longer valid. When these deviations become too large, the system usually needs to adjust the plan during operation.

The difficulty in dynamic replanning is that the system has to react to changes, but not react too often. If every small deviation triggers a full replanning process, the system can become unstable and inefficient. He et al. [20] proposed an event-based strategy where replanning is only activated when there is a clear mismatch between expected and actual actions, or when the operator pauses for more than a few seconds. This works in controlled experiments such as reducer assembly, but the conditions for triggering are still quite simple and may not cover more complex situations.

In general, events that require replanning can be roughly grouped into three types: changes in operator behavior, changes in product quality, and external disruptions. The first type is relatively easier to observe, while the other two are still not well captured in many current systems.

How the system responds depends on what kind of deviation happens. If the operator skips a step or performs an extra action, the system can try to infer the current process state again and update the remaining task sequence. Before sending new instructions to the robot, the updated plan is usually checked in a digital twin environment to avoid obvious collisions or reachability problems. When quality problems appear, the system may not only adjust the remaining tasks but sometimes also needs to go back and correct previous steps or insert additional operations. In more serious cases, such as missing parts or broken supply, the whole task sequence may need to be recalculated.

At the same time, most current dynamic replanning methods still struggle with unstructured or unexpected events. Many triggering rules are still manually designed, and they do not always work well when disturbances become more complex, such as tool absence, part defects, or multi-person interference. In practice, these methods are often small adjustments to an existing plan rather than a full rethinking of the task. Their performance in real factory environments is also still unclear, especially under noisy sensing, delays, and safety-critical constraints.

Because of this, the main issue is not only how to compute a better allocation result at a single moment, but whether the system can keep updating its understanding of the task over time. Human state, safety constraints, and quality feedback need to be reflected continuously, otherwise the plan quickly becomes outdated.

So, rather than saying there is a final solution, it is more accurate to say that current systems are still trying to move from fixed planning toward something that can adjust itself during execution. This is still not stable enough in most real applications, but it points to the direction where the field is going.

5. Task Execution and Quality Feedback Technology

After the perception layer extracts operator actions, intentions, and scene information, and the decision-making layer converts them into an allocation scheme of "who–when–what," the scheme is implemented as physical robot actions. Deviation signals generated during execution need to be structured and fed back to update subsequent plans and process parameters. The following sections address this from three aspects: skill acquisition, safe execution, and closed-loop quality feedback.

5.1 Teaching-Learning and Skill Generalization

For robots, learning assembly skills from humans is usually done in two basic ways. One is offline programming, where specific programs are written for each product and each process. This works in stable production, but in real factories where products change often, the cost of rewriting and maintaining code becomes quite heavy. The other way is hand-guided teaching, where the operator physically moves the robot to demonstrate actions. During this process, the system records joint angles, end-effector motion, and gripper states, and then tries to learn patterns from several demonstrations instead of relying on a single example.

Some studies use Gaussian Mixture Models (GMM) to represent these demonstrated trajectories in a probabilistic form. The idea is that when the position of a workpiece changes, the robot can still generate a similar motion without repeating the teaching process. Zhang et al. [28] reported that with an improved GMM combined with Bayesian priors and some evaluation rules, the grasping success rate of a Baxter dual-arm robot increased from 76.8% to 89.3% in a multi-part assembly setting.

Still, this kind of improvement is mostly seen in relatively simple pick-and-place tasks. Once the task becomes more precise, such as inserting shafts into holes or tightening screws, trajectory-based models start to show clear limits. These operations depend not only on motion paths, but also on contact force, direction of force, and small tolerance ranges, which are difficult to capture from trajectories alone. Even in grasping tasks, many experiments are carried out under fairly controlled conditions with fixed sequences and simple parts. When unexpected situations appear, such as object movement or interference, the robot often cannot adapt and may just stop or wait.

Looking at production line balancing studies, Kheirabadi et al. [10] pointed out in their CRALBP review that most existing models still do not really consider how robot skills transfer across different stations. This creates a gap between learning a skill in one place and actually improving efficiency at the system level. Because of this, teaching-based motion learning alone is usually not enough. It still needs to be combined with better perception of the environment and more precise force control if it is to be used in real assembly chains.

5.2 Grasping, Assembly, and Safe Trajectory Control

Once tasks are assigned, the robot still has to actually do the work in a shared space with humans. This stage is usually difficult because two things happen at the same time: one is picking up parts that may be stacked, similar in shape, or partially hidden; the other is avoiding collisions while humans are moving around in the same workspace.

For grasping, many methods split the problem into two parts. First, the system detects the object in a 2D RGB image, often using models like an improved YOLOv7. Then, based on the detected region, a 3D point cloud is used to estimate the full 6D pose, often through point-pair feature matching. The general idea is simple: first narrow down where the object is, then refine the pose more accurately in 3D space [21]. This two-stage pipeline is widely used because it is relatively stable in practice.

For safety during motion, one common idea is to directly add constraints into trajectory planning. Some methods use Control Barrier Functions to turn safety distance rules (like those in ISO/TS 15066) into mathematical constraints. On top of that, human motion prediction can be included in reinforcement learning so that the robot learns to avoid risky states while still trying to reach its target [29]. In simulation, these approaches usually behave better than classical planners like Rapidly-exploring Random Tree Star (RRT*) or PSO when humans move in moderate or slightly aggressive ways. But in real environments, things are less stable. When human motion becomes unpredictable, most systems still rely on slowing down or emergency stop as a backup rather than fully reliable prediction.

There is also a line of work using large models for assembly planning. A typical setup is a "cloud large model + edge small model" structure [30]. The cloud side reads instructions and generates rough constraints or step-level guidance, while the edge side handles perception and low-level execution, sometimes with digital twin checking in between. This setup looks flexible in general tasks, but in precise operations like shaft-hole insertion, the success rate of automatically generated key points is still not very reliable. So at this stage, it is better to treat it as an early exploration rather than something ready for real production lines.

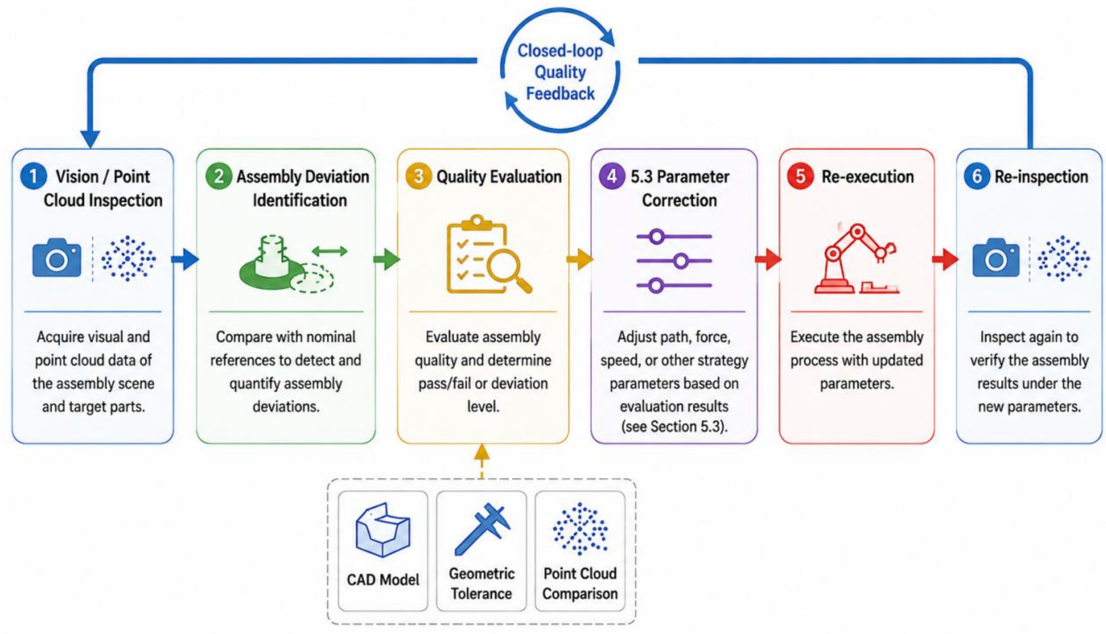
After assembly, quality checking is also usually done immediately. The HRCA framework from Zheng et al. [22] uses a model called DuHa-v2 to segment hand actions in real time. Once a key step is finished, the system uses interaction features from the segmented actions to decide whether the assembly is correct or not. It can detect several common direction errors without adding much computation cost. But at the moment, it mostly focuses on whether the part is oriented correctly, and still does not cover more subtle problems like small gaps, loose fit, or force and torque abnormalities.

5.3 Visual Quality Detection and Closed-Loop Feedback

Putting a part into the correct position does not always mean the assembly is finished. In practice, even after placement, there can still be small deviations caused by pose estimation errors, accumulated tolerances, or simple operational mistakes. If these deviations are not noticed early, they may continue to affect later steps in the assembly process.

In HMLV production, it is usually not realistic to collect enough defect data for every product type to train a supervised model. Because of this, many methods avoid learning "defects" directly and instead compare the actual part state with an ideal reference. A common way is to use the CAD model as the nominal standard, then align the measured point cloud using ICP, and compute translation and rotation differences. Compared with simple pass/fail output, this kind of result is more informative. For instance, a shift in translation may suggest fixture misalignment, while a rotation error can sometimes indicate that the part is slightly tilted due to accumulated tolerances. These signals are then used as hints for later adjustment.

Figure 4: Assembly Quality Closed-Loop Feedback Mechanism Based on Visual Quality Detection



In this sense, quality detection is gradually not only about judging whether something is correct or wrong, but also about providing some direction for correction. Ideally, the output should include not just a label, but also information like deviation size, direction, and possible source. Only when this kind of information is passed back into earlier stages—such as skill adjustment or task planning—can it actually influence the execution process.

In practice, the use of these deviation signals usually happens in two ways. One is gradual correction over batches: the deviation from the current run is fed into iterative learning control, and small adjustments are made in grasping or assembly motions in the next cycle. In HMLV settings, this kind of learning is often applied to general skills like grasping, positioning, or insertion, so that experience can slowly accumulate across different products. The other is more direct correction: when a deviation becomes too large in a key step, the system may trigger replanning, such as redoing a process, reassigning a task, or inserting a calibration step. Together, these two ways form a loop where execution, detection, and correction are continuously connected, although not always in a very stable or fully reliable way.

Overall, closed-loop quality feedback is still more of a research idea than something fully stable in real production lines. There are still many practical issues, such as how well models generalize to different products, how to connect perception results with task-level meaning, and how to balance safety and efficiency when multiple corrections happen at the same time. In many systems, the feedback signal is still not strong enough to fully drive decision-making or skill updates, so even if detection accuracy is high, it does not always translate into real closed-loop behavior.

So rather than saying the system already forms a complete loop, it is probably more accurate to say that current work is still trying to make detection results more usable for later correction. The main direction seems to be moving from simple “detecting errors” toward “using errors to adjust skills,” but this is still not very stable in practice.

6. Challenges and Development Trends

From what has been discussed in the previous sections, most of the existing methods can work reasonably well in controlled or semi-controlled environments. But once the system is moved into real HMLV production lines, a lot of problems start to appear, especially in the interfaces between perception, decision, execution, and feedback. It is less about whether a single module works or not, and more about whether they can still function together without breaking down under changing conditions. So the main difficulty has gradually shifted to this kind of system-level coupling problem.

6.1 Key Challenges

(1) Cross-scenario generalization problems.

Most multimodal models still perform quite well in the same type of environment where they are trained, but when the layout, lighting, or workstation structure changes, performance usually drops a lot [31]. Some methods like few-shot learning or domain adaptation try to fix this, but they often assume the new environment is still “similar enough” to the old one. In real factories, this assumption is often not true, especially when production lines are frequently reconfigured.

(2) Gap between perception and decision-making.

In many cases, visual or skeleton-based signals cannot really tell the difference between actions that look similar but actually mean different things in the process. On the other hand, decision-making models often assume they already have a clean and structured input from perception, which is not always the case. They also usually only look at the current frame or current state, without really using long process history. Because of this, even if perception accuracy goes up, it does not always help decision quality in a direct way.

(3) Safety, human factors, and efficiency are not well balanced.

There are many studies trying to include human factors in task allocation, but in practice this often leads to some loss in efficiency. Safety control is still mostly handled after a risk is detected, usually at the trajectory level, rather than being integrated earlier into planning. At the same time, things like fatigue or psychological state are still very hard to measure in real time, so most models just simplify them or ignore them. Because of this, multi-objective optimization is still not very stable in real use.

(4) Weak closed-loop feedback in execution.

In many systems, quality checking still ends up as a simple pass/fail result. Even when vision-based methods are used, they often do not give detailed information that can directly guide correction. Digital twins are mostly used before execution for checking paths or motions, but not really for understanding the full system state during operation. So when something unexpected happens, the robot usually does not have enough understanding to adjust itself properly [7,32].

These problems are actually not separate from each other. If generalization is weak, then perception itself becomes unstable, and everything built on top of it is affected. If perception and decision are not well connected, then even good recognition results are not very useful. If safety is handled separately from planning, then the system becomes fragmented. And if feedback is weak, then the system cannot really improve over time using real operation data.

So in general, improving just one part of the system does not seem enough anymore. The bigger issue is how to make different parts of the system work together more consistently, even when the environment is changing and not well structured.

6.2 Development Trends

(1) Embodied intelligence and vision-language-action models will probably receive more attention. One reason is that current perception systems are still not very good at connecting what they see with what the task actually means. Vision-language-action models try to put language, images, and robot actions into the same representation, so they may help bridge part of this gap [19]. At the same time, there are still many practical issues. Real-time response, safety checking, unexpected situations, and even responsibility during failures are all difficult problems. Because of this, these models are better viewed as a possible direction than as something that can directly replace existing industrial control systems.

(2) Digital twins may play a bigger role before robots actually start moving. Instead of checking safety only after planning is finished, some recent work uses digital twins to test different task plans in advance. Reachability, collision risk, and other basic constraints can all be checked before execution begins [30]. This makes planning a little safer, at least in theory. But whether it works well still depends on how closely the virtual model matches the real production line. If the model quickly becomes outdated, the benefit is much smaller. Real-time computing is also another practical limitation.

(3) Quality feedback is likely to become more useful if it can really change robot behavior.

At the moment, many systems still stop after saying whether the assembly is qualified or not. A more useful direction may be to output richer information, such as pose errors, force or torque changes, and abnormal process states. If these signals can actually modify robot skills or trigger local replanning, then quality inspection becomes part of the execution process instead of simply checking the final result [23]. But whether these corrections can be reused across different products, or remain stable after many update cycles, is still an open question.

(4) Low-sample adaptation is still important for HMLV production.

Frequent changes in products, workstation layouts, and lighting conditions make it difficult to collect enough new training data every time a production line changes. Methods such as parameter-efficient fine-tuning, retrieval augmentation, knowledge distillation, and domain adaptation all try to reduce this dependence on large datasets. Even so, most of them still assume that the old and new environments are reasonably similar. Once the difference becomes too large, especially for vision or force-controlled assembly, only using a small number of samples may still not be enough.

These directions are closely related, although they are often studied separately. Better vision-language-action models may improve task understanding, digital twins can provide an extra layer of safety checking, quality feedback can gradually improve robot skills, and low-sample learning may make production line changes easier to handle. None of them is likely to solve the whole problem alone. They seem more useful when they are combined with existing systems instead of replacing them.

Overall, future work will probably pay less attention to improving a single model in isolation. A more practical goal may be building systems that can keep working when conditions change, explain their own decisions, and use feedback to improve over time. Exactly how to achieve this is still far from settled, but it is becoming a common direction in recent research.

7. Conclusion

This review discussed recent work on multimodal perception and task execution for intelligent assembly human-robot collaboration. The discussion was organized around the process of perception, decision-making, execution, and feedback. Looking at the existing studies together, it can be found that intelligent assembly is becoming a system problem rather than a problem of one single technology. Perception, task planning, robot execution, and quality inspection are closely connected, and weakness in one part usually affects the others. This is especially true in HMLV production, where products and working conditions often change.

A lot of progress has been made in recent years. Different perception methods now combine vision, skeleton information, point clouds, force signals, and language information instead of relying on only one source. Task allocation also considers more than production efficiency and has gradually included factors such as fatigue, workload, and trust. In execution, teaching-based learning, trajectory planning, and visual inspection have all improved compared with earlier work. However, these improvements do not solve every problem. Models still have difficulty when they are used in different workstations. Safety planning and task allocation are often developed separately. In many cases, quality inspection is still mainly used to tell whether the assembly is correct, instead of helping the robot improve the next operation.

Because of this, future research may need to spend more effort on the links between different modules. Better perception alone does not always make task allocation better. Better planning also does not necessarily lead to more stable execution. The information produced by perception still needs to be translated into task understanding, while quality feedback should become useful information for later decisions instead of simply ending the process after inspection.

Some directions seem worth paying more attention to. One is reducing the amount of data needed when production lines or workstations change. Another is making vision-language-action models more reliable in real assembly environments instead of only showing good performance in demonstrations. Digital twins may also be useful if they can support safety checking before execution and continue monitoring during operation. At the same time, quality feedback could become more valuable if it is connected with robot skill updating or task replanning rather than remaining only as a final inspection result.

There are still some limitations in this review. The discussion mainly comes from representative studies reported in recent years and focuses on the technical relationships between different parts of the assembly process. It is not a systematic bibliometric review, and the proposed framework has not been verified with long-term industrial operation data. More evidence from real production lines and different assembly scenarios will still be needed to further evaluate and improve the framework discussed in this paper.

References

[1] Keshvarparast, A., Battini, D., Battaia, O., & Pirayesh, A. (2024). Collaborative robots in manufacturing and assembly systems: literature review and future research agenda. Journal of Intelligent Manufacturing, 35(5), 2065-2118.

[2] Lu, Y., Zheng, H., Chand, S., Xia, W., Liu, Z., Xu, X., ... & Bao, J. (2022). Outlook on human-centric manufacturing towards Industry 5.0. Journal of manufacturing systems, 62, 612-627.

[3] Zhang, C., Wang, Z., Zhou, G., Chang, F., Ma, D., Jing, Y., ... & Zhao, D. (2023). Towards new-generation human-centric smart manufacturing in Industry 5.0: A systematic review. Advanced Engineering Informatics, 57, 102121.

[4] Vianello, L., Ivaldi, S., Aubry, A., & Peternel, L. (2024). The effects of role transitions and adaptation in human–cobot collaboration. Journal of Intelligent Manufacturing, 35(5), 2005-2019.

[5] Wang, B., Zheng, L., Wang, Y., & Qi, Z. (2025). LLM-based multi-agent task planning for human-robot collaborative assembly balancing operator experience and efficiency. Journal of Manufacturing Systems, 82, 1020-1045.

[6] Suryoputro, M. R., Zhang, T., & Kiridena, S. (2025). Ergonomics and safety in human–collaborative robot interaction: A review of literature for manufacturing industries. International Journal of Industrial Ergonomics, 110, 103837.

[7] Liu, C., Qian, Y., Tang, D., Zhu, H., Pang, J., & Cai, Q. (2026). From insight to action: Embodied multi-agent system integrating vision language model for digital twin-assisted human-robot collaborative assembly. Journal of Manufacturing Systems, 85, 531-556.

[8] Liu, C., Song, J., Tang, D., Wang, L., Zhu, H., & Cai, Q. (2026). From insight to autonomous execution: VLM-enhanced embodied agents towards digital twin-assisted human-robot collaborative assembly. Robotics and Computer-Integrated Manufacturing, 98, 103176.

[9] Petzoldt, C., Harms, M., & Freitag, M. (2023). Review of task allocation for human-robot collaboration in assembly. International Journal of Computer Integrated Manufacturing, 36(11), 1675-1715.

[10] Kheirabadi, M., Keivanpour, S., Chinniah, Y. A., & Frayret, J. M. (2023). Human-robot collaboration in assembly line balancing problems: Review and research gaps. Computers & Industrial Engineering, 186, 109737.

[11] Fathi, M., Sepehri, A., Ghobakhloo, M., Iranmanesh, M., & Tseng, M. L. (2024). Balancing assembly lines with industrial and collaborative robots: Current trends and future research directions. Computers & industrial engineering, 193, 110254.

[12] Angleraud, A., Ekrekli, A., Samarawickrama, K., Sharma, G., & Pieters, R. (2024). Sensor-based human–robot collaboration for industrial tasks. Robotics and computer-integrated manufacturing, 86, 102663.

[13] Zhang, Y., Jiang, X., Qi, X., Cui, E., & Fu, H. (2026). Few-shot assembly action recognition in smart manufacturing: A cross-domain metric framework. Advanced Engineering Informatics, 74, 104610.

[14] Wang, T., Liu, Z., Wang, L., Li, M., & Wang, X. V. (2024). Data-efficient multimodal human action recognition for proactive human–robot collaborative assembly: A cross-domain few-shot learning approach. Robotics and Computer-Integrated Manufacturing, 89, 102785.

[15] Dong, L., Hu, T., Sun, T., Li, J., & Ma, S. (2025). RGB video and inertial sensing fusion method for human action recognition in human-robot collaborative manufacturing. Journal of Manufacturing Systems, 83, 216-234.

[16] Matin, A., Islam, M. R., Wang, X., & Huo, H. (2024). Robust multimodal approach for assembly action recognition. Procedia Computer Science, 246, 4916-4925.

[17] Gao, Q., Liu, Z., Hou, M., Sa, G., & Tan, J. (2026). Multimodal action recognition in human–robot collaborative assembly: A contrastive semantic query approach. Robotics and Computer-Integrated Manufacturing, 98, 103163.

[18] Ding, P., Zhang, J., Zhang, P., & Lyu, Y. (2025). Large Language Model-powered Operator Intention Recognition for Human-Robot Collaboration. IFAC-PapersOnLine, 59(10), 2280-2285.

[19] Fan, J., & Zheng, P. (2024). A vision-language-guided robotic action planning approach for ambiguity mitigation in human–robot collaborative manufacturing. Journal of Manufacturing Systems, 74, 1009-1018.

[20] He, J., Zhang, C., Wang, Y., Zheng, P., & Zhang, J. (2026). Intention recognition and task allocation in human–robot collaborative assembly based on adaptive networks and DT belief space-GA. Journal of Manufacturing Systems, 84, 20-39.

[21] Hui, J., Zhang, Y., Ding, K., Guo, L., Chen, C. H., & Wang, L. (2024). A multi-stage approach for desired part grasping under complex backgrounds in human-robot collaborative assembly. Advanced Engineering Informatics, 62, 102778.

[22] Zheng, H., Xia, W., & Xu, X. (2025). A human-robot collaborative assembly framework with quality checking based on real-time dual-hand action segmentation. Robotics and Computer-Integrated Manufacturing, 94, 102976.

[23] Geiser, A., Fuhrmann, A., Hailu, M., Hashem, S., Benfer, M., & Lanza, G. (2026). Automated Vision-Based Quality Control of Robotic Assembly in HMLV Manufacturing. Procedia CIRP, 139, 198-203.

[24] Alessio, A., Aliev, K., & Antonelli, D. (2024). Multicriteria task classification in human-robot collaborative assembly through fuzzy inference. Journal of Intelligent Manufacturing, 35(5), 1909-1927.

[25] Kiyokawa, T., Shirakura, N., Wang, Z., Yamanobe, N., Ramirez-Alpizar, I. G., Wan, W., & Harada, K. (2023). Difficulty and complexity definitions for assembly task allocation and assignment in human–robot collaborations: A review. Robotics and Computer-Integrated Manufacturing, 84, 102598.

[26] Weyel, J., Kalmbacher, B., Beinke, T., Schultheis, M., Müller, T., Milde, S., ... & Hartmann, F. (2025). Dynamic Human-Cobot Task Allocation: Productivity versus Balance in Manufacturing. IFAC-PapersOnLine, 59(26), 295-300.

[27] Zhu, Y., Chen, S., Zhang, C., Piao, Z., & Yang, G. (2024). Development of adaptive safety constraint by predicting trajectories of closest points between human and co-robot. Journal of Intelligent Manufacturing, 35(3), 1197-1206.

[28] Zhang, S., Huang, H., Huang, D., Yao, L., Wei, J., & Fan, Q. (2022). Subtask-learning based for robot self-assembly in flexible collaborative assembly in manufacturing. The International Journal of Advanced Manufacturing Technology, 120(9), 6807-6819.

[29] Gao, J., Wu, K., Li, W., Glogowski, P., & Kuhlenkötter, B. (2026). Human-centered safe trajectory planning for human–robot collaborative assembly of electric vehicle battery copper busbar. CIRP Journal of Manufacturing Science and Technology, 68, 84-102.

[30] Ma, D., Zhang, C., Xu, Q., & Zhou, G. (2026). Large and small-scale models' fusion-driven proactive robotic manipulation control for human-robot collaborative assembly in industry 5.0. Robotics and Computer-Integrated Manufacturing, 97, 103078.

[31] Ding, K., Mao, Q., Zhang, Y., Zhang, Y., Zheng, P., & Wang, L. (2026). Review and perspectives on multimodal perception, mutual cognition, and embodied execution for human–robot collaboration in Industry 5.0. Robotics and Computer-Integrated Manufacturing, 101, 103280.

[32] Müller, M., Ruppert, T., Jazdi, N., & Weyrich, M. (2024). Self-improving situation awareness for human–robot-collaboration using intelligent Digital Twin. Journal of Intelligent Manufacturing, 35(5), 2045-2063.

Funding

This research received no external funding.

Conflicts of Interest

The authors declare no conflict of interest.

Acknowledgment

This paper is an output of the science project.