Journal of Manufacturing Systems 86 (2026) 247–263

Contents lists available at ScienceDirect

Journal of Manufacturing Systems

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

Technical paper

Industrial-grade trustworthy embodied system for power battery disassembly based on the disassembly NeuroSymbolic World Model ☆

Yanlong Peng [a], Zhigang Wang [b], Yuping Zhang [c,*,1], Pengxu Chang [a], Ziwen He [a], Ming Chen [a,*,1]

[a] Shanghai Jiaotong University, School of Mechanical Engineering, No. 800, Dongchuan Road, Shanghai, 200240, China
[b] Intel Labs China the 8F, Raycom InfoTech Park Tower A, No. 2, Kexueyuan Nanlu, Haidian District, Beijing, 100190, China
[c] Wuhan Power Battery Recycling Technology Co., Ltd No. 258 Xiangfei Road, Intelligent Manufacturing Industrial Park, Changjiang New Area, Wuhan City, Hubei Province, Wuhan, 430070, China

ARTICLE INFO

Keywords:
NeuroSymbolic embodied intelligence
Industrial-grade battery disassembly
Trusted robotic systems

ABSTRACT

The power battery disassembly scenario is characterized by high heterogeneity, strong uncertainty, and a largely unstructured environment, which poses severe bottlenecks for conventional automation and simple intelligent approaches in real industrial processes. To address the challenges of safety, reliability, and generalization in such complex settings, this paper proposes a trustworthy NeuroSymbolic embodied intelligence system for power battery disassembly. The system is built upon a dual-system (System 1 + System 2) hybrid control framework, forming a Disassembly NeuroSymbolic World Model. The Deliberative System leverages large language models (LLMs) to enable long-horizon task reasoning, multi-robot collaboration planning, and knowledge-driven, interpretable decision-making. In contrast, the Reactive System employs a kinematics informed neural network to accelerate motion computation, achieving trustworthy end-to-end perception and fast, reactive control. Meanwhile, the system incorporates both mobile and stationary embodied hardware units, namely the Mobile Autonomous Unit (MAU) and the Stationary Autonomous Unit (SAU), to establish a scalable disassembly execution platform. By deeply integrating symbolic planning with data-driven models, the proposed world model supports multimodal cross-validation and continual learning, thereby realizing an industrial-grade, "usable and trustworthy" closed-loop perception–reasoning–execution framework. Extensive experimental results demonstrate the safety, robustness, and generalization capability of the proposed approach in complex and open disassembly environments, providing a viable technical pathway toward trustworthy and deployable intelligent solutions for industry.

1. Introduction

The rapid global growth of electric vehicles has been accompanied by a continuous increase in the number of waste power batteries reaching end of life [1,2]. The enormous installed base of vehicles, together with the mandatory retirement of power batteries within approximately eight years depending on their state of health (SOH) [3], poses significant challenges to the disassembly and green manufacturing industries. Current power battery disassembly processes remain largely manual, exposing operators directly to severe safety hazards, including fires and explosions triggered by thermal runaway and mechanical abuse (e.g., dropping, crushing, and puncturing) [4]. Moreover, non-standardized operations and disassembly procedures further exacerbate heavy metal contamination and hazardous gas emissions.

Automation technologies and robot preprogramming methods encounter substantial bottlenecks when applied to power battery disassembly [5]. Unlike highly standardized forward manufacturing with stable takt times, battery disassembly is inherently a strongly uncertain scenario. Variations in vehicle models, generations, and suppliers lead to highly heterogeneous layouts of fasteners, sealing structures, and high-voltage wiring [1]. In addition, differences in battery health conditions and accident-induced deformations introduce further unknowns,

☆ This document is the results of the research project "2021 High Quality Development Project (TC210H02C)" funded by the Ministry of Industry and Information Technology of China.

* Corresponding authors.
E-mail addresses: me-pengyanlong@sjtu.edu.cn (Y. Peng), zhi.gang.wang@intel.com (Z. Wang), zhangyuping@gem.com.cn (Y. Zhang), 19119175490@sjtu.edu.cn (P. Chang), heziwen@sjtu.edu.cn (Z. He), mingchen@sjtu.edu.cn (M. Chen).
[1] These authors contributed equally and jointly supervised this work.

https://doi.org/10.1016/j.jmsy.2026.03.008
Received 12 January 2026; Received in revised form 13 March 2026; Accepted 14 March 2026

rendering conventional offline programming and rigid process pipelines ineffective [6]. Although forward-looking approaches such as knowledge graphs and flexible production lines have been explored, they remain some distance away from practical industrial deployment [7].

The recent surge in artificial intelligence research has opened new perspectives for addressing these challenges [6]. Our previous work has focused on power battery disassembly processes and the concept of flexible mixed-model production lines [8], as well as explorations and validations at the level of embodied intelligence technologies [9], laying a solid foundation for the present study. However, systems that rely solely on end-to-end control are prone to critical failures caused by large-model "hallucinations", which can lead to severe consequences in industrial settings. We therefore argue that, in response to the current challenges faced by the disassembly industry, there is an urgent need for usable and trustworthy embodied disassembly agents [10]. Such agents must operate in open and unstructured workstations while providing verifiable safety boundaries, real-time closed-loop perception–reasoning–execution, and support for multi-robot collaboration.

To this end, we develop a trustworthy[2] embodied intelligence system equipped with a Disassembly NeuroSymbolic World Model.[3] NeuroSymbolic AI [11,12] effectively combine the complementary strengths of knowledge-driven and data-driven paradigms [11,13], enabling a dual-system (System 1 + System 2) hybrid control architecture [14]. We design a complete embodied robotic hardware system to support this framework. The Deliberative System employs large language models to perform autonomous reasoning and planning for long-horizon tasks, while enabling multi-robot collaboration. The Reactive System accelerates motion planning through an end-to-end kinematics informed neural network. By jointly leveraging knowledge–data co-driving mechanisms and symbolic planning, the proposed embodied system achieves industrial-grade trustworthiness while exhibiting strong generalization capability.

The remainder of this paper is organized as follows. Section 2 reviews related work on disassembly and state-of-the-art embodied intelligence. Section 3 introduces the architecture of the trustworthy embodied intelligence system based on the Disassembly NeuroSymbolic World Model. Technical details of the dual-system (brain–cerebellum) hybrid control paradigm are presented in Section 4. Section 5 reports a series of experiments conducted to validate the proposed system. Finally, Section 6 concludes the paper and outlines directions for future work.

2. Literature review

This section reviews the relevant literature, organized into five key research areas closely related to this study. We first survey the current state of the art in power battery disassembly technologies. The second and third subsections discuss the implementation of embodied intelligence robot control architectures. The fourth subsection introduces task and motion planning as well as multi-robot collaboration. Finally, the last subsection reviews physics-informed neural networks and their applications in embodied intelligence.

2.1. State of the art in power battery disassembly

For large-scale disassembly of electric vehicle battery packs, extensive research has focused on replacing manual labor with automated equipment or offline-programmed robots [5,15]. Automated disassembly platforms based on industrial robotic arms [16] have demonstrated empirical progress in operations such as screw removal and cover detachment. Semi-automated collaborative platforms or predefined cognitive robotic systems can further ensure relatively high disassembly efficiency [17,18]. In addition, some studies have attempted to adopt standardized production-line methodologies, such as template-matching-based tool recommendation [19] and reusable unscrewing systems [16]. However, the generalization capability and robustness of such approaches remain limited, and the diversity of disassembly targets poses significant challenges for real-world deployment.

In recent years, an increasing number of studies have incorporated industrial vision into robotic disassembly processes for task planning [20] or visual servoing [21], thereby enabling a certain degree of flexible disassembly. Learning-based disassembly systems have further improved overall efficiency [5]. For example, robot end-effector control with online localization and dynamic correction [22], as well as compliant control based on visual–force fusion [23], have been shown to enhance robots' ability to cope with deformable workpieces under uncertainty. Comprehensive surveys have illustrated how the entire workflow of battery disassembly can be integrated with advanced methods such as machine learning [6]. Nevertheless, current end-to-end approaches still suffer from hallucination issues [10], which hinder their practical adoption on industrial production lines. As a result, there remains a lack of truly trustworthy embodied intelligence systems capable of autonomously disassembling waste power batteries.

Beyond system-level approaches, substantial research efforts have also focused on disassembly actuator design [8,24], human–robot collaborative disassembly [25,26], and battery design for disassembly [27], collectively driving the maturation of power battery disassembly technologies.

2.2. Explicit and implicit robot control technologies in industrial applications

Robotic systems that are widely deployed and relatively mature in industrial settings predominantly adopt explicit model-based control strategies [28,29]. Built upon accurate mathematical models of robot kinematics and dynamics [30], such approaches-including impedance control [31]-have demonstrated outstanding performance in highly structured and repetitive industrial environments. However, these methods rely on the assumption that both the system model and the environment are precisely known, and they are inherently sensitive to modeling errors. When strong nonlinear variations arise in the system or the environment, the models must be continuously updated, which significantly limits adaptability [29,32].

In contrast, implicit control systems are driven by perception based feedback and generate control policies through neural networks, imitation learning, reinforcement learning, or fuzzy logic [23,33]. These approaches are capable of handling complex nonlinear scenarios and often exhibit greater flexibility and resilience for certain tasks [33,34]. In recent years, the vision–language–action(VLA) paradigm has emerged as a popular end-to-end framework [35–37], offering strong robustness against disturbances. Nevertheless, end-to-end methods that directly output low-level control commands remain fundamentally "black-box" in nature. Their stability typically requires massive amounts of training data, and the hallucination problem remains difficult to eliminate [10, 37,38].

[2] Here, trustworthiness is a primary design objective: beyond reliability, which measures success frequency, trustworthiness reflects whether an embodied intelligence system can operate safely, transparently, and be responsibly integrated into industrial environments despite non-zero failure rates.

[3] In this work, we define a world model as a structured internal representation that encodes task-level states, symbolic relations, and their transitions, enabling long-horizon reasoning and decision-making for embodied robotic disassembly. Unlike generative world models that focus on continuous dynamics prediction.

2.3. Hybrid control for embodied intelligence

To address these limitations, recent advances in embodied intelligence have increasingly shifted from purely end-to-end learning toward two-layer robot control architectures, a conceptual framework inspired by Daniel's dual-system theory [14,38]. In this paradigm, the Deliberative System (slow thinking) is responsible for perception, understanding, and task-level decision making at a high semantic level, while the Reactive System (fast thinking) handles real-time motion control at the level of low-level trajectories or motor commands. A key aspect of this approach is the use of intermediate output tokens as an interface bridging the high-level and low-level layers. These tokens represent action primitives that induce changes in the robot itself or in the environment. Prominent humanoid robotics systems, such as Helix [39], π [40], and GR00T [41], adopt this architectural paradigm. However, training intermediate-layer tokens that are well aligned in both temporal scale and semantic meaning requires enormous amounts of data, and the resulting lack of interpretability further hampers industrial adoption.

Another mainstream direction is the construction of NeuroSymbolic frameworks [11,12,42]. In this paradigm, end-to-end policies are assigned to System-1 to enable rapid action mapping, while high-level planning, composition, and reasoning based on symbolic knowledge, logic, or large language models are handled by System-2 [43–45]. This division allows the learning process to be constrained by a small set of domain-specific symbolic rules, thereby significantly enhancing trustworthiness and interpretability—key requirements for industrial-grade embodied intelligence [9,13,42]. Recent studies in robotics have demonstrated that the integration of symbolic models and neural networks provides both interpretability and scalability [46]. A principled combination of LLMs with PDDL-compatible symbolic reasoning has been shown to overcome limitations of traditional symbolic planners and improve planning capability [47]. Beyond planning, [48] integrates symbolic goal representations with learning in open-world environments, enabling continual adaptation to novel situations. Furthermore, the survey on NeuroSymbolic reinforcement learning and planning [49] emphasizes that combining neural and symbolic components enhances interpretability, transparency, and the ability to manage system complexity.

Defining action primitives through symbolic logic rather than intermediate tokens endows them with explicit semantic meaning. As a result, perception outcomes, planning logic, and execution strategies can be directly aligned within a unified, interpretable semantic space, enabling real-time verification and continual learning. In this study, we adopt this framework to construct a Disassembly NeuroSymbolic World Model for power battery disassembly.

2.4. TAMP and multi-robot collaboration

One of the primary ways embodied agents address complex real-world long-horizon tasks is through task and motion planning (TAMP) [50–52]. Model-based or prior-policy-driven TAMP approaches have seen extensive and mature adoption in industrial applications [53,54]. Beyond single-robot settings, Multi-robot Collaboration has attracted increasing attention [55,56], encompassing research on global optimality under limited resources [57], multi-robot collision avoidance, and flow-aware scheduling [55,58]. With the introduction of learning-based methods, notable progress has been achieved in multi-arm collaborative planning using graph neural networks and reinforcement learning [51,59], as well as in TAMP frameworks powered by LLMs and multimodal models [60,61]. The intrinsic complexity of long-horizon tasks and multi-robot collaboration places stringent requirements on the trustworthiness and interpretability of control systems, for which hybrid embodied control architectures offer a promising solution.

2.5. Physics-informed neural networks (PINNs)

Physics-informed neural networks(PINNs) have emerged as a prominent direction in recent machine learning research and have also provided valuable insights for embodied intelligence [62,63]. Within a NeuroSymbolic framework, action primitives are defined through clearly specified semantics and serve as explicit interfaces. Consequently, the underlying end-to-end control logic should likewise incorporate knowledge guidance to improve data efficiency. In this context, adopting physics-informed neural network principles for training embodied action primitives becomes essential. This approach enables end-to-end neural networks to move beyond pure data-driven fitting by embedding prior physical knowledge, thereby substantially improving both data efficiency and prediction accuracy [64,65]. For embodied agents, challenges such as limited data availability and long-tail sparsity [66,67] often lead to poor convergence or weak generalization in robot action policies learned via deep learning or reinforcement learning [68]. To mitigate these issues, prior studies have integrated physical priors with robot embodiment, applying physics-informed concepts to dynamics modeling [69], robotic manipulation [70], and kinematic configuration sampling [71]. In this work, we similarly train kinematics informed neural networks(KINNs) within the Reactive System to address the sampling of constrained solution spaces for robotic motion [72,73]. Compared with PINNs, which primarily focus on partial differential equations, KINNs shift the emphasis toward the intrinsic properties of robotic systems, such as complex analytical kinematic solutions and Jacobian structures. This is achieved through differentiable programming techniques [74,75], which significantly enhance both the inference speed and the trustworthiness of robot action policy networks.

In summary, the above review outlines the key technologies required to construct a trustworthy embodied intelligence system for power battery disassembly. The main contributions of this paper are as follows:

1. To address the challenges faced in power battery disassembly, we develop a trustworthy embodied system equipped with a Disassembly NeuroSymbolic World Model, following a dual-system hybrid control paradigm based on NeuroSymbolic artificial intelligence. In addition, embodied hardware units, including a Mobile Autonomous Unit (MAU) and a Stationary Autonomous Unit (SAU), are specifically designed.
2. For the Deliberative System, trustworthy long-horizon multi-robot collaboration is enabled by a multi-agent planning architecture, consisting of heuristic symbolic search, white-box symbolic checking, and rule-constrained task assignment carried out by dedicated agents.
3. For the Reactive System, kinematics informed neural networks(KINNs) are designed to embed kinematic constraints into the manipulation process, supporting fast reactive control. Furthermore, the dual-system architecture incorporates multimodal cross-validation and continual learning mechanisms, which further enhance the trustworthiness of the embodied system.
4. Extensive experimental evaluations demonstrate that the proposed embodied system achieves industrial-grade trustworthiness through the synergy of knowledge driven and data driven approaches. Moreover, the Disassembly NeuroSymbolic World Model exhibits strong generalization capability, as it fundamentally focuses on understanding and learning disassembly skills.

3. Overview of the disassembly NeuroSymbolic world model framework and embodied system design

This section introduces the overall framework of the industrial embodied system for power battery disassembly(Fig. 1), covering both

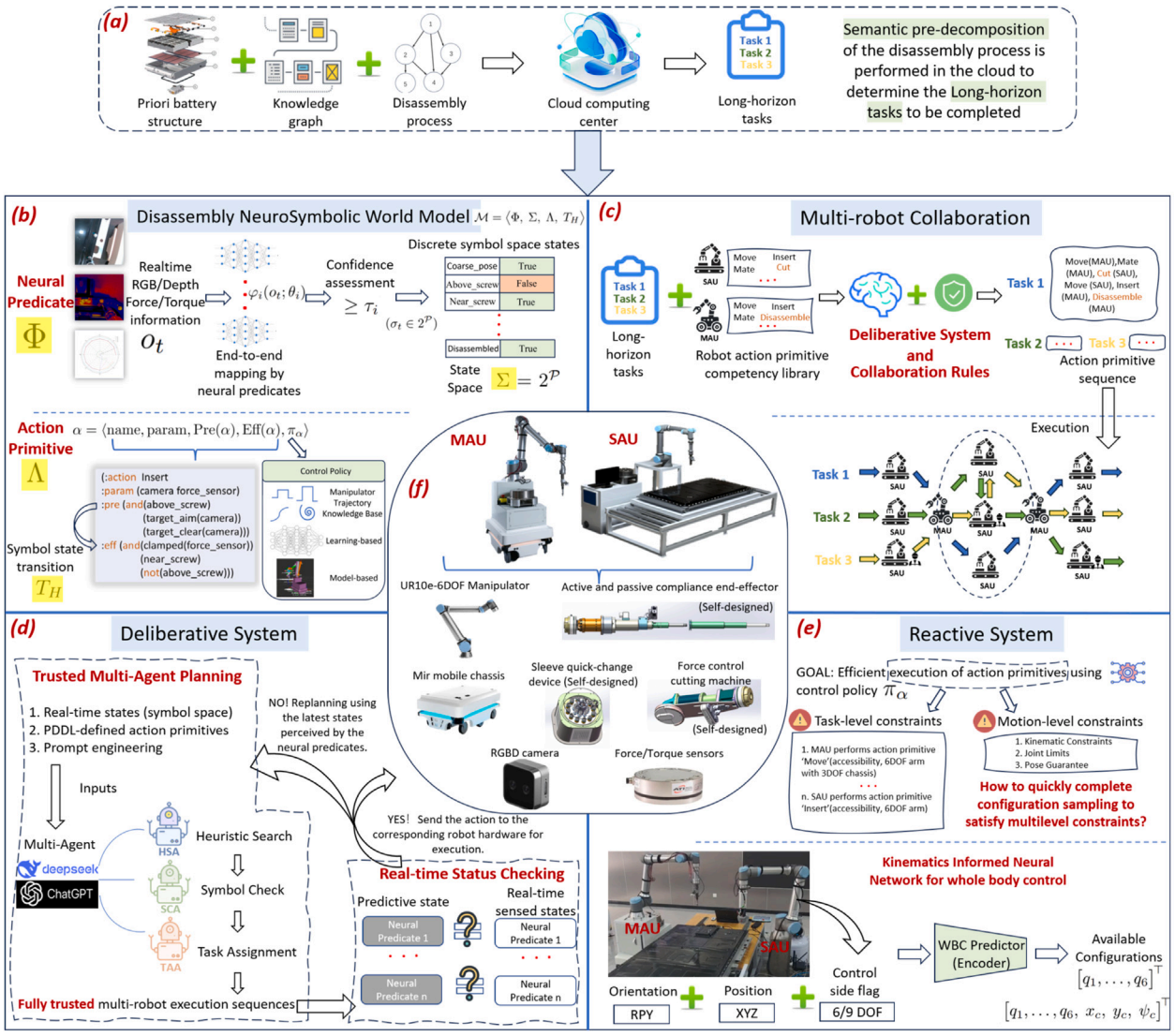

Fig. 1. An overview of the proposed embodied system framework is illustrated from (b) to (f). (a) The cloud layer performs semantic-level task decomposition based on prior knowledge, producing long-horizon tasks as inputs to the embodied system. (b) The Disassembly NeuroSymbolic World Model, defines the neural predicates, state space, robot action primitives with policies, and state transitions. (c) Multi-robot collaboration is employed to accomplish the planning logic for long-horizon tasks. (d) The Deliberative System, where multi-agent heuristic search is performed while guaranteeing full trustworthiness. (e) The Reactive System, which focuses on execution policies for action primitives. (f) The embodied hardware design, consisting of mobile workstations and stationary workstations.

the embodied algorithmic layer and the embodied hardware layer. In addition, rigorous mathematical and symbolic definitions are provided for key concepts such as the world model and the dual-system architecture.

3.1. Disassembly NeuroSymbolic world model

The disassembly processes of power batteries are relatively mature [76,77] and can be broadly divided into preprocessing (information registration, battery discharge, and removal of thermal conductive media), voltage reduction disassembly (disconnecting high-voltage harnesses and busbars), and core component disassembly (signal cables, fasteners, and modules) [8]. Designing dedicated robot end-effectors for each specific disassembly process would inevitably increase system complexity and lead to a reliance on preprogrammed solutions. Therefore, to enhance the generalization capability of the disassembly system and expand its applicability across scenarios, the design of both the Disassembly NeuroSymbolic World Model and the embodied hardware follows the principles below [78]:

Principle 1: From the fundamental principles of object connectivity, disassembly targets can be categorized into detachable connections and non-detachable connections according to the disassembly process.

Principle 2: The Disassembly NeuroSymbolic World Model and the corresponding end-effector designs should be defined based on connection properties, rather than on differences in battery pack models or types.

Accordingly, for power batteries, detachable connections include fasteners, wire harnesses, and modules, whereas non-detachable components include severely corroded or deformed fasteners and welded joints. Since screws are a type of fastener that are widely present in locations such as the top cover and internal modules, accounting for more than 40% of the total disassembly workload [77,79], this study uses rusted screws that cannot be removed through conventional means as an illustrative long-horizon task to explain the working principles of the entire system.

Fig. 1(b) presents the modeling of the Disassembly NeuroSymbolic World Model. For embodied robotic systems, real-time perception of the environment is a fundamental prerequisite for autonomous planning. NeuroSymbolic techniques employ neural predicates to map environmental states into a symbolic space. Each neural predicate φ_i is

an independent end-to-end network that maps sensor observations o_t (e.g., vision and force sensing) to a confidence value:

$$\varphi_i(o_t;\theta_i) \in [0,1] \tag{1}$$

By applying a confidence threshold τ_i, each neural predicate is discretized into a symbolic state σ_t.

$$\sigma_t = NP(o_t) = \{p_i \mid \varphi_i(o_t) \geq \tau_i\}, \sigma_t \in 2^{\mathcal{P}} \tag{2}$$

For example, $Near\_screw$ denotes the relative state between a visually detected target screw and the end-effector, while $Disassembled$ represents the disassembly status of a specific screw. Multiple neural predicates jointly define discrete symbolic states, which together constitute the real-time perceived state space Σ of the embodied agent.

Different robots are equipped with different end-effectors and therefore possess different executable skills. To account for this heterogeneity, we design action primitives as Planning Domain Definition Language(PDDL)-defined options, abstracted from the sub-actions commonly used by human engineers during disassembly. Each primitive α in the global action primitive set Λ is defined as:

$$\alpha = \langle \text{name}, \text{param}, \text{Pre}(\alpha), \text{Eff}(\alpha), \pi_\alpha \rangle \tag{3}$$

Each action primitive is characterized by a name, a set of parameters, execution preconditions, expected effects, and an associated control policy $\pi_\alpha(u|w)$. The policy generates joint-level control commands u, driving the world state Σ to satisfy $\text{Eff}(\alpha)$ within the minimum expected time. Consequently, each action primitive is paired with a dedicated control policy, which can be learned via reinforcement learning, VLA models (Section 2.3), or PINN techniques (Section 2.5). This modular design greatly facilitates the extensibility of the overall world model and constitutes the essence of the Reactive System in the dual-system architecture.

Accordingly, the Disassembly NeuroSymbolic World Model can be formally represented as:

$$\mathcal{M} = \langle \Phi,\ \Sigma,\ \Lambda,\ T_H \rangle \tag{4}$$

Where $\Phi = \{\varphi_1, \ldots, \varphi_i\}$ denotes the set of neural predicates, $\Sigma = 2^{\mathcal{P}}$ denotes the symbolic state space, Λ denotes the set of action primitives, and $T_H : \Sigma \times \Lambda \to \Sigma'$ denotes the symbolic-level state transition function. The Disassembly NeuroSymbolic World Model integrates knowledge-driven symbolic representations of human disassembly expertise with data-driven end-to-end perception networks and control policies. It therefore provides the foundational framework for perception, planning, execution, verification, and learning within the embodied system.

3.2. Dual-system framework

This work focuses on constructing a truly trustworthy, industrial-grade embodied intelligence system capable of completing long-horizon tasks during the disassembly process. Accordingly, semantic-level coarse task decomposition for disassembly (Fig. 1(a)) is treated as a prerequisite. The cloud computing center performs this decomposition based on a knowledge graph and disassembly process knowledge, and dispatches the resulting tasks to the embodied system as inputs.

Deliberative System (Fig. 1(d))

Each long-horizon task is encapsulated within the Disassembly NeuroSymbolic World Model and is associated with a symbolic-space goal state g. Planning in the Deliberative System aims to search for a valid sequence of action primitives such that the resulting symbolic state satisfies the target predicate set g. This planning process can be formulated as finding:

$$\text{Plan} = \langle \alpha_1, \ldots, \alpha_t \rangle,$$
$$where \begin{cases} \text{Pre}(\alpha_t) \subseteq \sigma_{t-1} \\ \sigma_t = T_H(\sigma_{t-1}, \alpha_t) \end{cases} \wedge g \subseteq \sigma_t \tag{5}$$

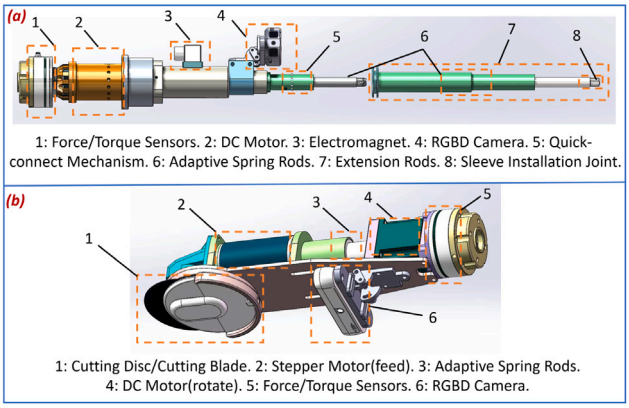

Fig. 2. Self-designed end effector. (a) Active–passive compliant screw disassembly end-effector. (b) Active–passive compliant cutting tool.

Prior to executing each action, the current state σ_{t-1} must satisfy the corresponding preconditions. After execution, the symbolic state is updated according to the state transition function T_H to reflect the expected effects. We adopt a Trusted Multi-Agent Planning framework as the planner, which is described in detail in Section 4.1.1. Planning in the Deliberative System is performed online: before each planned action primitive is dispatched for execution, the planner verifies consistency between the predicted state and the real-time sensor observations, and re-plans optimally whenever discrepancies are detected.

Multi-robot Collaboration (Fig. 1(c))

Once the optimal sequence of action primitives is obtained through task planning, coordination is performed via Collaboration Rules before dispatching the primitives. The decision to execute actions using a single robot or multiple robots is made autonomously by the system and depends on the action primitive libraries available to different robots. The detailed implementation of this process is presented in Section 4.1.2.

Reactive System (Fig. 1(e)) After each action primitive is assigned to the appropriate robot, the Reactive System executes it at the primitive level. Specifically, for each action primitive α, the Reactive System optimizes its control policy $\pi_\alpha(u|w)$ as defined in Eq. (3). This optimization can be formulated as the search for an optimal execution policy:

$$\pi_\alpha^\star = \arg\min\ \mathbb{E}\left[\sum c(\pi_\alpha)\right] \tag{6}$$

The cost function c incorporates factors such as execution time, energy consumption, and collision penalties. Constraints in the Reactive System ensure that symbolic-level effects are achieved while maintaining fast and stable execution. The more efficient the policy π_α, the smoother the operation of the Reactive System, thereby reducing the need for frequent re-planning. For instance, when multiple task-level and motion-level constraints are simultaneously imposed on a motion planner, configuration sampling in the robot's solution space is typically required to satisfy kinematic constraints while avoiding collisions, singularities, and unreachable configurations. Compared with sample-based solvers, control policies guided by prior physical knowledge can significantly improve both efficiency and accuracy (Section 4.2.2).

In summary, the Dual-System Framework relies on action primitives as the standardized interface between the Deliberative and Reactive Systems. After each primitive is executed by KINNs-based low-level control, the symbolic state in the World Model is updated according to PDDL rules and compared with neural predicate-based perception. This state-level verification forms a closed-loop checking mechanism, enabling real-time replanning when inconsistencies are detected.

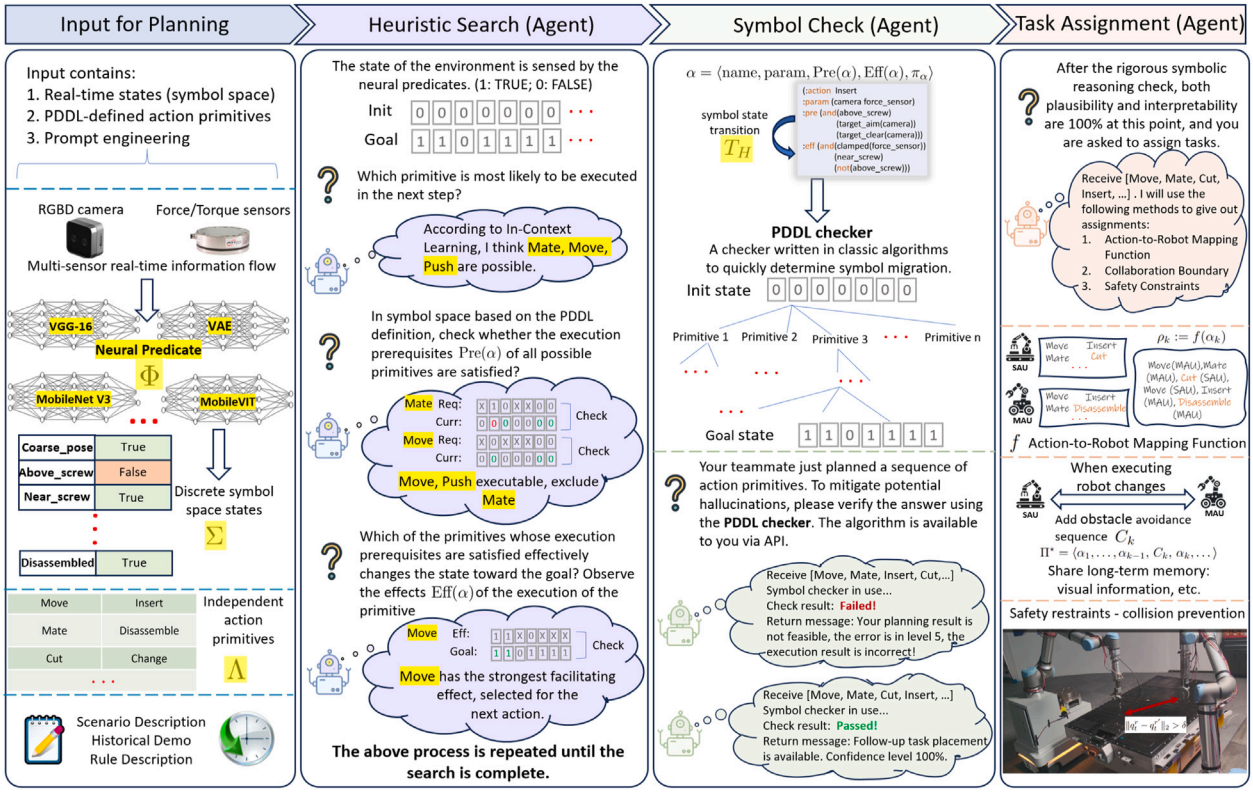

Fig. 3. The Deliberative System adopts a trustworthy multi-agent planning framework, in which the entire planning process is accomplished by three agents. The Heuristic Search Agent performs efficient search in the symbolic space to avoid exhaustive enumeration. The Symbol Checking Agent serves as a critical guarantee of trustworthiness by performing rapid validation using a PDDL checker. The Task Assignment Agent is responsible for coordinating complex scenarios involving multi-robot collaboration.

3.3. Embodied hardware design

In accordance with the aforementioned connection types and the requirements of embodied agents, we design two types of workstations (Fig. 1(f)). The MAU consists of an autonomous mobile manipulator. It is introduced to provide flexible scheduling within the multi-robot collaborative disassembly system, thereby accelerating the overall efficiency of the disassembly process. The SAU, in contrast, is a workstation composed of fixed robotic manipulators.

Following the same design principles, we develop dedicated end-effectors for both the MAU and SAU. As shown in Fig. 2(a), a hybrid active–passive compliant screw disassembly end-effector is designed for detachable connections, specifically threaded joints. A quick-change sleeve interface is integrated, supporting up to 20 different sleeve types for rapid replacement. Fig. 2(b) illustrates a hybrid active–passive compliant cutting tool, designed for non-detachable connections such as severely corroded screws. Both end-effectors are mounted on 6-DOF robotic manipulators and integrate vision and force sensing, enabling active compliance according to the action primitive policies. Passive safety compliance is achieved through a mechanically integrated axial telescopic flexible spring structure.

4. Technical details

This section delves into the technical details and implementation principles of the proposed Deliberative System with trustworthy multi-agent planning, the multi-robot collaboration planning framework for long-horizon tasks, and the Reactive System control strategies based on kinematics informed neural networks.

4.1. Deliberative system: Multi-robot collaboration for long-horizon tasks

4.1.1. Trustworthy multi-agent planning

The proposed trustworthy multi-agent planning framework implements heuristic search driven by multiple LLM-based agents and offers the following advantages:

1. The entire planning process selects an optimal sequence of action primitives in the symbolic space, constituting white-box planning and effectively avoiding hallucination issues commonly associated with conventional LLM-based approaches.
2. Heuristic search is employed to circumvent the combinatorial explosion of the search space caused by symbolic enumeration in optimal planning.
3. A strict symbolic checker ensures industrial-grade trustworthiness.

Fig. 3 illustrates the overall architecture of the Deliberative System. The planner takes as inputs the Disassembly NeuroSymbolic World Model (Section 3.1), real-time sensor signals, and prompt engineering. Symbolic representations of the current state and the goal state are provided to the Heuristic Search Agent. Leveraging in-context learning and structured prompts, the heuristic search proceeds in three stages: (i) predicting the next candidate action primitive, (ii) evaluating whether the preconditions of the action primitive are satisfied, and (iii) assessing whether the expected effects drive the symbolic state toward the goal. This process iterates rapidly until a potentially optimal action sequence is identified. Additionally, the three-step search methodology enhances the interpretability of the LLM, rendering the entire search process human-readable and enabling traceability of the query.

The heuristic search exploits the prior knowledge encoded in the LLM while adhering to symbolic criteria to produce an initial plan. The

	Action primitive	Preconditions and Expected Effects	Description
Both	Move	**:pre** have_coarse_pose ∧ ¬ near_screw **:eff** near_screw ∧ above_screw	Approach or move away from the target bolt.
MAU	Aim_V	**:pre** ¬ pattern ∧ near_screw ∧ ¬ target_aim **:eff** pattern ∧ above_screw ∧ target_aim	Visually estimate the precise pose of the target and adjust the end-effector's orientation to align it coaxially with the bolt.
	Aim_F	**:pre** pattern ∧ near_screw ∧ ¬above_screw ∧ ¬ target_aim **:eff** ¬ pattern ∧ above_screw ∧ target_aim	Estimate the precise pose of the target through a force network, and adjust the end-effector's orientation to align it coaxially with the bolt.
	Recognize	**:pre** above_screw ∧ target_aim ∧ target_clear **:eff** target_match ∧ exist_sleeve ∧ ¬ target_rust ∧ complete_recognize	Identify the target screw type, size, and degree of corrosion (whether cutting is required).
	Insert	**:pre** target_aim ∧ above_screw ∧ complete_recognize ∧ target_match ∧ ¬ target_rust **:eff** clamped ∧ ¬ above_screw	Drive the robotic arm towards the screw to complete the fitting of the sleeve onto the screw.
	Quick_dis	**:pre** familiar_scenes ∧ clamped **:eff** disassembled	Quick disassembly in familiar settings.
	Clean_obs	**:pre** above_screw ∧ ¬ target_clear **:eff** target_clear	Drive the robotic arm to clear obstacles around the screw.
	Change	**:pre** complete_recognize ∧ exist_sleeve ∧ ¬ target_match ∧ ¬ target_rust **:eff** target_match	The robotic arm will replace the end effector with a socket that matches the type, size of the current screw.
	Disassemble	**:pre** clamped ∧ ¬ disassembled **:eff** disassembled	Drive the end-effector motor to rotate, thereby removing the current screw.
SAU	Aim	Do not change the symbol space	synchronise the MAU cross-validation results.
	Insert	**:pre** target_aim ∧ above_screw ∧ complete_recognize ∧ target_match ∧ target_rust **:eff** clamped ∧ ¬ above_screw	Drive the robotic arm to approach the bolt to be cut using the cutting feed strategy.
	Cut	**:pre** target_rust ∧ clamped ∧ ¬ above_screw **:eff** above_screw ∧ complete_recognize ∧ target_match ∧ ¬ target_rust	Drive stepper motors and DC motors to perform slotting and cutting operations on rusted bolts.

Fig. 4. Definition and interpretation of action primitives. According to the specific end-effector capabilities, MAU and SAU maintain their own Robot Action Primitives Competency Library.

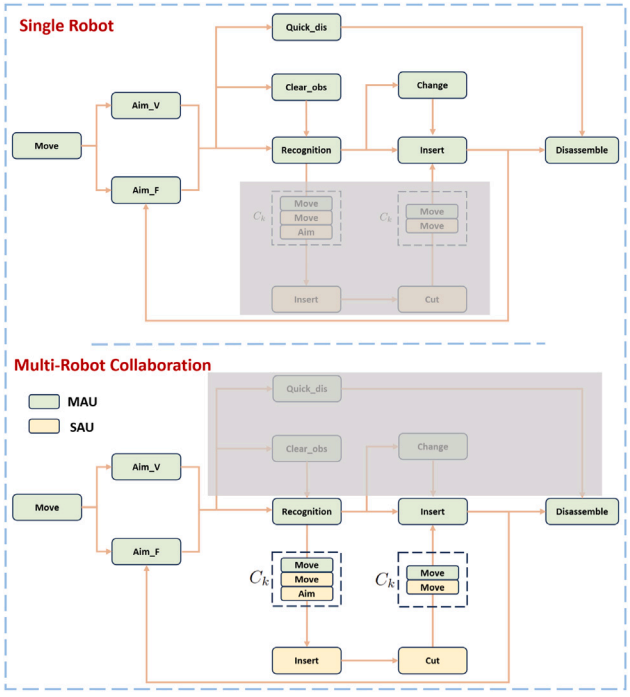

Fig. 5. The action flow for a screw disassembly task. Based on the planned sequence of action primitives, the Deliberative System autonomously determines whether the task should be completed by a single robot or through multi-robot collaboration.

essence of this framework lies in leveraging foundational models and high-quality prompts encompassing successful cases to stimulate the model's in-context learning capabilities. This approach delivers heuristic, rapid search results when the search space becomes prohibitively large. The search agent's architecture remains fixed; adaptation across multiple scenarios or tasks requires no modification to the model itself, instead relying on symbolic-level alterations to the world model. This design further enhances the system's scalability and transferability.

A key guarantee of trustworthiness is provided in the symbolic space by the Symbol Checking Agent. Based on the symbolic-level transition function $T_H : \Sigma \times \Lambda \to \Sigma'$ defined in the Disassembly NeuroSymbolic World Model, we implement a PDDL checker with an API interface accessible to the Symbol Checking Agent. If validation fails, the Heuristic Search Agent is re-invoked to continue searching and to return the corresponding failure reasons. Upon successful validation, a fully trustworthy sequence of action primitives is obtained and passed to the Task Assignment Agent, which determines whether the task should be executed by a single robot or through multi-robot collaboration. This requires the agents to dispatch actions according to rules such as the Action-to-Robot Mapping Function, which will be detailed in the following subsection.

4.1.2. Multi-robot task planning

For the Deliberative System, our objective is to enable autonomous real-time planning for a single manipulator while allowing the resulting plans to be directly applicable to multi-robot tasks. This requires a clear decoupling between NeuroSymbolic planning and task dispatching, with multi-robot collaboration handled independently by the Task Assignment Agent. Accordingly, we establish a multi-robot collaboration system and its associated coordination logic (Fig. 3, Task Assignment Agent). The core procedure is outlined as follows.

Robot Action Primitives Competency Library

$$R = \{r_1, \ldots, r_m\}, \quad \Lambda_r \subseteq \Lambda \ (r \in R), \quad f : \Lambda \to R \tag{7}$$

We first construct a competency library of action primitives based on the available end-effectors. Λ denote the global set of action primitives, and $\Lambda_r \subseteq \Lambda$ represent the subset that robot r can actually execute. The function f is a lookup-based Action-to-Robot Mapping Function that assigns each action primitive to a specific robot according to the dynamically maintained competency libraries of the end-effectors. For example, the MAU is equipped with the end-effector shown in Fig. 2(a) and therefore supports the action primitive *Disassemble*. The SAU,

equipped with the end-effector shown in Fig. 2(b), supports the action primitive *Cut*.

Planning Stage

$$\Pi = \langle \alpha_1, \ldots, \alpha_K \rangle = \text{Planner}(\sigma_0, g, \Lambda) \tag{8}$$

During planning, the Heuristic Search Agent does not distinguish between robots. Given the initial symbolic state σ_0 and the goal g, the Deliberative planner outputs an action sequence Π without specifying which robot will execute each action. The executor for each action primitive α_k is determined at execution time via the mapping function f.

Action-to-Robot Mapping Function

$$\rho_k := f(\alpha_k), \quad \vec{\rho} = \langle \rho_1, \ldots, \rho_K \rangle \tag{9}$$

Here, ρ_k denotes "which robot executes the kth action". Multi-robot task dispatching is therefore determined during this stage, based on the real-time updates of the robot action primitives competency libraries.

Collaboration Boundary

Based on the above formulation, we can explicitly define the Collaboration Boundary among robots:

$$B = \left\{ k \mid \rho_k \neq \rho_{k-1} \right\} \quad (2 \leq k \leq K) \tag{10}$$

Let B denote the set of indices at which robot switching occurs. For each $k \in B$, a coordination segment C_k (e.g., pose sharing or collision-avoidance motion) is inserted:

$$\Pi^\star = \langle \alpha_1, \ldots, \alpha_{k-1}, C_k, \alpha_k, \ldots \rangle \tag{11}$$

This mechanism ensures that whenever the "baton" is handed over to another robot, a short coordination sequence is automatically inserted. This allows the previous robot to communicate its pose and target information to the next robot (enabling local viewpoint sharing) or to move aside to avoid collisions.

Execution-Monitoring-Replanning

$$\text{Done}_k(t) = \left[\texttt{Eff}(\alpha_k) \subseteq \sigma_t \right] \tag{12}$$

During execution, information collected from multiple sensors is continuously mapped into symbolic states via neural predicates, as defined in Eq. (2). The robot ρ_k executing the current action α_k operates according to the corresponding action primitive (either a predefined policy or a real-time adaptive policy) while continuously verifying consistency between expected and observed outcomes. If the preconditions of the next action are no longer satisfied, the remaining plan is immediately discarded and the planner is re-invoked. After replanning, the action-to-robot mapping and collaboration boundary planning are recomputed accordingly:

$$\begin{aligned} &\neg \texttt{Pre}(\alpha_{k+1}) \subseteq \sigma_t \\ \Longrightarrow\ &\Pi = \texttt{Planner}(\sigma_t, g, \Lambda) \end{aligned} \tag{13}$$

Safety restraints To prevent inter-robot collisions, let q_t^r denote the end-effector pose of robot r, and let δ be a predefined safety distance. The following Safety Constraints are imposed:

$$\forall r \neq r', \ \| q_t^r - q_t^{r'} \|_2 > \delta \tag{14}$$

When inserting a coordination segment C_k, the above constraint is enforced as a hard constraint within the local trajectory optimization.

Task Assignment Implementation Task allocation is performed at the action primitive level. Each primitive is assigned at runtime through the Action-to-Robot Mapping Function based on the predefined capability library. The mapping process is centralized, while execution is fully decentralized: each robot executes its assigned primitive locally using its own motion controller and policy.

The system adopts an event-driven coordination scheme over a local network. Only primitive-level completion signals and symbolic state updates are exchanged between robots and the Task Assignment Agent. Since motion control is executed locally and primitives are coarse-grained, communication latency has negligible impact on execution stability. Synchronization is enforced by verifying primitive completion before releasing dependent actions, forming a deterministic coordination cycle.

In summary, the executor of each action is specified by the Action-to-Robot Mapping Function (Eq. (7)), the timing of information sharing and collision avoidance is defined by the collaboration boundary (Eq. (10)), and replanning is triggered by the symbolic monitoring condition (Eq. (13)).

4.1.3. PDDL action definitions for multi-robot long-horizon tasks

This subsection continues to use the long-horizon task of rusted screw disassembly as an illustrative example to further detail the PDDL action definitions. Based on differences in end-effector capabilities, we design distinct robot action primitives competency libraries (Fig. 4). This process incorporates practical experience from skilled disassembly workers, encoding continuous actions and potential contingencies into the symbolic space via neural predicates. Notably, after the execution of *Recognize* and before *Insert*, the neural predicate *target_rust* evaluates whether the screw is rusted or otherwise non-detachable. If the screw is in a normal condition, it is autonomously disassembled by a single robot, even in the presence of obstacles or when sleeve replacement is required. This is because the MAU's action primitives competency library contains the complete set of primitives required for detachable connections, enabling real-time autonomous planning of appropriate disassembly sequences. If a non-detachable connection is detected, replanning is triggered online based on the current state. The resulting action sequence containing the *Cut* primitive activates the multi-robot task assignment mechanism shown in Fig. 3.

Accordingly, the potential single-robot and multi-robot reasoning processes for this long-horizon task are illustrated in Fig. 5. Neural predicate checks preceding each action primitive perform high-frequency validation of state consistency. Under extreme environmental conditions or disturbances, replanning allows action primitive execution to be repeated multiple times. Leveraging multimodal sensor information, this replanning process not only ensures trustworthy execution but also continuously accumulates data for continuous learning. These aspects are further discussed in Section 4.3.

4.2. Reactive system: Efficient manipulation driven by KINNs

4.2.1. The manipulation challenge

The Deliberative System focuses on high-level task planning, which is a key capability for hybrid-control embodied systems to handle the structural diversity and variability inherent in power battery disassembly. In contrast, the Reactive System addresses low-level motion execution, particularly manipulation-related challenges arising during the disassembly process. In complex robotic manipulation tasks, the system must simultaneously satisfy high-level semantic constraints at the task layer and physical feasibility constraints at the motion layer. The former typically encodes the intended interaction between the robot and the environment, such as target positions, contact directions, or grasp poses, while the latter concerns low-level execution limits, including kinematic singularities, joint limits, and inter-link collision constraints. As these multi-level constraints accumulate, the robot's feasible configuration space is progressively compressed into a high-dimensional, continuous, and constrained differential manifold. The resulting complexity renders conventional global search or simulation-based planning methods inefficient. Fundamentally, this problem concerns the alignment between planning outputs generated by the Deliberative System and executable control signals in the Reactive System. When multiple constraints are imposed simultaneously, the key question becomes how to efficiently compute configurations that enable effective manipulation. In this context, the Reactive System is embodied in the control policy $\pi_\alpha(u \mid w)$ associated with each action primitive, as defined in Eq. (3).

To realize an end-to-end Reactive System without relying on solver-based planning in simulation, we formulate the problem as follows. During manipulation, after the convergence of multi-level constraints, the system faces a configuration-space sampling problem. These constraints are categorized into two types: (i) a goal manifold, defined by task intent (e.g., alignment with disassembly points or end-effector pose consistency), and (ii) a feasibility envelope, determined by the interaction between the robot embodiment and the environment, including joint limits and the existence of kinematic solutions. Accordingly, the final feasible configuration space is modeled as the intersection of these two constraint sets:

$$\mathcal{S}(w) = \underbrace{\mathcal{M}_{\text{goal}}(w)}_{\text{goal manifold}} \cap \underbrace{\mathcal{E}_{\text{feas}}(w)}_{\text{feasibility envelope}} \tag{15}$$

Where w denotes the environment or task state. Here, $\mathcal{S}(w)$ represents the effective configuration space after multi-level constraint convergence; $\mathcal{M}_{\text{goal}}(w)$ denotes the goal manifold encoding task-level semantic objectives (e.g., pose and position requirements); and $\mathcal{E}_{\text{feas}}(w)$ denotes the feasibility envelope capturing kinematic constraints such as joint limits, self-collision avoidance, and singularity avoidance. This formulation can be illustrated using a specific action primitive. Consider the *Insert* primitive shown in Fig. 4, which requires the robotic arm to approach a screw and fit the sleeve onto it. This process involves two primary constraints: the sleeve pose must match the screw pose, and the robot kinematics must admit a valid solution at the insertion location. Under these constraints, the robot's configuration space is compressed, and the resulting constrained differential manifold constitutes $\mathcal{S}(w)$.

As the robot's degrees of freedom(DOF) increase, configuration sampling becomes increasingly challenging. This is particularly true for the MAU, which combines a mobile base with 3-DOF and a manipulator with 6-DOF. In such cases, we adopt the principles of PINNs by embedding robotic kinematics directly into end-to-end network training. This enables efficient and trustworthy Reactive System control under complex, multi-constraint configuration spaces. It is important to emphasize that, under the rigorous problem formulation described above, the Reactive System addresses low-level kinematic solving rather than trajectory-level motion planning. KINNs does not involve path search algorithms (e.g., RRT-based planners). Instead, it directly maps task-level pose and multi-level constraints to feasible configurations on a constrained kinematic manifold.

4.2.2. KINNs encoder–decoder architecture

The goal of the Reactive System is to accelerate the execution of each action primitive by directly outputting trustworthy kinematic configurations through an end-to-end network. To this end, we design a PINN-style encoder–decoder architecture, as illustrated in Fig. 6. The encoder predicts joint configurations based on the desired end-effector pose, while the decoder is implemented as a forward kinematics module via differential programming, enforcing physical consistency during training.

Unlike conventional PINNs, which are primarily designed to solve partial differential equations by enforcing physics residual constraints, the proposed KINN embeds the complete analytical kinematic chain (DH-based kinematics and its differentiable structure) directly into the computational graph. The network is structurally constrained by exact kinematic mappings during optimization. Therefore, KINN is not a PDE solver, but a knowledge-embedded kinematic learning framework tailored for high-DOF robotic systems.

Specifically, the Reactive System must account for both MAU and SAU execution. Accordingly, a control-side flag is included in the training input to distinguish between robot types. For the Whole-Body Control (WBC) Predictor (Encoder), the input consists of the desired

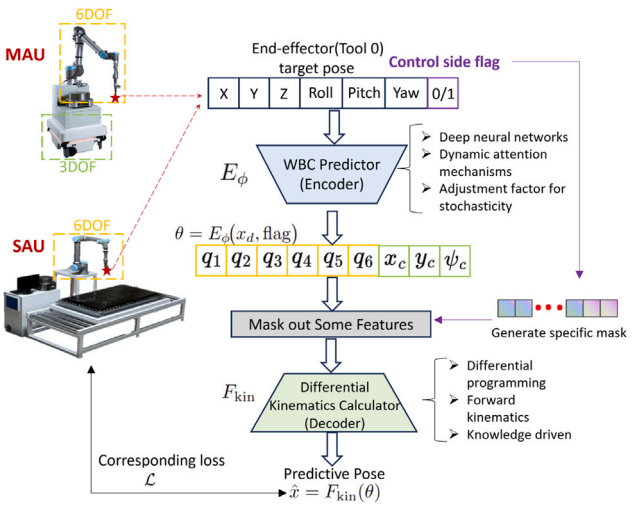

Fig. 6. Reactive System Training Pipeline. (KINNs)An encoder–decoder architecture, where the input is the pose of the object to be disassembled, and the end-to-end network directly outputs the control configuration. This design enables efficient and trustworthy robot control under complex configuration spaces with multi-level constraints.

end-effector pose and the control-side indicator, reflecting ideal task-level constraints. The encoder directly outputs the corresponding joint configuration:

$$\theta = E_\phi(x_d, \text{flag}) = \begin{cases} [q_1, \ldots, q_6]^\top, & \text{flag} = 0, \\ [q_1, \ldots, q_6,\ x_c,\ y_c,\ \psi_c]^\top, & \text{flag} = 1, \end{cases} \tag{16}$$

Where $x_d = [\mathbf{p}_d,\ \boldsymbol{\phi}_d] \in \mathbb{R}^6$ denotes the desired end-effector pose (position $\mathbf{p}_d$ and RPY orientation $\boldsymbol{\phi}_d$), flag $\in \{0, 1\}$ indicates the robot type, and E_ϕ is the WBC Predictor (encoder network) producing the joint configuration θ.

Neural network training often requires large datasets with ground-truth labels, and learning is driven by minimizing discrepancies between predictions and labels. This tightly couples prediction accuracy to dataset scale and quality. Moreover, from differential geometry, constrained manifolds (Eq. (15)) remain continuous, implying that discrete datasets cannot adequately cover the entire solution space, leading to long-tail sparsity issues. Our approach instead enables the end-to-end network to learn robot physical constraints rather than dataset-specific mappings. The encoder is implemented as a multi-layer perceptron (MLP), which is sufficient since the primary complexity lies in the subsequent differentiable kinematics module. After the WBC Predictor outputs joint configurations, we introduce a Differential Kinematics Calculator (Decoder) implemented via differential programming [74, 75], ensuring continuity of the computational graph. Specifically, forward kinematics and the associated Jacobian are embedded as differentiable operations, allowing gradients to propagate through all kinematic transformations:

$$\hat{x} = F_{\text{kin}}(\theta) \tag{17}$$

Where F_{kin} denotes the forward kinematics decoder and $\hat{x} = [\hat{\mathbf{p}},\ \hat{\boldsymbol{\phi}}]$ is the predicted end-effector pose. Prior physical knowledge is embedded in this stage, guiding the network to respect motion-level constraints. The decoder output is evaluated using the loss function:

$$\mathcal{L} = \| \hat{\mathbf{p}} - \mathbf{p}_d \|^2 + \lambda \| \hat{\boldsymbol{\phi}} - \boldsymbol{\phi}_d \|^2 \tag{18}$$

The loss function $\mathcal{L}$ quantifies the discrepancy between the predicted pose and the target. The coefficient λ balances positional and postural errors, which are first normalized to comparable magnitudes. In practice, λ is selected within a stable range via hyperparameter

search (Ray Tune [80]), and empirical results indicate low sensitivity within this range. Backpropagation is then executed to complete a training iteration. Owing to differential programming techniques, the computational graph remains coherent throughout, enabling gradient propagation via the chain rule.

4.2.3. Extensibility and future reactive system designs

Fig. 6 illustrates the core principles of the KINNs encoder–decoder architecture. Forward kinematics constraints suffice for certain action primitives, such as $Recognize$ and $Insert$, which are implemented in this work. Importantly, the framework is highly extensible: different complex constraints can be accommodated by modifying the decoder's differential programming formulation.

Accordingly, additional Reactive System strategies are deployed. For example, when optimizing the MAU chassis pose during motion, the ideal configuration should both enable arm-level disassembly and maintain a safe distance from the target object. Beyond WBC, we design a decoupled control (DC) strategy, which is applied to primitives such as $Move$ for MAU. This involves two modifications: (i) the Predictor Encoder outputs planar chassis position and yaw $\{x_c,\ y_c,\ \psi_c\}$, and (ii) the forward kinematics decoder F_{kin} is replaced by an inverse kinematics decoder I_{kin} to ensure the existence of kinematic solutions. The corresponding loss function is redefined to reflect base collision avoidance and inverse kinematic solvability constraints. Experimental of WBC and DC are presented in the next section. Compared with random sampling, neural network-based inverse kinematics learning, and deep reinforcement learning methods, the proposed knowledge-guided KINNs encoder–decoder architecture achieves substantially higher data efficiency and accuracy. Unlike purely data-driven approaches that approximate joint-task mappings from embodied datasets, KINNs are structurally constrained by differentiable, gradient-based modular kinematic computations embedded in the training process. As validated in Section 5.2, this knowledge-guided formulation leads to consistently improved performance under high-DOF configuration spaces. Notably, training does not depend on the quality of embodied datasets; instead, deterministic kinematic structure provides stable supervision, thereby enhancing both generalization capability and the trustworthiness of the Reactive System.

In future Reactive System designs, additional constraints will be incorporated. Differential programming and tailored loss definitions will enable implicit alignment of richer physical knowledge within the NeuroSymbolic architecture. This approach allows end-to-end Reactive networks to genuinely learn complex constraints rather than merely fitting relationships present in supervised datasets. We will further extend the design to a broader range of disassembly action primitives, enhancing the generalization capability of trustworthy Reactive Systems across diverse scenarios.

4.3. Multimodal cross-validation and continual learning

In the Deliberative System, the Symbol Check Agent (Fig. 3) serves as a key mechanism for ensuring reliability at the task-planning level. In addition, we introduce a complementary trustworthiness guarantee at the primitive execution level. This design stems from the inherent redundancy and robustness embedded in both the action primitive definitions and the PDDL state-space formulation. As illustrated in Fig. 5, two action primitives are defined for estimating the pose of the object to be disassembled: Aim_V and Aim_F. In practical robotic execution, reliance on a single perceptual modality is often insufficient to cope with disturbances in complex environments. Sensor noise can induce fluctuations in perception results, thereby degrading the overall stability and accuracy of task execution. To maintain reliable decision-making in dynamic scenarios, we design a vision–force cross-validation mechanism [81], which exploits the complementary nature of heterogeneous sensory modalities to continuously correct perceptual deviations. Specifically, the visual module processes RGB-D inputs acquired from cameras, while the force-sensing module extracts force signal variations generated during exploratory contact. By fusing and interacting across these two modalities, the system continuously updates its neuro predicates. The real-time information from both modules is further provided to the action primitives Aim_V and Aim_F as pose estimation inputs.

A representative example is illustrated as follows. Consider a single robot performing a screw disassembly task (Fig. 5). Given the initial state and the goal state, the multi-agent trustworthy planner produces an optimal action primitive sequence:

$$\{Move, Aim\_V, Recognition, Insert, Disassemble\}$$

Each action primitive in the sequence is dispatched sequentially, and its corresponding preconditions are verified prior to execution according to the definitions in Fig. 4. If the action primitive Aim_V suffers from perceptual errors due to environmental disturbances, the system detects a failed sleeve fitting before executing $Disassemble$. In this case, the neuro predicate $Clamped$ evaluates to False, which triggers replanning. Since the state predicate *pattern* remains True at this stage, the replanning result becomes:

$$\{Aim\_F, Insert, Disassemble\}$$

Thereby activating a force-dominant action primitive for pose estimation. This cross-validation process is repeatedly executed until all neuro predicates are satisfied under ideal conditions.

During the cross-validation process, all sensory and decision data are logged and retained for continual learning. For tasks that are ultimately successful but require replanning, we further introduce a backward correction mechanism. For example, when visual perception fails and the task is instead completed through force-based execution, the neuro predicate *target_aim* along the visual decision pathway is flagged as unreliable. This is because its inaccuracy or misclassification leads the system to proceed with subsequent action primitives, such as $Insert$, under incorrect assumptions. Accordingly, during backward correction, the historical inputs associated with *target_aim* are relabeled with opposite annotations, indicating that the neural network should have classified these states as misaligned. Such data are continuously accumulated during execution and, once a sufficient volume is reached, are used to update the corresponding neuro predicates. This iterative refinement progressively enhances their abstraction capability and improves robustness to environmental variations.

Furthermore, system trustworthiness can be quantitatively defined, as each aforementioned module and component contributes to the overall reliability of the system. To quantitatively characterize system trustworthiness, we introduce the Rule Coverage Ratio (RCR) and the Runtime Rule Compliance Rate (RRCR) (see Appendix B), both defined based on Checkable Prior Rule Set (CPRS) (see Appendix A). All task-critical operations in battery disassembly are governed by 24 checkable prior rules encoded in PDDL and neural predicates, resulting in an RCR of 100%. Across more than 200 real-world trials, the RRCR also reached 100%, demonstrating complete logical compliance during execution. Unlike end-to-end policy-based approaches, our framework supports explicit runtime verification at every task-critical checkpoint.

5. Experimental verification

To validate the proposed embodied robotic disassembly system and its associated techniques, this section presents comprehensive algorithm-level experiments and reports their deployment on a real-world power battery disassembly production line. As shown in Fig. 7, our industrial power battery disassembly line consists of multiple workstations. The figure illustrates a representative workstation scenario in which multiple robots collaborate to complete the disassembly task. The system mainly comprises four components: a disassembly workstation, waste power batteries, a MAU, and a SAU. For both the MAU and SAU, we employ UR10e robotic manipulators (6-DoF) with a payload capacity of 12.5 kg and a working radius of 1300 mm.

Table 1
Robotic workstation controller parameters (edge computing device).

Hardware/software	SAU	MAU
CPU	Intel i7-12700	Intel i7-11700
GPU	GeForce RTX 3060 Ti	Mesa Intel Graphics
RAM	64G	16G
Operating System	Ubuntu 22.04.5 LTS	Ubuntu 22.04.5 LTS
Robot Operating System	ROS Noetic	ROS Noetic
AI Framework	Pytorch	Pytorch

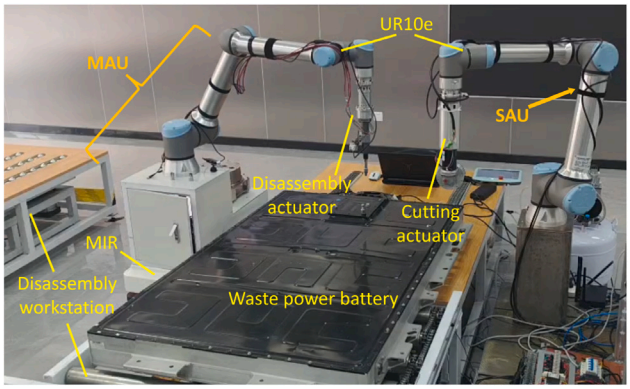

Fig. 7. Hardware platform for experiments: industrial power battery disassembly line.

The MAU is mounted on a MiR-100 mobile base, forming a mobile manipulator. Both UR10e manipulators are equipped with ATI six-axis force/torque sensors at the flange as part of the end-effector assembly, providing real-time force feedback during disassembly. The end-effectors are selected as shown in Fig. 2 and are further equipped with Intel RealSense D405/D415 RGB-D cameras. Based on the functional configurations of the end-effectors, the resulting robot action primitive capability library is illustrated in Fig. 4.

For the MAU and SAU, we select appropriate edge computing devices as the supervisory controllers at each workstation and deploy the complete embodied system on them. Detailed hardware and software specifications of the computing platforms are summarized in Table 1.

5.1. Multi-agent trustworthy planning in the deliberative system

To evaluate the proposed multi-agent trustworthy planning framework, we adopt GPT-4 as the backbone model for the agents and construct the Deliberative System of the Disassembly NeuroSymbolic World Model, targeting the detachable connections of power battery fasteners. In this experimental setting, ablation studies are conducted to systematically verify that the introduced multi-agent design is non-redundant and efficient. Among the agents, the Symbol Check Agent and the Task Assignment Agent share similar roles, as both rely on invoking external PDDL checkers or symbolic mapping functions to accomplish their tasks. Consequently, the specific prompt formulations and linguistic quality of these agents have limited impact on the overall planning accuracy. Instead, they serve as critical industrial-grade safeguards at the symbolic level, reinforcing system trustworthiness and closing the planning loop. In contrast, the prompt design and reasoning quality of the Heuristic Search Agent require particular attention. Although heuristic search alleviates the inefficiency of exhaustive symbolic search or brute-force enumeration, it also constitutes the primary source of potential hallucinations at the task-planning level. Low-quality outputs from the Heuristic Search Agent may propagate erroneous information to downstream agents, resulting in repeated iterations and degraded planning performance.

In the Heuristic Search Agent, the prompt is structured into six major components, among which the description, specification, and

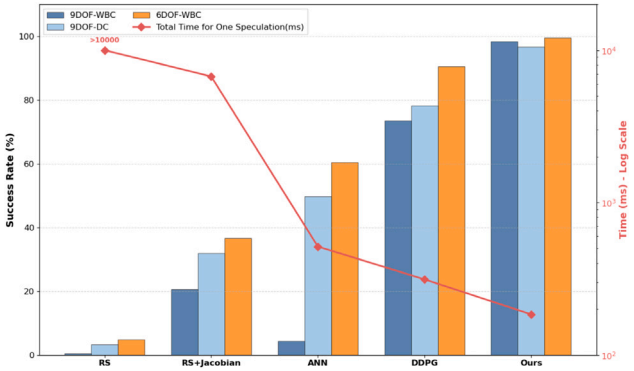

Fig. 8. Comparison of success rate and efficiency across methods.

problem formulation of action primitives are indispensable. To analyze the contributions of the remaining components, we conducted 150 randomly sampled single-step planning trials for each configuration. Furthermore, we designed 200 random reasoning tasks that allow multi-agent systems to perform multiple iterations. The experimental results are summarized in Table 2. For the heuristic search process, character descriptions and scene descriptions do not significantly promote the generation of feasible action sequences. In contrast, the inclusion of priori examples enables the agent to achieve a single-shot reasoning success rate of 58.5%. For the Deliberative System with multi-agent collaboration, introducing symbolic-level verification increases the success rate to 99.6%. This is because, although PDDL provides strict guarantees in the symbolic space, omitting the Task Assignment Agent neglects the safeguards required for Multi-robot Collaboration. When all agents participate jointly in the reasoning process, the Deliberative System attains a 100% accuracy rate, yielding trustworthy and transparent optimal action sequences at the task-planning level.

5.2. Motion planning in the reactive system

To evaluate the accuracy and efficiency of the Reactive System under prior-knowledge guidance, we conducted experiments across three control scenarios, each compared against five different methods: whole-body control for a 9-DoF MAU (9DoF-WBC), decoupled control for a 9-DoF MAU (9DoF-DC), and whole-body control for a 6-DoF SAU (6DoF-WBC). For computing motion configurations under multiple constraints, we evaluated random sampling (RS), random sampling with Jacobian iteration (RS + Jacobian), artificial neural networks (ANN), deep deterministic policy gradient (DDPG), and the proposed method (Ours). For the ANN baseline, 50,000 analytical DH-based kinematic solutions from diverse scenarios were collected and used to train the network for 3000 epochs until convergence. For DDPG, deterministic kinematic solvers were employed as a resampling strategy within the corresponding control dimensions. The replay buffer initially contained 30,000 samples and was augmented with 512 newly resampled data points per iteration. All sampling-based methods were configured with termination tolerances and maximum iteration limits, while all learning-based methods were optimized using Ray Tune [80] over five learning rates and four batch sizes to identify the best-performing configurations. The quantitative results are presented in Fig. 8.

Due to its purely random nature, the RS method achieves a maximum accuracy of only 4.84%, with an average total computation time per speculation exceeding the 10,000 ms timeout threshold. Incorporating Jacobian iteration improves the accuracy to 36.61%, with an average computation time of 6726.17 ms. In comparison, ANN learns kinematic mappings from data, and DDPG leverages deep reinforcement learning to approximate deterministic policies by maximizing

Table 2
Success rate and ablation experiments for key components in multi-agent task planning.

	Heuristic search agent			Symbol check agent	Task assignment agent	Success rate
	Priori example	Character description	Scene description			
Single Inference	–	–	–	–	–	30.1%
	–	–	✓	–	–	37.2%
	–	✓	–	–	–	22.1%
	✓	–	–	–	–	58.5%
	–	✓	✓	–	–	24.2%
	✓	✓	–	–	–	53.9%
	✓	–	✓	–	–	50.3%
Multiple Iterations		✓		–	–	68.4%
		✓		✓	–	**99.6%**
		✓		–	✓	53.6%
		✓		✓	✓	**100.0%**

cumulative rewards, achieving accuracies of 60.49% and 90.47%, respectively. Notably, the proposed method attains accuracies of 98.36% and 99.45% in the 9DoF-WBC and 6DoF-WBC settings, respectively, while also exhibiting the fastest inference time of 184.76 ms.

Deep analysis of the experimental data reveals that the 6DoF setting achieves the highest success rate across all methods, whereas the 9DoF-WBC setting exhibits relatively lower performance. This is primarily due to the dramatic expansion of the solution space introduced by higher DoF, while WBC is inherently more difficult to predict in terms of ideal configurations compared to DC strategies. From sampling-based approaches to learning-based methods, the overall success rate shows a consistent upward trend, reflecting the progressive incorporation of more effective algorithmic components. Specifically, Jacobian-based methods introduce numerical computation, ANN methods incorporate mappings derived from real data, DDPG introduces reward-penalty mechanisms for policy optimization, and our proposed method further integrates explicit physical constraints. In contrast, improvements in efficiency are reflected in the continuous reduction of inference time. In particular, end-to-end methods demonstrate significantly faster performance compared with numerical iterative approaches.

These results demonstrate that knowledge-injected Reactive System learning enables the network to internalize deterministic physical principles, rather than merely capturing latent correlations from datasets. Consequently, the method achieves substantially higher accuracy and remains robust to out-of-distribution scenarios. Moreover, owing to its more compact network architecture and improved precision, the average inference time per query is further reduced.

5.3. Cross-validation and continual learning

To validate the effectiveness of the vision–force cross-validation framework, we conducted experiments on the disassembly of cover screws from power battery packs. In each trial, the objective was to successfully remove a single screw. The system leverages vision–force cross-validation to obtain more reliable neuro predicates and perception estimates. During the experiments, the system dynamically updates both the neuro predicates and the screw pose estimation based on visual and force sensory feedback. To emulate real-world uncertainty, disturbances such as illumination variations were introduced. Replanning is triggered whenever an action fails. If the number of replanning attempts exceeds a predefined threshold, the task is considered a failure. For tasks that eventually succeed but require replanning, the system generates new samples through cross-modal correction (e.g., using force feedback to rectify visual errors) and neuro predicate backtracking, and stores them in a data pool. Once a sufficient number of samples has been accumulated, incremental training is triggered to further improve perception stability.

The results, summarized in Table 3, demonstrate that vision–force cross-validation significantly enhances system robustness in dynamic environments. Visual perception is particularly sensitive to environmental factors such as lighting changes, and inaccurate screw pose predictions were the primary cause of task failures and frequent replanning in early experiments. In contrast, force perception is more stable during contact interactions and can provide reliable compensation when visual perception degrades. Moreover, vision–force cross-validation supplies high-quality training samples for continual learning, facilitating long-term self-improvement of the system. As continual learning progresses and the system iteratively updates, the task success rate improves from 81.68% to 100%, while the average number of replanning events steadily decreases. Thus, cross-validation not only enhances immediate perceptual robustness but also establishes a stable data foundation for continual learning. Quantitative ablation results (Appendix C) show that vision-force fusion improves the initial task success rate from 72.39% (vision only) and 70.51% (force only) to 81.68%, demonstrating a clear complementary effect. Moreover, as shown in Appendix C, both the vision and force networks exhibit monotonic error reduction during incremental training, indicating stable convergence of model parameter updates in continual learning.

Table 3
Under the cross-validation mechanism, the collected data are continuously utilized for continual learning.

Iteration	Number of tasks	Success rate (%)	Avg. replans per task
0	131	81.68	3.389
1	162	89.51	2.648
2	62	100	1.984
3	39	100	1.128

5.4. Real-world experiments

After extensive algorithmic validation, the proposed industrial-grade trustworthy embodied system was deployed on a real power battery disassembly production line. We designed both single-robot tasks and multi-robot collaboration tasks (Fig. 9). Single-robot tasks include large-scale autonomous disassembly, disassembly under scheduling constraints, disassembly with obstacle interference, and disassembly involving multiple types of screws requiring sleeve changes. Multi-robot collaboration tasks focus on the removal of rusted or worn screws. These complex long-horizon tasks comprehensively evaluate the system's real-time perception, planning, execution, and verification capabilities.

The empirical samples consist of multiple wasted power battery packs collected from real industrial recycling scenarios. These battery packs exhibit heterogeneous physical conditions, including bolt rust, outer nut edge wear, and inconsistent illumination reflections caused by metallic surfaces. Notably, severely rusted or worn bolts may lose their detachable mechanical characteristics and cannot be removed through standard unscrewing operations. Such cases correspond directly to the non-detachable long-horizon experimental tasks considered in our system. As illustrated in Fig. 10, a long-horizon task example shows the action sequence and state transitions generated through real-time multi-agent planning. In addition to material

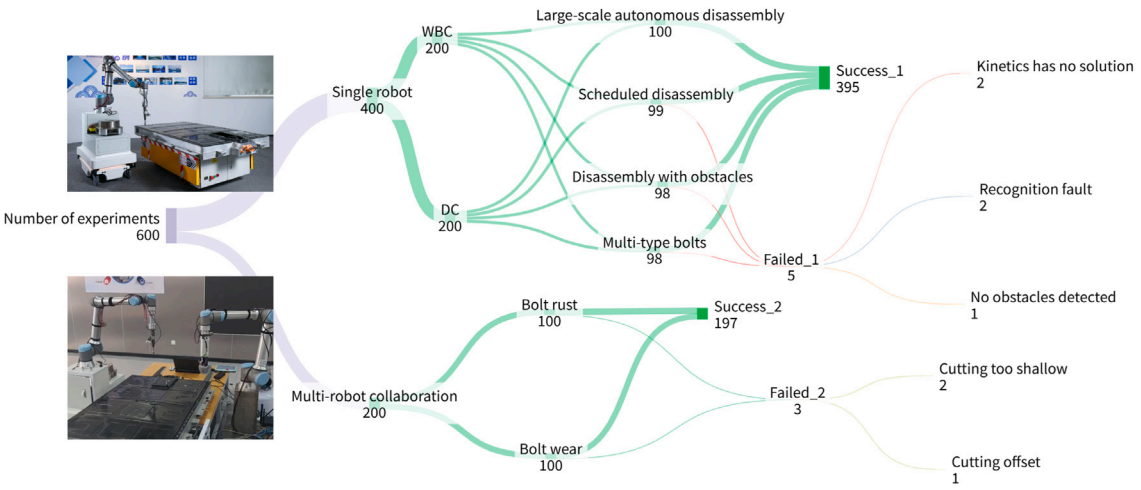

Fig. 9. Real-Robot Deployment Results. The experiments were conducted on real robotic platforms under both single-robot autonomous disassembly tasks and multi-robot collaboration tasks.

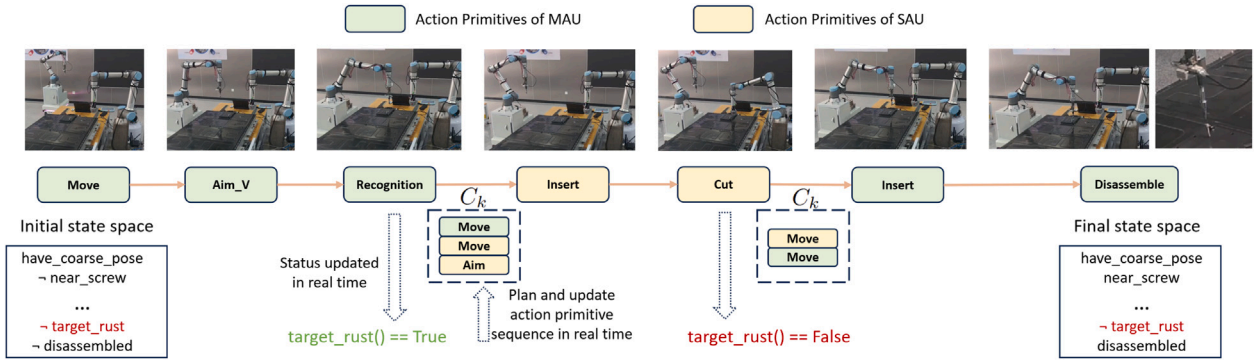

Fig. 10. A representative Long-horizon task example involves cooperation between the SAU and the MAU to remove a worn screw. Compared with standard disassembly planning, this process additionally requires operations such as cutting.

degradation, the experiments also include physical disturbance factors such as obstacle interference, dynamic lighting variations affecting visual neural predicates, and occasional policy failures leading to action execution deviations. These conditions further validate the Real-time Status Checking and online replanning capabilities of the NeuroSymbolic framework, enabling the robotic system to maintain robustness under realistic industrial perturbations.

Experimental results indicate that the success rate of single-robot tasks reaches 98.75%, while multi-robot collaboration tasks achieve a success rate of 98.5%. This high level of performance stems from the fact that the Disassembly NeuroSymbolic World Model fundamentally learns disassembly skills rather than overfitting to specific battery pack models. Consequently, the system exhibits strong generalization capability, enabling the embodied system to flexibly plan and complete disassembly tasks in real time by leveraging prior knowledge and available tools.

6. Conclusion and future work

This work addresses the core challenges in power battery disassembly, including high uncertainty, significant structural variability, and the frequent failure of traditional automation. We propose and implement an industrially deployable, trustworthy NeuroSymbolic embodied intelligence system. By leveraging NeuroSymbolic artificial intelligence, the system establishes a dual-system hybrid control architecture, in which the Deliberative System enables knowledge-driven long-horizon autonomous reasoning and multi-robot collaboration planning, while the Reactive System achieves efficient, fast, and verifiable low-level control based on kinematic priors. The coordinated interaction between the two systems ensures both reliability and real-time performance. In addition, we design both mobile and stationary embodied execution units, providing flexible physical carriers for complex disassembly tasks. Through a dual strategy driven by knowledge and data, together with symbolic planning, cross-modal verification, and continual learning mechanisms, the Disassembly NeuroSymbolic World Model enables stable, safe, and interpretable operational workflows in open and unstructured disassembly scenarios. This work lays a solid foundation for large-scale intelligent disassembly in the battery recycling industry and provides a demonstrative pathway for the trustworthy deployment of embodied intelligence in industrial environments.

Looking forward, several directions merit further improvement and investigation:

1. Enhancing multi-robot collaboration efficiency, extending from single-station, centralized planning to multi-station, efficient hierarchical planning for flexible disassembly production lines.
2. Further strengthening and extending the Reactive System, evolving from kinematics based neural networks under complex constraints to a library of embodied skill learning, and incorporating additional high-frequency sensors (e.g., tactile sensing and force control) to enable the Reactive System to acquire richer disassembly skills.
3. Advancing the multimodal cross-validation and continual learning framework, moving beyond engineering-level modeling toward a more comprehensive, scalable reasoning, execution, verification, learning pipeline.

CRediT authorship contribution statement

Yanlong Peng: Writing – review & editing, Writing – original draft, Visualization, Validation, Methodology, Formal analysis. **Zhigang Wang:** Writing – review & editing, Visualization, Supervision, Methodology, Conceptualization. **Yuping Zhang:** Writing – review & editing, Supervision, Project administration, Investigation, Conceptualization. **Pengxu Chang:** Writing – review & editing, Validation, Software, Data curation. **Ziwen He:** Software, Investigation, Data curation. **Ming Chen:** Writing – review & editing, Supervision, Project administration, Methodology, Funding acquisition, Conceptualization.

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.

Appendix A. Rule-based checkpoints for trustworthy disassembly

To formally characterize system trustworthiness, we explicitly enumerate task-critical prior rules governing battery disassembly. These rules, distilled by skilled human engineers, constitute a set of a priori principles governing the smooth execution of disassembly within automated or intelligent systems. Each rule corresponds to a verifiable checkpoint, implemented either through neural predicates within the World Model or distinct system modules within the Deliberative System. All action primitives in our PDDL-based NeuroSymbolic planner are associated with these rules via preconditions and effects, ensuring that no task-critical transition can bypass verification.

These rules collectively define the Checkable Prior Rule Set (CPRS) in Fig. A.11, which serves as the structural foundation for quantifying trustworthiness. Furthermore, the relationship between the specific modules involved in this study and system trustworthiness can be clearly characterized, with different modules collectively forming the key checkpoints for trustworthiness verification.

Appendix B. Quantitative definition of trustworthiness

B.1. Rule coverage ratio (RCR)

We define trustworthiness as the ability of the system to explicitly represent and verify task-critical prior rules during execution.

Let:

$$RCR = \frac{N_{verifiable}}{N_{critical}}$$

where $N_{critical}$ denotes the total number of task-critical prior rules. $N_{verifiable}$ denotes the number of rules explicitly encoded and verifiable at runtime.

In our system:

$$N_{critical} = 24, \quad N_{verifiable} = 24$$

Thus:

$$RCR = 100\%$$

All task-critical transitions are governed by neural predicates or runtime validation modules. In contrast, end-to-end policy-based or black-box reinforcement learning methods do not explicitly encode intermediate rule checkpoints, yielding:

$$RCR_{E2E} = 0\%$$

since no internal decision structure is inspectable.

Table C.4
Task success rate under different perception modalities (Iteration 0).

Perception mode (Iteration 0)	Task success rate (%)
Vision Only	72.39
Force Only	70.51
Vision + Force (Cross-Validation)	81.68

Table C.5
Pose estimation error of the vision network under incremental training.

Training samples	Pos. RMSE (mm)	Pos. MAE (mm)	Ori. RMSE (deg)	Ori. MAE (deg)
1000 (Iter 0)	8.78	7.34	5.39	2.52
2000	3.81	3.19	3.60	2.14
3000	2.53	2.10	2.68	1.68
4000	1.71	1.43	2.12	1.30
5000	1.19	0.99	1.13	0.87

B.2. Runtime rule compliance rate (RRCR)

We further define execution-level reliability:

$$RRCR = \frac{N_{passed}}{N_{checked}}$$

where $N_{checked}$ denotes total rule checks during execution. N_{passed} denotes successfully validated checks. Over extensive industrial trials, all 24 rule types were verified in runtime execution without bypass, yielding near-perfect compliance.

Appendix C. Quantitative analysis of vision-force fusion and continual learning

This appendix provides a quantitative analysis addressing two key aspects: (1) the performance improvement introduced by vision-force fusion, and (2) the convergence characteristics of model parameter updates during continual learning.

C.1. Quantitative evaluation of vision-force fusion

To isolate the contribution of cross-modal fusion, we report the task success rate under three perception settings at Iteration 0 (i.e., before continual learning is triggered). In this stage, both the vision network (MobileNetV3-based) and the force network (LSTM-based) are trained using the initial dataset only. Regarding evaluation metrics, we employ Root Mean Square Error (RMSE) and Mean Absolute Error (MAE).

As shown in Table C.4, vision-force fusion improves the initial task success rate by +9.29% compared to vision-only perception and +11.17% compared to force-only perception. This demonstrates a clear complementary effect between the two modalities. Vision provides global geometric estimation but is sensitive to environmental disturbances (e.g., illumination changes), whereas force feedback offers stable local contact information during interaction. Their cross-validation mechanism therefore enhances robustness and reduces failure cases caused by single-modality uncertainty.

C.2. Convergence characteristics of continual learning

We further analyze the convergence behavior of both perception networks during incremental training. Continual learning is triggered when sufficient corrected samples are accumulated through cross-modal correction and neuro-predicate backtracking. The training dataset gradually expands to better match real deployment conditions.

NO.	Rule Description	Verified By
1	The coarse pose of the target must be known before task initiation.	(Neural predicate)have_coarse_pose
2	Coarse motion must be executed in local coordinate frame (eye-in-hand).	(Neural predicate)near_screw
3	Target detection must follow coarse positioning.	(Neural predicate)above_screw
4	Fine positioning should prioritize vision.	(Neural predicate)pattern
5	Force-based exploration should be triggered if vision fails.	(Neural predicate)pattern
6	Precise pose must be known before disassembly attribute judgment.	(Neural predicate)target_aim
7	Disassembly object attributes must be recognized (detachable/non-detachable).	(Neural predicate)complete_recognize
8	Rusted or mechanically damaged bolts are categorized as non-detachable.	(Neural predicate)target_rust
9	For detachable connections, sleeve must match screw type.	(Neural predicate)target_match
10	If sleeve mismatches, replanning with replacement is required.	(Neural predicate)exist_sleeve
11	Similar familiar scenes allow quick disassembly logic.	(Neural predicate)familiar_scenes
12	Before insertion, obstacle clearance must be verified.	(Neural predicate)target_clear
13	After cutting, detachability must be re-evaluated.	(Neural predicate)target_rust
14	Post-disassembly state must update to disassembled==true.	(Neural predicate)disassembled
15	Fine pose estimation must precede insertion in detachable connections.	Multimodal fusion + Kalman filter
16	Vision-force logs must be preserved and cross-validated.	Multimodal Cross-Validation
17	Task assignment must consider robot-specific capabilities.	Task Assignment Agent
18	Multi-robot coordination and collision avoidance must be considered	Task Assignment Agent
19	non-detachable connections require pose sharing among robots.	Task Assignment Agent
20	LLM outputs must be constrained by symbolic verification.	Symbol Check Agent
21	Action sequence must adapt based on neural predicate states (heuristic search).	Heuristic Search Agent
22	Planning must reference prior successful experience.	Heuristic Search Agent
23	Post-action state must match expected symbolic effects.	Real-time Status Checking
24	For execution robots, the target point should satisfy physical principles such as kinematic reachability, solvable inverse kinematics, and collision-free conditions.	KINNs

Fig. A.11. Checkable Prior Rule Set (CPRS) governing trustworthy battery disassembly.

Table C.6
Pose estimation error of the force network under incremental training.

Training samples	Pos. RMSE (mm)	Ori. RMSE (deg)
880 (Iter 0)	0.785	0.323
2760	0.595	0.287
2640	0.481	0.276
3520	0.442	0.271
4400	0.433	0.268

C.2.1. Vision network (MobileNetV3-based)

Table C.5 shows that both position and orientation errors decrease monotonically as the dataset expands. The position RMSE decreases from 8.78 mm to 1.19 mm (an 86.4% reduction), while the orientation RMSE decreases from 5.39° to 1.13° (a 79.0% reduction). The progressively flattening improvement trend indicates stable convergence of parameter updates without oscillatory behavior.

C.2.2. Force network (LSTM-based)

As shown in Table C.6, the force network also exhibits consistent error reduction during incremental training. The position RMSE decreases from 0.785 mm to 0.433 mm (approximately 44% reduction), while orientation RMSE decreases from 0.323° to 0.268° (approximately 17% reduction). The improvement trend becomes gradually saturated as the dataset approaches 4000 samples, suggesting that the model converges toward a stable solution.

C.3. Summary

The quantitative results in this appendix demonstrate that:

1. Vision-force fusion provides a clear and measurable improvement in task success rate compared to single-modality perception.
2. Both perception networks exhibit monotonic error reduction and stable convergence during continual learning.
3. The cross-validation mechanism not only enhances immediate robustness but also supplies high-quality correction samples that facilitate long-term performance improvement.

These results provide explicit quantitative evidence supporting the robustness and convergence properties of the proposed framework.

References

[1] Wu Shubiao, Kaden Nicolaj, Dröder Klaus. A systematic review on lithium-ion battery disassembly processes for efficient recycling. Batteries 2023;9(6):297.
[2] Ma Ruifei, Tao Shengyu, Sun Xin, Ren Yifang, Sun Chongbo, Ji Guanjun, Xu Jiahe, Wang Xuecen, Zhang Xuan, Wu Qiuwei, et al. Pathway decisions for reuse and recycling of retired lithium-ion batteries considering economic and environmental functions. Nat Commun 2024;15(1):7641.
[3] Etxandi-Santolaya Maite, Casals Lluc Canals, Montes Tomás, Corchero Cristina. Are electric vehicle batteries being underused? A review of current practices and sources of circularity. J Environ Manag 2023;338:117814.
[4] Dai Yixin, Panahi Aidin. Thermal runaway process in lithium-ion batteries: A review. Next Energy 2025;6:100186.

[5] Kaarlela Tero, Villagrossi Enrico, Rastegarpanah Alireza, San-Miguel-Tello Alberto, Pitkäaho Tomi. Robotised disassembly of electric vehicle batteries: A systematic literature review. J Manuf Syst 2024;74:901–21.

[6] Meng Kai, Xu Guiyin, Peng Xianghui, Youcef-Toumi Kamal, Li Ju. Intelligent disassembly of electric-vehicle batteries: a forward-looking overview. Resour Conserv Recycl 2022;182:106207.

[7] Wan Yuwei, Liu Ying, Chen Zheyuan, Chen Chong, Li Xinyu, Hu Fu, Packianather Michael. Making knowledge graphs work for smart manufacturing: Research topics, applications and prospects. J Manuf Syst 2024;76:103–32.

[8] Zhang Hengwei, Zhang Yisheng, Wang Zhigang, Zhang Shengmin, Li Huaicheng, Chen Ming. A novel knowledge-driven flexible human–robot hybrid disassembly line and its key technologies for electric vehicle batteries. J Manuf Syst 2023;68:338–53.

[9] Peng Yanlong, Wang Zhigang, Zhang Yisheng, Zhang Shengmin, Cai Nan, Wu Fan, Chen Ming. Revolutionizing battery disassembly: The design and implementation of a battery disassembly autonomous mobile manipulator robot (BEAM-1). 2024, arXiv preprint arXiv:2407.06590.

[10] Farquhar Sebastian, Kossen Jannik, Kuhn Lorenz, Gal Yarin. Detecting hallucinations in large language models using semantic entropy. Nature 2024;630(8017):625–30.

[11] Zhang Bo, Zhu Jun, Su Hang. Toward the third generation artificial intelligence. Sci China Inf Sci 2023;66(2):121101.

[12] Garcez Artur d'Avila, Lamb Luis C. Neurosymbolic AI: The 3 rd wave. Artif Intell Rev 2023;1–20.

[13] Lin Hsiao-Ying. Team with the strong, work with the wise. Computer 2023;56(7):116–20.

[14] Kahneman Daniel. Thinking, fast and slow. macmillan; 2011.

[15] Poschmann Hendrik, Brueggemann Holger, Goldmann Daniel. Disassembly 4.0: A review on using robotics in disassembly tasks as a way of automation. Chem Ing Tech 2020;92(4):341–59.

[16] Qu Mo, Pham Duc Truong, Altumi Faraj, Gbadebo Adeyemisi, Hartono Natalia, Jiang Kaiwen, Kerin Mairi, Lan Feiying, Micheli Marcel, Xu Shuihao, et al. Robotic disassembly platform for disassembly of a plug-in hybrid electric vehicle battery: a case study. Automation 2024;5(2):50–67.

[17] Chen Hao, Shen Julia. A degradation-based sorting method for lithium-ion battery reuse. PLoS One 2017;12(10):e0185922.

[18] Villagrossi Enrico, Dinon Tito. Robotics for electric vehicles battery packs disassembly towards sustainable remanufacturing. J Remanufacturing 2023;13(3):355–79.

[19] Zhang Xinyao, Eltouny Kareem, Liang Xiao, Behdad Sara. Automatic screw detection and tool recommendation system for robotic disassembly. J Manuf Sci Eng 2023;145(3):031008.

[20] Weyrich Michael, Wang Yongheng. Architecture design of a vision-based intelligent system for automated disassembly of E-waste with a case study of traction batteries. In: 2013 IEEE 18th conference on emerging technologies & factory automation. IEEE; 2013, p. 1–8.

[21] Rastegarpanah Alireza, Mineo Carmelo, Contreras Cesar Alan, Aflakian Ali, Paragliola Giovanni, Stolkin Rustam. Electric vehicle battery disassembly using interfacing toolbox for robotic arms. Batteries 2024;10(5):147.

[22] Rastegarpanah Alireza, Hathaway Jamie, Stolkin Rustam. Vision-guided mpc for robotic path following using learned memory-augmented model. Front Robot AI 2021;8:688275.

[23] Jin Piaopiao, Lin Yinjie, Song Yaoxian, Li Tiefeng, Yang Wei. Vision-force-fused curriculum learning for robotic contact-rich assembly tasks. Front Neurorobotics 2023;17:1280773.

[24] Fleischer Jürgen, Gerlitz Eduard, Rieß Simon, Coutandin Sven, Hofmann Janna. Concepts and requirements for flexible disassembly systems for drive train components of electric vehicles. Procedia CIRP 2021;98:577–82.

[25] Wu Tengfei, Zhang Zeqiang, Yin Tao, Zhang Yu. Multi-objective optimisation for cell-level disassembly of waste power battery modules in human-machine hybrid mode. Waste Manage 2022;144:513–26.

[26] Duan Liangliang, Li Jie, Bao Jinsong, Lv Jianhao, Zheng Hangbin. A MR-assisted and scene perception system for human-robot collaborative disassembly of power batteries. In: 2023 IEEE 19th international conference on automation science and engineering. IEEE; 2023, p. 1–8.

[27] Blankemeyer Sebastian, Wiens Denise, Wiese Tobias, Raatz Annika, Kara Sami. Investigation of the potential for an automated disassembly process of BEV batteries. Procedia CIRP 2021;98:559–64.

[28] Estévez Elisabet, Marcos Marga. Model-based validation of industrial control systems. IEEE Trans Ind Informatics 2011;8(2):302–10.

[29] Nguyen-Tuong Duy, Peters Jan. Model learning for robot control: a survey. Cogn Process 2011;12(4):319–40.

[30] Wensing Patrick M, Posa Michael, Hu Yue, Escande Adrien, Mansard Nicolas, Del Prete Andrea. Optimization-based control for dynamic legged robots. IEEE Trans Robot 2023;40:43–63.

[31] Abu-Dakka Fares J, Saveriano Matteo. Variable impedance control and learning—a review. Front Robot AI 2020;7:590681.

[32] Kim DongWook, Lee Sudong, Hong Tae Hwa, Park Yong-Lae. Exploration-based model learning with self-attention for risk-sensitive robot control. Npj Robot 2023;1(1):7.

[33] Garg Kunal, Zhang Songyuan, So Oswin, Dawson Charles, Fan Chuchu. Learning safe control for multi-robot systems: Methods, verification, and open challenges. Annu Rev Control 2024;57:100948.

[34] Rana Krishan, Dasagi Vibhavari, Haviland Jesse, Talbot Ben, Milford Michael, Sünderhauf Niko. Bayesian controller fusion: Leveraging control priors in deep reinforcement learning for robotics. Int J Robot Res 2023;42(3):123–46.

[35] Ma Yueen, Song Zixing, Zhuang Yuzheng, Hao Jianye, King Irwin. A survey on vision-language-action models for embodied ai. 2024, arXiv preprint arXiv:2405.14093.

[36] Brohan Anthony, Brown Noah, Carbajal Justice, Chebotar Yevgen, Chen Xi, Choromanski Krzysztof, Ding Tianli, Driess Danny, Dubey Avinava, Finn Chelsea, et al. Rt-2: Vision-language-action models transfer web knowledge to robotic control. 2023, arXiv preprint arXiv:2307.15818.

[37] Din Muhayy Ud, Akram Waseem, Saoud Lyes Saad, Rosell Jan, Hussain Irfan. Vision language action models in robotic manipulation: A systematic review. 2025, arXiv preprint arXiv:2507.10672.

[38] Matsuo Yutaka, LeCun Yann, Sahani Maneesh, Precup Doina, Silver David, Sugiyama Masashi, Uchibe Eiji, Morimoto Jun. Deep learning, reinforcement learning, and world models. Neural Netw 2022;152:267–75.

[39] Figure AI. Helix: A vision-language-action model for generalist humanoid control. Fig AI News 2024.

[40] Intelligence Physical, Black Kevin, Brown Noah, Darpinian James, Dhabalia Karan, Driess Danny, Esmail Adnan, Equi Michael, Finn Chelsea, Fusai Niccolo, et al. π0. 5: a vision-language-action model with open-world generalization. 1, (2):2025, p. 3, https://arxiv.org/abs/2504.16054.

[41] Bjorck Johan, Castañeda Fernando, Cherniadev Nikita, Da Xingye, Ding Runyu, Fan Linxi, Fang Yu, Fox Dieter, Hu Fengyuan, Huang Spencer, et al. Gr00t n1: An open foundation model for generalist humanoid robots. 2025, arXiv preprint arXiv:2503.14734.

[42] Colelough Brandon C, Regli William. Neuro-symbolic AI in 2024: A systematic review. 2025, arXiv preprint arXiv:2501.05435.

[43] d'Avila Garcez Artur, De Raedt Luc, Peter Földiak, Pascal Hitzler, Thomas Icard, Kai-Uwe Kühnberger, Risto Miikkulainen, et al. Neural-symbolic learning and reasoning: contributions and challenges. In: AAAI spring symposium series, AAAI press. 2015, p. 20–3.

[44] Bengio Yoshua, et al. From system 1 deep learning to system 2 deep learning. In: Neural information processing systems. 2019.

[45] Ren Wei, Wang Zhigang, Yang Hua, Zhang Yisheng, Chen Ming. NeuroSymbolic task and motion planner for disassembly electric vehicle batteries. J Comput Res Dev 2021;58:2604.

[46] Sun Xiaowu, Shoukry Yasser. Neurosymbolic motion and task planning for linear temporal logic tasks. IEEE Trans Robot 2024;40:2749–68.

[47] Capitanelli Alessio, Mastrogiovanni Fulvio. A framework for neurosymbolic robot action planning using large language models. Front Neurorobotics 2024;18:1342786.

[48] Lorang Pierrick, Goel Shivam, Shukla Yash, Zips Patrik, Scheutz Matthias. A framework for neurosymbolic goal-conditioned continual learning in open world environments. In 2024 IEEE. In: RSJ international conference on intelligent robots and systems. p. 12070–7.

[49] Acharya Kamal, Raza Waleed, Dourado Carlos, Velasquez Alvaro, Song Houbing Herbert. Neurosymbolic reinforcement learning and planning: A survey. IEEE Trans Artif Intell 2023;5(5):1939–53.

[50] Guo Huihui, Wu Fan, Qin Yunchuan, Li Ruihui, Li Keqin, Li Kenli. Recent trends in task and motion planning for robotics: A survey. ACM Comput Surv 2023;55(13s):1–36.

[51] Zhao Zhigen, Cheng Shuo, Ding Yan, Zhou Ziyi, Zhang Shiqi, Xu Danfei, Zhao Ye. A survey of optimization-based task and motion planning: From classical to learning approaches. IEEE/ASME Trans Mechatronics 2024.

[52] Garrett Caelan Reed, Chitnis Rohan, Holladay Rachel, Kim Beomjoon, Silver Tom, Kaelbling Leslie Pack, Lozano-Pérez Tomás. Integrated task and motion planning. Annu Rev Control Robot Auton Syst 2021;4(1):265–93.

[53] Toussaint Marc. Logic-geometric programming: An optimization-based approach to combined task and motion planning. In: IJCAI. 2015, p. 1930–6.

[54] Kaelbling Leslie Pack, Lozano-Pérez Tomás. Integrated task and motion planning in belief space. Int J Robot Res 2013;32(9–10):1194–227.

[55] Antonyshyn Luke, Silveira Jefferson, Givigi Sidney, Marshall Joshua. Multiple mobile robot task and motion planning: A survey. ACM Comput Surv 2023;55(10):1–35.

[56] Zhang Hejia, Chan Shao-Hung, Zhong Jie, Li Jiaoyang, Kolapo Peter, Koenig Sven, Agioutantis Zach, Schafrik Steven, Nikolaidis Stefanos. Multi-robot geometric task-and-motion planning for collaborative manipulation tasks. Auton Robots 2023;47(8):1537–58.

[57] Motes James, Sandström Read, Lee Hannah, Thomas Shawna, Amato Nancy M. Multi-robot task and motion planning with subtask dependencies. IEEE Robot Autom Lett 2020;5(2):3338–45.

[58] Bui Hoang-Dung. A survey of multi-robot motion planning. 2023, arXiv preprint arXiv:2310.08599.

[59] Lai Matthew, Go Keegan, Li Zhibin, Kröger Torsten, Schaal Stefan, Allen Kelsey, Scholz Jonathan. RoboBallet: Planning for multirobot reaching with graph neural networks and reinforcement learning. Sci Robot 2025;10(106):eads1204.

[60] Ahn Michael, Brohan Anthony, Brown Noah, Chebotar Yevgen, Cortes Omar, David Byron, Finn Chelsea, Fu Chuyuan, Gopalakrishnan Keerthana, Hausman Karol, et al. Do as I can, not as I say: Grounding language in robotic affordances. 2022, arXiv preprint arXiv:2204.01691.

[61] Aghzal Mohamed, Plaku Erion, Stein Gregory J, Yao Ziyu. A survey on large language models for automated planning. 2025, arXiv preprint arXiv:2502.12435.

[62] Raissi Maziar, Perdikaris Paris, Karniadakis George E. Physics-informed neural networks: A deep learning framework for solving forward and inverse problems involving nonlinear partial differential equations. J Comput Phys 2019;378:686–707.

[63] Luo Kuang, Zhao Jingshang, Wang Yingping, Li Jiayao, Wen Junjie, Liang Jiong, Soekmadji Henry, Liao Shaolin. Physics-informed neural networks for PDE problems: a comprehensive review. Artif Intell Rev 2025;58(10):1–43.

[64] Chiu Pao-Hsiung, Wong Jian Cheng, Ooi Chinchun, Dao My Ha, Ong Yew-Soon. CAN-PINN: A fast physics-informed neural network based on coupled-automatic–numerical differentiation method. Comput Methods Appl Mech Engrg 2022;395:114909.

[65] Peng Yanlong, Wang Zhigang, Zhang Yisheng, Chang Pengxu, He Ziwen, Gu Kai, Zhang Hongshen, Chen Ming. RobKiNet: Robotic kinematics informed neural network for optimal robot configuration prediction. 2024, arXiv preprint arXiv:2402.16281.

[66] Zhou Yucan, Hu Qinghua, Wang Yu. Deep super-class learning for long-tail distributed image classification. Pattern Recognit 2018;80:118–28.

[67] Zhou Weitao, Cao Zhong, Deng Nanshan, Liu Xiaoyu, Jiang Kun, Yang Diange. Dynamically conservative self-driving planner for long-tail cases. IEEE Trans Intell Transp Syst 2022;24(3):3476–88.

[68] Zhang Chiyuan, Vinyals Oriol, Munos Remi, Bengio Samy. A study on overfitting in deep reinforcement learning. 2018, arXiv preprint arXiv:1804.06893.

[69] Liu Jingyue, Borja Pablo, Della Santina Cosimo. Physics-informed neural networks to model and control robots: A theoretical and experimental investigation. Adv Intell Syst 2024;6(5):2300385.

[70] Wang Xing, Dabrowski Joel Janek, Pinskier Josh, Liow Lois, Viswanathan Vinoth, Scalzo Richard, Howard David. PINN-ray: A physics-informed neural network to model soft robotic fin ray fingers. In: 2024 IEEE/RSJ international conference on intelligent robots and systems. IEEE; 2024, p. 247–54.

[71] Peng Yanlong, Wang Zhigang, He Ziwen, Chang Pengxu, Zhou Chuangchuang, Yan Yu, Chen Ming. Decoding RobKiNet: Insights into efficient training of robotic kinematics informed neural network. 2025, arXiv preprint arXiv:2509.07646.

[72] Qureshi Ahmed H, Dong Jiangeng, Choe Austin, Yip Michael C. Neural manipulation planning on constraint manifolds. IEEE Robot Autom Lett 2020;5(4):6089–96.

[73] Kim Seungsu, Perez Julien. Learning reachable manifold and inverse mapping for a redundant robot manipulator. In: 2021 IEEE international conference on robotics and automation. IEEE; 2021, p. 4731–7.

[74] Christodoulou Sebastian, Naumann Uwe. Differentiable programming: Efficient smoothing of control-flow-induced discontinuities. 2023, arXiv preprint arXiv:2305.06692.

[75] Liao Hai-Jun, Liu Jin-Guo, Wang Lei, Xiang Tao. Differentiable programming tensor networks. Phys Rev X 2019;9(3):031041.

[76] Hellmuth Jan F, DiFilippo Nicholas M, Jouaneh Musa K. Assessment of the automation potential of electric vehicle battery disassembly. J Manuf Syst 2021;59:398–412.

[77] Wegener Kathrin, Andrew Stefan, Raatz Annika, Dröder Klaus, Herrmann Christoph. Disassembly of electric vehicle batteries using the example of the Audi Q5 hybrid system. Procedia Cirp 2014;23:155–60.

[78] Asif Mohammed Eesa, Rastegarpanah Alireza, Stolkin Rustam. Robotic disassembly for end-of-life products focusing on task and motion planning: A comprehensive survey. J Manuf Syst 2024;77:483–524.

[79] Li Ruiya, Pham Duc Truong, Huang Jun, Tan Yuegang, Qu Mo, Wang Yongjing, Kerin Mairi, Jiang Kaiwen, Su Shizhong, Ji Chunqian, et al. Unfastening of hexagonal headed screws by a collaborative robot. IEEE Trans Autom Sci Eng 2020;17(3):1455–68.

[80] Liaw Richard, Liang Eric, Nishihara Robert, Moritz Philipp, Gonzalez Joseph E, Stoica Ion. Tune: A research platform for distributed model selection and training. 2018, arXiv preprint arXiv:1807.05118.

[81] He Ziwen, Wang Zhigang, Peng Yanlong, Chang Pengxu, Yang Hong, Chen Ming. Embodied intelligence in disassembly: Multimodal perception cross-validation and continual learning in neuro-symbolic TAMP. 2025, arXiv:2509.11270.