Journal of Manufacturing Systems 86 (2026) 264–276

Contents lists available at ScienceDirect

Journal of Manufacturing Systems

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

Technical paper

Optimizing PCBA e-waste management: Intelligent inspection sequencing and recovery strategies using graph neural networks and Reinforcement Learning

Florian Stamer [a,*], Gisela Lanza [b], Stefano Puttero [c], Maurizio Galetto [c]

[a] Institute for Production Technology and -systems (IPTS), Leuphana University, Universitätsallee 1, Lüneburg 21335, Germany
[b] wbk Institute of Production Science, Karlsruhe Institute of Technology (KIT), Kaiserstr. 12, Karlsruhe 76131, Germany
[c] Politecnico di Torino, Department of Management and Production Engineering, Corso Duca degli Abruzzi 24, Torino 10129, Italy

ARTICLE INFO

Keywords:
E-Waste
PCBA
Inspection sequencing
R-strategies
Graph Neural Networks
Reinforcement Learning

ABSTRACT

Electronic waste management faces critical challenges due to the complexity and variability of printed circuit board assemblies (PCBAs), which contain both high-value recoverable materials and hazardous components. Current inspection and recovery processes are predominantly manual and static, resulting in inefficiencies and limited scalability. This paper proposes a novel framework that integrates Graph Neural Networks (GNNs) with Reinforcement Learning (RL) to enable adaptive, real-time inspection sequencing and recovery decision-making for PCBAs. By modelling each board as a graph of interconnected components, the GNN encodes structural and defect-related information, providing a dynamic state representation for the RL agent. The agent then chooses a sequence of inspections or recovery strategies, such as reuse, repair or recycle, balancing the cost of diagnostics against the potential value of recovery. A case study on an industrial I/O device demonstrates the approach's effectiveness with simulations showing that the system learns profitable inspection and recovery policies under uncertainty while reducing unnecessary tests. A comparative analysis of state-of-the-art graph architectures reveals that Graph Attention Networks (GAT) outperform standard Graph Convolutional Networks (GCN). Results confirm the potential of GNN-RL integration to improve economic viability and sustainability in PCBA inspection for e-waste management.

1. Introduction

The global surge in electronic waste (e-waste) poses a multifaceted challenge at the intersection of environmental policy, materials science, and industrial engineering. The Global E-Waste Monitor 2024 reports that more than 60 million metric tons of e-waste are generated per year, with only approximately 17% formally documented as having been collected and recycled in a proper manner [1]. This phenomenon is driven by a combination of factors, including the rapid obsolescence of electronic products, and the increasing complexity of these systems. Printed circuit board assemblies (PCBAs) are of particular significance in this regard, given their presence in virtually all electronic devices.

Improper handling of PCBAs leads to the irreversible loss of critical raw materials such as gold, palladium, and rare earth elements [2], as well as the release of toxic substances that pose serious environmental and health risks [3]. Consequently, PCBAs contribute disproportionately to the economic and environmental burdens associated with electronic waste [4]. In response to these challenges, regulatory and economic pressures are driving industries to adopt circular economy models that prioritise reuse, remanufacturing, and recycling [5]. The efficacy of these strategies hinges on the capacity to reliably determine the appropriate R-strategy (e.g. reuse, repair, or recycling) for each board [6], a process guided by inspection results. The inspection process is particularly challenging in the case of PCBAs due to the heterogeneity of their components and failure modes, which renders them highly complex systems. In particular, a generic inspection approach is often inadequate, as certain boards may only necessitate minimal testing and cleaning, while others may conceal multiple faults that require in-depth diagnostics prior to decision-making [7,8]. At present, the inspection process is predominantly manual and labour-intensive, resulting in a bottleneck in the recovery chain [6]. Addressing this gap necessitates the development of intelligent, adaptive inspection methods capable of

* Corresponding author.
E-mail address: florian.stamer@leuphana.de (F. Stamer).

https://doi.org/10.1016/j.jmsy.2026.03.009
Received 16 September 2025; Received in revised form 4 January 2026; Accepted 15 March 2026
Available online 24 March 2026

operating under uncertainty and variability, and of supporting real-time, data-informed decision-making [7,9,10].

Traditionally, the industry has relied on rigid inspection sequences or simple decision trees, governed by preset guidelines that cannot adapt to the variability of end-of-life boards [11]. Exhaustive testing of every component is prohibitively costly and time-consuming, while insufficient testing risks overlooking opportunities to recover high-value parts. In addition, the low intrinsic value of many components (e.g., resistors, capacitors) relative to the labour required for inspection undermines the economic viability of detailed manual approaches [12,13].

These challenges highlight the need for intelligent, adaptive inspection methods capable of adjusting actions in real time. For example, if a critical component proves functional, the board may be suited for refurbishment, whereas its failure may justify immediate recycling or selective part recovery [14]. Today's remanufacturing systems largely lack the ability to make real-time decisions, essentially treating PCBA inspection and recovery as a linear, static sequence rather than a continuous, feedback-driven process.

This paper proposes a novel solution for the optimisation of PCB e-waste recovery by combining Graph Neural Networks (GNNs) and Reinforcement Learning (RL) to create an intelligent inspection controller that dynamically guides the process [15]. Within this framework, each PCBA is modelled as a graph that encodes the system-level structure of the board. This enables the decision agent to discern how the functionality of a given component may influence the usefulness of others. An RL agent, having undergone training to operate on this graph-based state representation, is responsible for determining which inspection to perform next and ultimately deciding on the most appropriate R-strategy for the board [8,16].

By leveraging a graph-based representation, the RL agent can handle a variety of PCBA designs and sizes seamlessly [17]. Its objective is to maximise the net recovered value of each board while minimising inspection costs and process time, in contrast to inflexible static plans, which cannot adjust to actual board conditions. This adaptive inspection sequencing strategy refines its decisions continuously as new information becomes available at each inspection step. This avoids low-value tests and identifies when to shift to the most appropriate reuse, repair or recycle action promptly. Furthermore, by employing data-driven learning (the integration of RL and GNNs), the decision policy improves continuously through experience across numerous cases [18], effectively managing the uncertainty inherent in PCBA failures.

2. Literature review

Research in end-of-life electronics management and remanufacturing has addressed various aspects of the problem of recovering PCBA, including disassembly planning, inspection techniques, and decision-making frameworks. This section provides a comprehensive review of the state of the art in these domains, with a focus on key contributions and their limitations in the context of developing an adaptive PCBA recovery system.

Early studies of disassembly planning typically approached the problem as an offline optimisation task, using methods such as mathematical programming, genetic algorithms and ant colony optimisation to pre-calculate near-optimal sequences for specific products [12]. For example, Gong et al. [18] formulated remanufacturing planning as a mathematical optimisation problem. While these methods can produce efficient plans in theory, they lack the ability to adapt in real time. Therefore, if unexpected conditions arise during disassembly, static plans will yield suboptimal results. To address this rigidity, knowledge-based frameworks have been proposed. He et al. [12] developed an ontology-driven model to capture expert knowledge and formalise process planning. Although this approach improves decision support, it still struggles to adapt to the heterogeneity of used electronics and the unpredictability of component failures [19]. As a result, traditional disassembly planning remains mostly static, assuming prior knowledge of product conditions, which is rarely the case in e-waste recovery.

Recent research efforts have begun to integrate AI-based reasoning and uncertainty modelling into the inspection and decision-making process [20]. However, substantial gaps persist in the pursuit of full adaptivity. Ahsan et al. [21], introduced computational intelligence techniques to product testing, using machine learning (ML) to predict reliability outcomes. Although this approach improves test efficiency by identifying potential failures at an early stage, it still relies to a predetermined sequence of tests and does not adapt to intermediate results. On the other hand, Bao et al. [22] proposed an integrated decision framework using conditional evidence theory to deal with uncertainty in remanufacturing. By converting component inspection data (e.g., wear measurements) into evidence and integrating it with prior knowledge, they improved decision accuracy. While the system is demonstrably effective, it produces a single static decision post-inspection, without supporting iterative reassessment throughout a multi-stage inspection process. As a result, it lacks the adaptability needed for dynamic, real-time planning. To address these limitations, recent research has introduced more flexible and adaptive strategies that integrate advanced diagnostics with probabilistic decision logic [23–25]. These approaches are designed to facilitate sequential and real-time decision-making processes, wherein the outcomes of each inspection serve as the basis for determining subsequent actions. Despite the advances represented by these methods, challenges persist in the full operationalisation of dynamic re-evaluation across inspection stages, particularly in complex or uncertain environments.

In parallel with advances in process planning, there have been significant advances in PCBA inspection technologies. Today's industrial systems can use multiple sensors and imaging modalities to detect defects and assess component health without disassembling the entire board. Common techniques include X-ray radiography for hidden solder joint defects, infrared thermography for identifying overheated components, and optical cameras for detecting surface defects [14]. These techniques are often combined to improve diagnostic accuracy, as shown by Li et al. [26] who combined optical and infrared imaging to improve defect detection. Similarly, previous research by Wright [27] investigated a multi-resolution sensor fusion approach to fault isolation on PCBAs, integrating signals at different scales to identify component failures [26,27]. Despite their technical sophistication, multi-sensor inspection systems typically operate in a static mode, following a predefined inspection sequence regardless of the initial findings. Accordingly, while the combination of multi-sensor data theoretically increases defect coverage, the lack of adaptive control prevents these systems from realising their full potential in reducing the time and cost of inspection processes [21].

In order to support truly adaptive sequences of operations, adaptive decision frameworks such as Markov Decision Processes (MDPs) and RL have been explored to support dynamic inspection sequencing in general [28]. MDPs model sequential decision problems where states represent current knowledge about a product and actions represent possible inspection steps [29]. An optimal policy aims to maximise rewards, such as recovered value, while minimising inspection costs [30–32]. Recent applications of RL in manufacturing demonstrate the technique's ability to learn adaptive strategies from data [33]. For example, Stricker et al. [29] used RL to optimise adaptive job scheduling to respond in real time to line variability. Another example is given by Stamer et al. [32] who applied RL to dynamically generate a pricing based on the production system usage and requested delivery time. Applying RL to PCBA remanufacturing would involve training an agent to determine which inspection to perform next based on current findings, rather than adhering to a static plan. However, challenges remain especially in defining an appropriate state representation that captures the complex structure and strongly varying composition of PCBAs [34]. To date, the extant literature has not provided a dedicated framework that integrates an RL or MDP-based decision engine with real-time inspection data specific to

PCBAs.

The concept of graph-based machine learning as proposed in this work can assist in this regard: by representing the PCBA as a graph of components, one can use GNNs to help the decision agent generalise its policy across different board designs and conditions – an approach that, to the best of our knowledge, has not been explored in prior works on electronics recycling or remanufacturing [14,35]. In this context, different GNN architecture are available. Foundational models like Graph Convolutional Networks (GCN) [36] effectively aggregate neighbourhood information using fixed weights, while treating all connections as equally significant. Conversely, architectures such as Graph Attention Networks (GAT) [37] employ self-attention mechanisms to dynamically weigh the importance of adjacent nodes. This ability to discern which connections are most relevant is particularly interesting for PCBAs, where the functional impact of a component failure depends heavily on its specific interdependencies This capability has recently been demonstrated to be effective in complex manufacturing process selection tasks [37] making it particularly promising for PCBAs where component interdependencies vary significantly.

A final point to consider in the existing research is the selection of R-strategies in parallel with the inspection process. Most existing studies consider decisions as a result of a fully completed inspection phase, rather than dynamically integrating them with the inspection itself [30–32]. For example, Bao et al. [22] consider recovery paths after data collection, without optimising the sequence of inspections to minimise cost while maximising component recovery. However, an integration of inspection and recovery decisions is necessary to balance the cost of additional testing with the potential salvage value. Current static strategies fail to adapt to specific cases, potentially leading to wasted inspection effort on defective, low-value boards or insufficient inspection of functioning, potentially valuable boards

In summary, recent approaches to PCBA inspection and recovery typically rely on static inspection plans, rule-based decision trees, or knowledge-driven systems. While effective in constrained scenarios, these methods inherently lack the ability to adapt dynamically to variable board conditions, integrate complex product structures, or sequentially update decisions based on partial diagnostic information. While Markov Decision Processes (MDPs) provide a formal structure for sequential decision-making, solving them via classic stochastic optimization in large state spaces is often rendered unfeasible. Reinforcement Learning (RL) has emerged as a promising alternative. However, traditional RL methods assume fixed-length inputs, making them unsuitable for variable PCBA layouts.

To bridge these gaps, this research presents a framework that integrates RL with Graph Neural Networks (GNNs). By encoding the board's relational structure, this approach enables a holistic, real-time decision engine that links inspection actions directly to ultimate remanufacturing outcomes. Specifically, by evaluating both GCN and GAT architectures, this work aims to identify the optimal state encoding for guiding PCBAs through inspection and recovery trajectories that are adaptive, structurally aware, and efficient under uncertainty.

3. Problem statement and methodology

Building upon the challenges discussed in the previous section, this part of the paper focuses on the formalisation of the decision problem and the description of the proposed solution framework. Whilst the prevailing PCBA inspection and recovery strategies are frequently predicated on fixed or heuristic rules, such approaches are inadequate in their inability to adapt to varying board configurations, defect patterns and economic constraints [38]. In this context, a more principled and adaptive strategy is required – one capable of guiding sequential inspection decisions under uncertainty, while optimising trade-offs between diagnostic cost, information gain, and potential recovery value [39].

The following subsections first present a formal problem definition, framing the decision-making challenge as a sequential, information-driven process. Secondly, a novel methodology integrating RL and GNNs is proposed. As will be discussed further, this integration enables the system to update its inspection policy dynamically based on component-level observations and the evolving state of the board. This algorithmic design is particularly well suited to the problem at hand for three reasons: it facilitates decision-making in uncertain situations; it captures the structural relationships between PCBA components; and it adapts the level of inspection in real time based on trade-offs between diagnostic cost and potential recovery value. These properties closely align with the core challenges of PCBA inspection, including heterogeneous layouts, variable defect patterns, and economic constraints. The methodological framework provides a robust foundation for identifying optimal inspection sequences and selecting the most appropriate recovery strategy for each board.

3.1. Problem statement

The central research problem addressed in this work is how to dynamically determine the most appropriate next step to support effective decision making across the full range of R-strategies. This decision-making process must take into account the current state of available information, the probabilistic outcomes of possible inspections, and the costs and benefits associated with alternative recovery actions. To address this challenge, the proposed methodology implicitly evaluates a number of potential sequences of inspection actions, estimating the likely information gain and economic impact of each one. Each decision is thus informed by both the history of past inspections, as well as the expected value of future inspections. This sequential approach allows the system to adapt dynamically as new data becomes available, balancing inspection costs against potential revenue from component recovery.

For example, consider a PCBA containing several critical components such as microcontrollers, capacitors and sensors. An initial inspection can target the component most likely to be faulty. If the result shows it to be functional, the system might recommend the inspection of a second component whose failure would significantly affect the recoverability of the board. Conversely, if the first inspection confirms a defect, the system may decide to stop further inspections and recommend a recycling strategy. In this way, the sequence of inspection steps supports an efficient and cost-effective selection among the available R-strategies. Furthermore, since electronic components can exhibit multiple failure modes with different probabilities, the ability to interpret inspection results probabilistically and adjust the course of action accordingly has a critical impact on both efficiency and profitability [29]. The optimisation framework therefore continuously assesses whether the expected value of the information gained from further inspections justifies the additional cost. Achieving this balance is essential for the development of sustainable, economically viable e-waste management strategies that balance environmental objectives with industrial practice [40].

3.2. Proposed methodology

The challenges outlined in the problem statement – particularly the need for dynamic, information-driven inspection under uncertainty – motivate the adoption of a learning-based approach. Consequently, the decision-making process is formulated as an MDP, which offers a rigorous framework for sequential decisions with probabilistic outcomes [41]. This formulation facilitates the development of an adaptive agent that perpetually refines its strategy as novel inspection data becomes available. The proposed data-driven architecture (Fig. 1) integrates RL with GNNs to address the structural and informational complexity of the problem. Within this framework, each state encodes the current knowledge about the PCB's components, including their observed condition and estimated failure probabilities [42]. Actions, in turn,

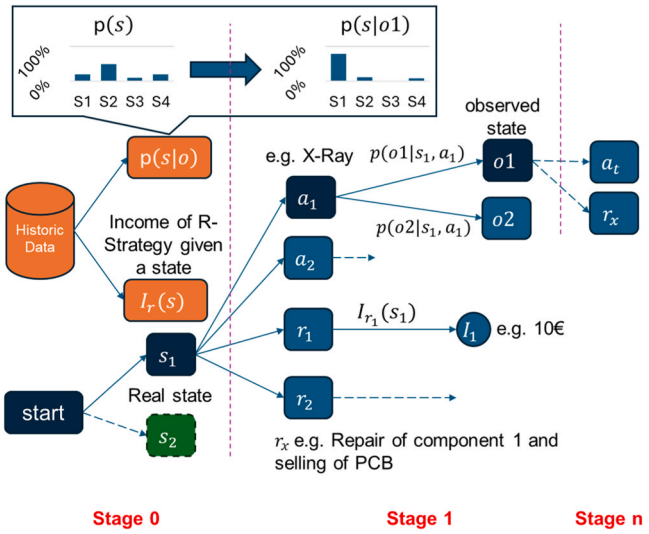

Fig. 1. Overview of the proposed data-driven decision framework for sequential PCBA inspection and recovery.

correspond to specific inspection steps or the selection of an R-strategy. State transitions reflect the stochastic outcomes of these actions, thereby capturing both the uncertainty in defect detection and the potential effectiveness of recovery strategies. The architecture is well-suited to the dynamic nature of the task, as it updates decisions in real time in response to evolving inspection outcomes [41].

As represented in Fig. 1, the sequential decision framework for PCBA inspection and recovery is comprised of a series of interconnected steps. The decision-making process starts at Stage 0 with a starting point (state s_1), which is based on past data about the PCB's condition. This initial belief shown as $p(s|o)$ is a way of working out the chances of defects based on past failures. At the same time, the expected economic value of using each R-strategy is estimated using the income function $I_r(s)$, based on the current belief state.

In the first stage, the RL agent looks at the possible actions. These can be specific inspections (a_1, a_2, for example, X-ray inspection) or choosing the best R-strategy to use at the end (r_1, r_2, for example, repair or recycle). If an inspection action is selected, such as a_1, carrying it out leads to an observation (o_1 or o_2) that gives a new probabilistic information ($p(o_1|s_1,a_1)$ about the true state of the board. Then, the agent updates its belief state, improving the probability distribution $p(s|o_1)$, because there is less uncertainty after the observation. If an R-strategy is chosen directly (e.g. r_1), the process ends immediately and the associated income $I_{r_1}(s_1)$ (e.g. 10€) is realised without further inspections. The selection of an R-strategy always marks the end of the process.

Accordingly, a key aspect of designing an RL agent is to define the action space, the state space, and the reward mechanism [32]. In this study, the action space is straightforward, consisting of a fixed set of possible inspection processes and a predefined collection of R-strategies (reuse, repair, recycle). In contrast, the state space is much more complex. To accurately represent the state space, each component of a PCBA – including transistors, resistors, capacitors and other elements – must be considered along with its respective state. The combination of all components and their interconnections fully characterise a PCBA. Effective modelling of the PCBA involves its representation as a graph, where each component corresponds to a node and the connections between components are represented as edges. Furthermore, the state of each component is defined by a set of defects selected from a predefined list of possible defect types, as shown in Fig. 2.

From a mathematical point of view, the resulting graph can be expressed as:

$$G = (C, E) \tag{1}$$

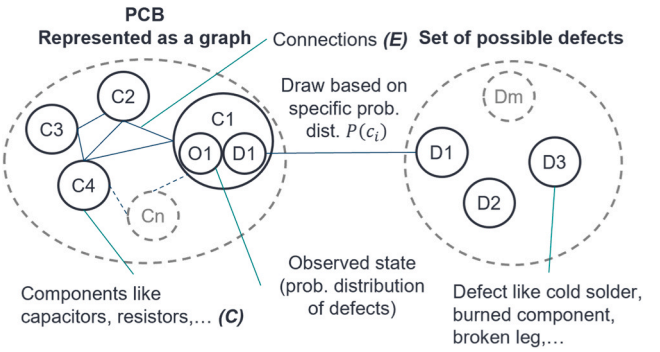

Fig. 2. Mathematical representation of a PCBA.

where C represents the PCBA as nodes (such as capacitors, resistors, and other electronic elements), and E represents the physical/electrical connections between these components. The connections (E) between components are not just passive links; they are essential because each interconnect can vary in nature (e.g. solder joints, wire connections, bus interfaces) and significantly influence the final structure and behaviour of the PCBA. Accurate tracking of these interconnects is critical because variations in their properties can lead to different failure modes or propagation of defects throughout the board. In this model, each node c_i in C is associated with a probability distribution:

$$P(c_i) = \{p_{i1}, p_{i2}, \ldots, p_{in}\} \tag{2}$$

where p_{ij} is the probability that component c_i exhibits defect d_j. In other terms, this probability describes the probability distribution of potential defects $D = \{d_1, d_2, \ldots, d_n\}$. Additionally, each component has an observed state o, which represents the current belief (probability distribution) of the real state (defect) of a component.

It's important to note that PCBA can vary greatly in terms of component count and defects. Traditional RL agents often use neural networks that require input vectors of fixed length, which is a challenge when dealing with the variable sizes found in PCBA [41,43]. GNNs provide a solution by efficiently processing graph-structured data to account for the variability in PCBA structures [17]. GNNs work by capturing the relationships between nodes and aggregating information from a node's neighbours to generate node embeddings. This results in a comprehensive graph representation that allows GNNs to dynamically adapt to varying numbers of components and defects, which is essential for optimising PCBA inspection and repair processes [35].

In this context, the state S of the model is defined by an instance of the graph G which includes all components of the board in both their real state and their observed state. The GNN processes the graph G – representing the state S – to produce node embeddings h_v:

$$h_v = GNN(S) \tag{3}$$

To evaluate the impact of the state encoding mechanism on decision quality, this work implements two distinct architectures for the $GNN(\cdot)$ function:

1. GCN: This approach aggregates information from a node's neighbours using fixed spectral rules. It assumes that the structural importance of a connection is determined solely by the degree of the connected nodes.
2. GAT: This architecture employs a self-attention mechanism to compute learnable coefficients (attention weights) for every edge. This allows the agent to dynamically value the information from specific neighbours, prioritising the state of a critical processor over a passive resistor, for example, regardless of the node degree.

The resulting embeddings h_v provide a compact and informative

representation for each component. By using h_v the policy π determines the probability of selecting each action based on the optimized policy learned by the RL agent:

$$a_t = \pi(h_v) \tag{4}$$

where a_t is the action chosen at time t. The actions that can be performed are either measurements such as X-ray, flying probe, etc., or the choice of an R-strategy for the board. More specifically, π is a function that maps the states of a system S to the probabilities of choosing each possible action a in that state. In practice, π determines which action is most likely to produce the best outcome given the current state of the system. This decision process is based on prior learning and optimising the expected reward.

After executing a_t, the system observes new data and transforms it into the next state S'. To quantify the quality of this action in a given state, the Q-function, a core component of RL updates, is use [41,44]. This Q-function is defined as:

$$Q(h_v, a_t) \leftarrow \quad Q(h_v, a_t) + \alpha[r_t + \gamma \max_a Q(h'_v, a_t) - Q(h_v, a_t)] \tag{5}$$

where $Q(h_v, a_t)$ is the $Q-value$ for taking action a_t when in state represented by embedding h_v, α is the learning rate, which determines the extent to which the newly acquired information will override the old information, r_t is the reward received after action a_t, γ is the discount factor which quantifies how much importance is given to future rewards and $\max_a Q(h'_v, a_t)$ is the maximum predicted reward achievable from the next state h'_v, for any possible action a_t.

The architecture provides a principled framework for real-time decision-making under uncertainty in PCBA inspection and recovery. Integrating GNN-based state encoding with RL allows the system to adapt to structural variability and defect distributions across different types of board. The Q-learning mechanism enables the continuous refinement of inspection strategies based on observed outcomes, allowing the agent to prioritise actions that maximise economic return. Performance is influenced by key components including the reward design, graph-based state encoding and exploration strategy. The reward function balances short-term inspection costs against potential recovery value, applying scaled penalties for unnecessary actions and bonuses for the early identification of valuable modules. The GNN-based representation captures component attributes and interdependencies, enabling the model to generalise across different board designs and focus on structurally critical areas. These components guide the agent towards efficient, high-value inspection-recovery trajectories. However, for such an intelligent controller to be viable in a high-throughput remanufacturing line, the decision-making process must not only be economically optimal but also computationally tractable.

3.3. Computational complexity analysis

To verify the system's suitability for real-time deployment, we analyse the asymptotic complexity of the underlying network operations. Let $n = |C|$ denote the number of components and $m = |E|$ denote the number of interconnections of a PCBA. Let K be the number of available actions, which includes inspection technologies plus R-strategies. The Q-network used in this work consists of an L -layer GNN with embedding dimension d, followed by a global pooling layer and a multilayer perceptron (MLP) that maps the aggregated graph representation to K Q-values (Fig. 4).

For a single decision step, the dominant cost arises from message passing and node updates in the GNN. Each layer requires $O(md)$ operations, and $O(nd^2)$ operations to update node embeddings. Accordingly, the time complexity for one forward pass of the GNN is:

$$T_{GNN}(n, m) = O(L(md + nd^2)) \tag{6}$$

In the case of the GAT architecture, the complexity remains asymptotically linear with respect to the number of nodes and edges. While this introduces a constant computational overhead compared to GCN, it does not alter the fundamental scalability of the approach.

The subsequent MLP layers operate on a fixed-size pooled embedding and contribute $O(d^2)$ (assuming hidden layers of width d), which is negligible compared to the GNN component. Thus, the overall per-step complexity of the policy evaluation is $O(L\left(m\,d + nd^2\right))$, and the memory requirement for activations is $O(L(nd + md))$.

During training, each episode comprises at most H decision steps. At each step, a forward and a backward pass through the network are executed, with backpropagation having the same asymptotic complexity as the forward computation. The time complexity per episode is therefore:

$$T_{train,episode}(n, m, H) = O(H\,L(md + nd^2)) \tag{7}$$

For N episodes, the total training effort is $O(N\,H\,L(md + nd^2))$. In the present case study, n and m are bounded by the small subsystems (3–10 components) considered for the I/O device, such that training time scales primarily with N and H, not with the PCBA size.

In deployment, only forward passes are required. The complexity of the online decision-making process is thus $O\left(L\left(md + nd^2\right)\right)$ per decision step. Given the limited number of components per board, this complexity is effectively linear in the number of nodes and edges, indicating that the proposed approach is computationally tractable for real-time application in various PCBA remanufacturing environments.

3.4. Exemplary runtime schedule and execution-time analysis

To complement the analysis of computational complexity, we generated an exemplary runtime schedule by instrumenting the full GCN–RL implementation. The profiling measured the time consumed by each major stage of the algorithm: Inference (action selection), Training (backpropagation and optimisation), and Belief Update (state refinement). Measurements were collected over 1000 episodes to capture meaningful variability in decision-sequence length, action frequency and state-transition patterns occurring during learning and execution. In this analysis, one episode corresponds to a complete decision sequence for a single PCBA, starting from the initial uncertain state and ending with the final strategic choice (reuse, repair, or recycle). Table 1 summarises the recorded mean execution times, total accumulated times and number of calls for each stage.

The results show a clear separation of computational effort across the stages. Training dominates the runtime with an average of 46.15 ms per optimisation step. This behaviour is expected, as deep RL involves repeated forward and backward passes through the network. Importantly, this cost is incurred only during the offline learning phase.

For deployment, the relevant quantity is the inference time required to select the next inspection or recovery action. The profiling shows that inference is highly efficient, requiring on average only 0.17 ms per decision. Combined with the negligible computation required for the belief update (< 0.01 ms), the overall per-decision runtime during operation remains well below 1 ms. This ensures that the controller can be integrated into high-throughput PCBA inspection processes without violating cycle-time constraints.

Overall, the runtime schedule confirms that the computationally intensive training phase can be executed offline or in parallel, while the

Table 1
Overview of runtime schedule.

Stage	Mean Time (ms)	Total Time (s)	Count
Inference	0.17	0.35	2010
Training	46.15	86.90	1883
Belief Update	< 0.01	0.02	6462

online decision-making phase operates at sub-millisecond latency. This demonstrates that the proposed GNN-RL framework is suitable for real-time use in industrial inspection and remanufacturing environments.

4. Case study

The feasibility and effectiveness of the approach were validated using an industrial use case involving a commonly used input/output (I/O) device, which had previously been studied by Stamer et al. [14]. Its ultrasonically welded plastic case design makes it suitable for R-strategies of typical PCB-based industrial products [45]. Fig. 3 shows an example of the I/O device without the plastic case. For these products, PCBA subsystems containing three to ten critical components were analysed, focusing on the most common defects identified within the use case: burnt components caused by e.g. short circuits, cold solder joints, and broken parts such as bent or detached legs of integrated circuits (IC) [46]. While qualitative data about the product (e.g. via interviews) are available for publication, quantitative data like resell price or defect probability distribution are not. Therefore, in order to facilitate a simulation and economic evaluation, it was hypothesised that a functional PCBA could be resold for 100 units. A simulation system was developed to generate 100 PCBA of the IO Link per episode based on defined statistics, each with different components and defects distributions. The prior belief regarding defect type, such as burnt components, cold solder joints, and broken parts, was initialised at 50% probability. By collecting inspection data over an extended period, these initial probabilities can be progressively updated to approximate the true distribution of defects. Although simplified in this study, the assumptions made allow for consistent benchmarking and controlled variability in agent training.

The inspection system uses three primary measurement technologies to detect defects and guide decision making: X-ray, visual inspection and flying probe. All of them are widely used. These technologies differ in terms of defect detection probabilities, inspection times and associated costs. X-ray (assumed cost of 10 units) and visual inspection (assumed cost of 1 unit) offer fixed costs and shorter inspection times, making them suitable for quick initial assessments [47]. In contrast, flying probe offers high accuracy for all defect types, but has higher costs that scale with the number of components on the board, making it more expensive for larger boards (assumed cost of 3 units per component) [48].

In operation, each inspection technology is modelled as a discrete, controllable resource with defined processing capacity. The latter is defined as the number of inspections that can be performed by the resource simultaneously. When multiple units of the same technology are available (e.g. multiple X-ray machines or visual inspection stations), they enable true parallel processing by distributing inspection tasks concurrently across these units. The scheduling system dynamically assigns boards to available inspection units based on current queue lengths and expected diagnostic value. After each inspection, the defect probabilities for the corresponding PCBA are updated using the technology-specific detection results, which refine the system's belief about the presence of specific defects. This Bayesian-like updating mechanism shifts the probability distribution to reflect the latest evidence. The combination of complementary measurement technologies, i.e. from low-cost, low-accuracy methods to high-fidelity, high-cost diagnostics, allows the system to strategically select inspections that balance diagnostic accuracy and operational costs. This flexible architecture enables the system to adjust inspection strategies in real time, optimising both economic performance and diagnostic accuracy throughout the process.

The RL algorithm has been implemented based on a GNN architecture. Fig. 4 shows the processing of node embeddings from a PCBA graph through a neural network, with the purpose of informing action selection (e.g. X-ray inspection or component reuse). This architecture serves as the common framework for both the GCN and GAT implementations evaluated in this study. The structured input on the left represents the graph-based state representation of a PCBA, where each node (for example, an IC, resistor, etc.) is encoded into an embedding vector reflecting its attributes and connections. The node embeddings, when considered collectively, function as the input to a deep neural network comprising multiple layers (illustrated as layers 1 through 4). This network employs a process of progressive integration and abstraction, whereby information is extracted from the entirety of the graph. The final output layer generates scores (Q-values) corresponding to the possible actions that an agent can take. These include conducting an X-ray diagnostic or opting to salvage and reuse a particular component. In this manner, the model translates the rich state information from the PCBA's graph-structured data into actionable decisions, selecting the operation that maximises the expected benefit based on the learned Q-values.

A key challenge encountered during initial training was the agent's overall tendency to reduce the number of inspection steps before deciding on a R-strategy in the course of the training. This behaviour stemmed from the negative immediate rewards associated with inspection costs and the conditional profitability of reuse and repair strategies. To address this, a dynamic reward mechanism was introduced using a two-segment linear function. This mechanism initially provided higher rewards for both inspection and R-strategy actions, gradually decreasing the rewards over time to align with actual costs and benefits.

For GCN as well as for GAT the training setup involved 100 episodes per training run, with each episode comprising the simulation of 100 individual PCBAs. To ensure the agents reached their maximum potential, an automated hyperparameter tuning process was conducted over 40 distinct optimization runs, automatically adjusting critical variables such as network dimensions (e.g., number of neurons) and the dynamic reward signal scaling. This process was critical to identify the optimal configuration for each architecture, thereby facilitating successful learning and adaptation. By applying this standardized protocol, any observed performance differences can be attributed solely to the superior state encoding capability of the respective GNN architectures rather than training disparities. A comprehensive summary of all simulation, agent, and domain-specific parameters is provided in Appendix A and results are discussed in Section 5.1. To ensure the proposed solution is not only performant but also robust for industrial application, a systematic parameter sensitivity analysis was integrated into the validation

Fig. 3. IO Link Master from Balluff.

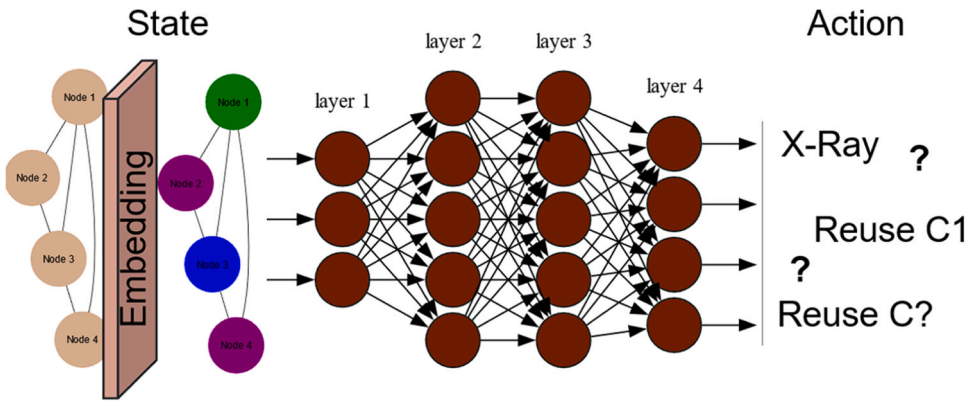

Fig. 4. Q-network architecture with interchangeable GNN-based state encoder (GCN/GAT) and action selection.

methodology (Section 5.2). This analysis was conducted using the baseline GCN configuration to isolate the fundamental dependencies between economic parameters (e.g., inspection costs) and the agent's decision logic. Finally, a performance comparison between GCN and GAT was conducted (Section 5.3).

5. Results and discussion

This section presents the results of the proposed GNN–RL framework and discusses its performance from three complementary perspectives. First, the learning behaviour and decision-making characteristics of the agent are analysed under the baseline parameter configuration derived from the industrial case study. This includes an evaluation of reward convergence, inspection sequence length, strategy selection, and achieved profit.

Second, a dedicated sensitivity analysis is conducted to assess the robustness of the learned policy with respect to key economic and probabilistic parameters. By systematically varying inspection costs, recovery-related parameters, and defect probabilities, this analysis evaluates how sensitive the system's performance is to modelling assumptions and input uncertainties, thereby improving reproducibility and practical interpretability of the results.

Third, a comparative analysis is performed to evaluate the impact of the underlying neural network architecture on overall system efficiency. By benchmarking the baseline Graph Convolutional Network (GCN) against a Graph Attention Network (GAT), this section quantifies the operational gains achieved through advanced state encoding. Focusing on cumulative profit over a validation sequence, the comparison assesses how different graph aggregation mechanisms affect the agent's ability to distinguish high-value recovery opportunities, ultimately determining the framework's potential to transition from a theoretical proof-of-concept to an economically viable industrial solution.

5.1. Learning behaviour and baseline performance

This subsection analyses the learning dynamics and performance of the baseline GCN agent in the use case defined in Section 4. Each training run corresponds to a distinct hyperparameter configuration, and the resulting learning curves are compared to assess stability and convergence behaviour. The main goal is to find an agent configuration which can handle the task well. The graphs shown use distinct coloured curves to represent individual training runs from the optimization (cf. Section 4), allowing visual comparison of different training runs. As the achieved reward is a basic metric commonly used for initial performance evaluation [41], Fig. 5 illustrates the progression of reward accumulated over time.

The figure shows in around 50% a downward trend in rewards, a predictable outcome of the dynamic rewarding (cf. Section 4) as the scaling adapts over time. Results show that approximately 50% of configurations end up with a net positive reward. In other words,

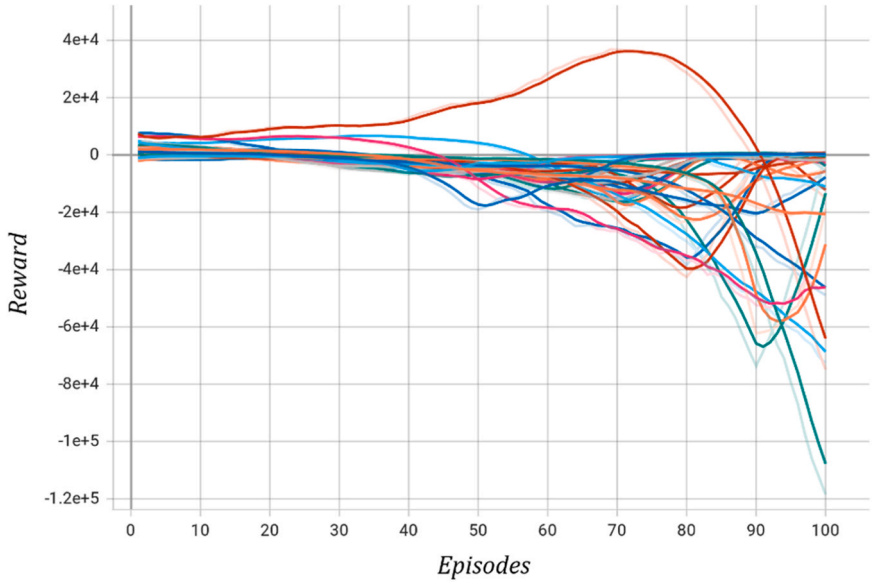

Fig. 5. Achieved reward over time (moving average of 10 episodes).

considering the dynamic reward increasing negative values for applied tests a constant or even slightly positive course is reflecting a successful learning.

Extending the strategic decision-making, Fig. 6 illustrates the reward progression for repair actions throughout the training process. Initially, repair rewards are high, driven by the dynamic reward mechanism, which incentivises the agent to prioritise repair actions regardless of their immediate profitability. Over time, the rewards associated with repair actions gradually decrease, reflecting the agent's growing awareness of their long-term value. By the end of the training period, some configurations lead to a consistent selection of repair strategies that maximise net positive rewards, demonstrating the general ability to identify and implement profitable repair actions based on insights gained from inspection data. This evolution from broad initial strategies to refined, targeted tactics underscores the adaptive nature of the learning process and the agent's increasing ability to handle complex decision scenarios.

Further analysis reveals the strategic dynamics of sequence optimisation and its relationship to economic outcomes. The overall profitability of a sequence - consisting of the combination of measurements and the final R-strategy - correlates with the length of the sequence. Longer sequences require more measurements before selecting an R-strategy, thereby reducing uncertainty. However, excessive measurements lead to diminishing returns due to escalating inspection costs (see Fig. 7).

Fig. 8 shows that some agents consistently increase their sequence lengths, leading to unprofitable outcomes due to high inspection costs.

In contrast, other agent configurations learn to balance their sequence lengths over the last 20 episodes, achieving greater profitability by minimising unnecessary measurements. In addition, a subset of agents converges to a sequence length of approximately one, opting primarily for recycling. While recycling is a relatively low-risk option, it tends to be less profitable than performing targeted measurements followed by repairs. In total, the fan-shaped curves suggest the optimal solution to be unstable and challenging to find.

Finally, the profit achieved is evaluated as the primary performance indicator. The profit for a PCBA is calculated by subtracting the cost of previous measurements and repair costs from the revenue generated by a successful R-strategy. Fig. 9 shows that approximately 60% of the agent configurations achieve significant profits by the end of the training process. This result demonstrates that, within the cost and revenue assumptions of the use case, the system effectively learns to optimise inspections and make profitable decisions. These results highlight the model's ability to balance measurement costs with recovery revenues, ensuring that the chosen R-strategies are viable.

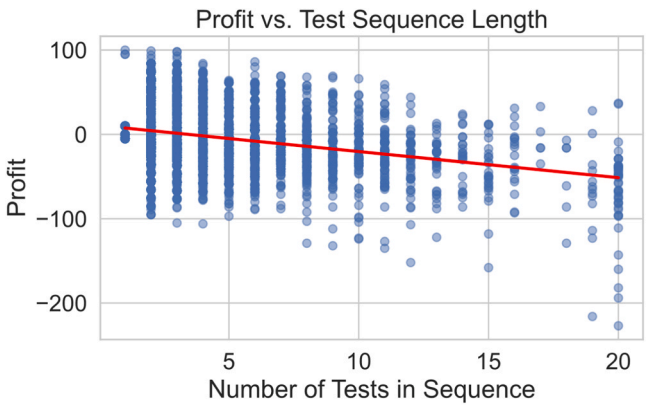

Fig. 7. Comparison of test sequence length and achieved profit.

After this examination, the most effective working model of the pool of optimisation runs was selected and implemented across 2000 boards. As shown in Fig. 10, the application of this agent has been shown to yield a profitable outcome in the specified use case. Overall, a profit of 2435 units was achieved. The variation of the profit can be explained by the random collection of PCBAs in different states. Lower profit might indicate a more “difficult” PCBA where less profit is achievable even in the best possible case.

These results confirm that the proposed RL-based inspection system can effectively learn recovery policies that balance cost and value across a wide range of scenarios. Despite the stochasticity of the board conditions and defect patterns, the agent displays a high degree of robustness in identifying profitable strategies. The variability observed across training runs highlights the importance of proper reward design and tuning, which remain key factors for successful deployment in real-world applications.

5.2. *Parameter sensitivity analysis*

To assess robustness and improve reproducibility, a parameter sensitivity analysis was conducted using the trained baseline (GCN) policy. While the absolute profit values are different for the GAT agent,

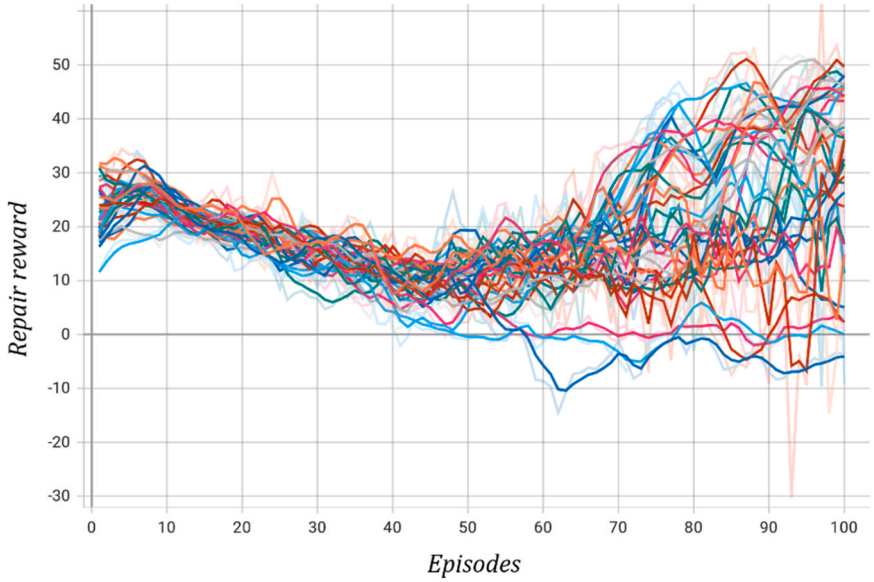

Fig. 6. Achieved reward over time for action “repair”.

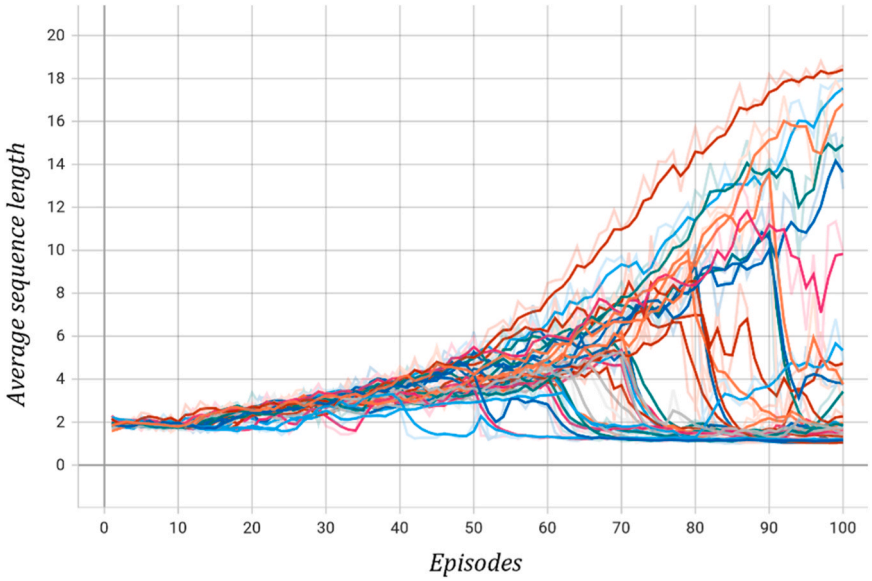

Fig. 8. Sequence length over episodes.

Fig. 9. Profit per episode.

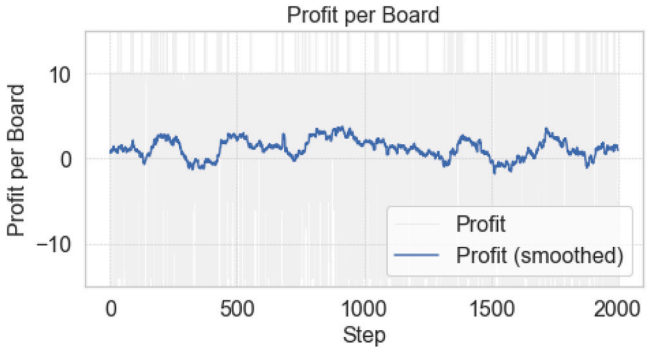

Fig. 10. Profit per episode for the selected agent.

the relative trends and economic trade-offs identified in this analysis are driven by the environment's structure and are therefore representative of the general system dynamics. Key economic and probabilistic parameters were varied individually one by one to representative low and high values around the baseline configuration, as summarised in Table. All other parameters were kept constant to isolate the influence of each factor. Each scenario was evaluated over a long simulation horizon to obtain stable average profit estimates. Table 2

Fig. 11 summarises the impact of low, baseline, and high parameter

Table 2
overview of parameter variations for sensitivity analysis.

Parameter	Low	Baseline	High
Defect probability per component	60%	75%	90%
X-ray inspection cost	5	10	20
Visual inspection cost	0.5	1	2
Flying-probe cost (per component)	1.5	3	6
Reuse income	80	100	120
Repair cost (fixed)	2.5	5	10
Recycle income	5	10	15

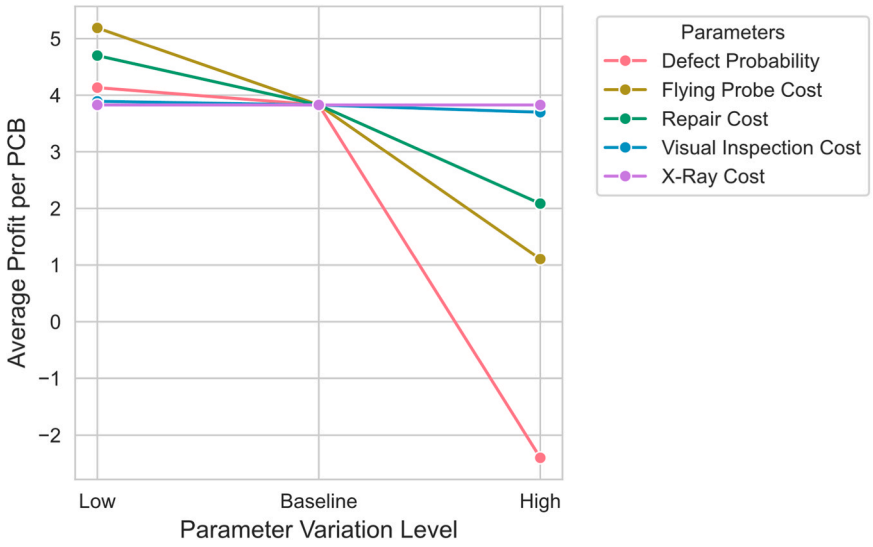

Fig. 11. Sensitivity Analysis: Impact of Parameters on Total Profit.

settings on the average profit per PCBA. The results indicate that the learned policy is most sensitive to defect probability. Increasing the overall defect rate leads to a substantial reduction in profit and, in extreme cases, to negative values. This behaviour reflects earlier termination decisions and a shift towards conservative recovery strategies, such as recycling, when the expected benefit of further inspections diminishes.

Inspection-related costs show differentiated effects. In particular, higher costs for the flying-probe inspection significantly reduce profitability, as this technology is typically selected at later stages of the inspection sequence and incurs costs proportional to board complexity. In contrast, variations in visual and X-ray inspection costs have a comparatively moderate impact on overall profit. The learned policy relies primarily on low-cost inspections for early information gain and uses higher-cost inspections selectively and iteratively.

Recovery-related parameters exhibit a similarly intuitive influence. Increasing repair costs leads to a noticeable decline in profit, reflecting the reduced attractiveness of repair-oriented strategies. Overall, the sensitivity analysis confirms that the learned inspection and recovery policy behaves consistently with economic intuition and remains stable across a broad range of parameter variations. At the same time, it is apparent that the policy needs to be retrained if parameters change significantly. The results demonstrate that the proposed framework is not overly tuned to a specific parameter set and can be transferred to different industrial settings with varying cost structures and defect characteristics, thereby supporting reproducibility and practical applicability.

5.3. *Comparative analysis: GCN vs. GAT*

To quantify the benefit of advanced state encoding, we compared the cumulative profit of the Baseline GCN agent against GAT agent over a validation sequence of 750 boards. As illustrated in Fig. 12, the GAT architecture significantly outperforms the GCN baseline, demonstrating

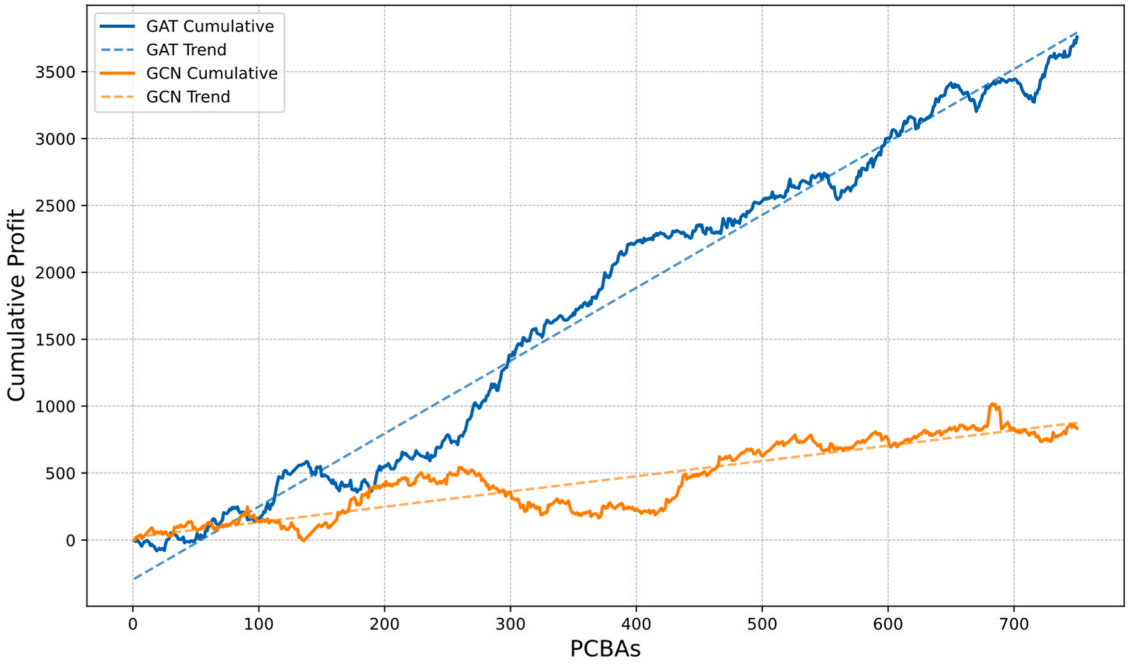

Fig. 12. Comparative performance analysis: Cumulative profit of Baseline GCN vs. GAT agents.

a superior ability to identify profitable recovery trajectories.

A linear regression analysis of the cumulative profit curves reveals a contrast in operational efficiency. The GAT agent achieves a robust growth rate (slope) of approximately 5.45 units of profit per PCBA. In comparison, the GCN baseline yields a modest growth rate of 1.14 units per PCBA. This corresponds to a nearly 5-fold increase in profitability per processed board. While the GCN agent maintains a positive margin, its flatter trajectory indicates a struggle to consistently distinguish high-value recovery opportunities from low-margin or break-even cases. Consequently, these findings suggest that the framework is not merely a theoretical proof-of-concept; the system possesses the necessary scalability and economic potential.

6. Conclusions and future works

This paper presents a methodology for optimising PCBA inspection sequences and selecting appropriate recovery strategies - reuse, repair and recycle - aimed at increasing value recovery in the context of e-waste management. The approach integrates real-time inspection data with reinforcement learning, relying on a GNN architectures to dynamically adjust decisions based on diagnostic feedback. The model weighs the cost of inspections against the expected revenue from recovery activities and improves overall inspection efficiency by learning to prioritise high-value measurement steps. This is particularly relevant in manufacturing environments where resources are limited, and inspection actions must be applied selectively.

Simulation results demonstrated that the system could learn inspection and recovery policies that adapt to board configurations and failure probabilities, even under imperfect information and evolving internal states. Crucially, the comparative analysis reveals that the choice of graph architecture significantly impacts economic performance. The GAT agent outperformed the GCN baseline. The increase in efficiency confirms that the ability to dynamically weigh component interdependencies via attention mechanisms is essential for identifying high-value recovery trajectories in complex electronics. Furthermore, this substantial performance gain illustrates the framework's optimization potential, proving that the approach can transcend initial feasibility demonstrations to achieve the high economic viability required for industrial implementation.

Future work will address current limitations by increasing the number and variety of inspection technologies as well as PCBAs. Especially, methods such as thermal analysis and functional testing can be added. Additional recovery strategies will also be introduced to reflect industrial practices, such as component cannibalisation, second-life use or resale through aftermarket channels. Additionally, while the demonstrated policies are profitable, the solution space can be explored even more. Future research should facilitate more extensive training regimes, as prolonged learning phases are likely to uncover even more refined and efficient inspection strategies. In addition, it is necessary to collect a real defect database to parametrize the initial belief probabilities of defects of a PCBA correctly. This might improve the performance as this context knowledge helps the agent find overall dominant strategies.

CRediT authorship contribution statement

Maurizio Galetto: Writing – review & editing, Visualization, Supervision, Resources, Methodology, Funding acquisition, Formal analysis, Conceptualization. **Florian Stamer:** Writing – review & editing, Writing – original draft, Visualization, Validation, Software, Methodology, Investigation, Formal analysis, Data curation, Conceptualization. **Gisela Lanza:** Writing – review & editing, Supervision, Resources, Funding acquisition, Formal analysis, Conceptualization. **Stefano Puttero:** Writing – review & editing, Writing – original draft, Visualization, Validation, Software, Methodology, Investigation, Formal analysis, Data curation, 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.

Acknowledgement

None.

Appendix A. comprehensive parameter settings

To ensure reproducibility of the experiments, all simulation, agent, and domain-specific parameters used in this study are summarised below. The values are taken from the best run (cf. Section 5).

Table A1
Simulation Setup

Parameter	Value	Description
PCBAs per Episode	100	Number of boards generated in each simulation episode
Max Actions	50	Maximum allowed inspection/recovery decisions per episode
Component Count	Uniform 3–10	Number of components sampled per PCBA
Smoothing Factor	0.1	Applied to reward curves for visualisation

Table A2
RL Agent Configuration

Parameter	Value	Description
Learning Rate (α)	$1e^{-4}$	Adam optimiser
Discount Factor (γ)	0.99	Weighting of future rewards
Batch Size	128	Samples per optimisation step
Replay Buffer Size	1000	Maximum stored transitions
Warmup Steps	100	Steps before training begins
Training Frequency	Every step	Optimisation executed after each action
Target Update	Every 10 episodes	Synchronisation of policy and target network

Exploration (ε-greedy)

$$\varepsilon = \varepsilon_{end} + (\varepsilon_{start} - \varepsilon_{end}) \cdot e^{-steps/\lambda} \quad (A.1)$$

Where: $\varepsilon_{start} = 1.0$, $\varepsilon_{end} = 0.01$, $\lambda = 15{,}000$ steps.

Table A3
GCN Architecture

Component	Setting
Hidden Dimension	64 (GCN) / 67 (GAT)
Activation	ReLU
Pooling	Global Mean Pooling
Output	Q-values for all actions

Table A4
Defect Generation Model

Parameter	Value
Defect Types	Solder, Leg, Burned, NoDefect
Initial Probabilities	Random uniform, normalised
Defect Assignment	Single defect per component (exclusive states)
Component Count	Sampled uniformly from 3 to 10

Table A5
Detection Rate per Defect, Cost and Time per Inspection Technology

Tech	Solder	Leg	Burned	Cost	Time
X-Ray	0.99	0.99	0.50	10	5
Visual	0.10	0.90	0.60	1	3
Flying Probe	0.99	0.99	0.99	3 per component	7

Table A6
Recovery Strategies

Strategy	Income	Cost
Reuse	100	0
Repair	100	Repair cost 10
Recycle	10	0

Reward shaping

Dynamic bonuses follow linear decay across episode progress.

References

[1] Baldé C.P., Kuehr R., Yamamoto T., McDonald R., D'Angelo E., Althaf S., 2024. Glob eWaste Monit2024.

[2] Kumar A, Holuszko M, Espinosa DCR. E-waste: an overview on generation, collection, legislation and recycling practices. Resour Conserv Recycl 2017;122:32–42. https://doi.org/10.1016/j.resconrec.2017.01.018.

[3] Lu C, Zhang L, Zhong Y, Ren W, Tobias M, Mu Z, et al. An overview of e-waste management in China. J Mater Cycles Waste Manag 2015;17:1–12. https://doi.org/10.1007/s10163-014-0256-8.

[4] Cucchiella F, D'Adamo I, Lenny Koh SC, Rosa P. Recycling of WEEEs: an economic assessment of present and future e-waste streams. Renew Sustain Energy Rev 2015;51:263–72. https://doi.org/10.1016/j.rser.2015.06.010.

[5] He P, Feng H, Chhipi-Shrestha G, Hewage K, Sadiq R. Life Cycle Assessment of e – Waste Cellphone Recycling. Electron Waste. Wiley; 2022. p. 231–53. https://doi.org/10.1002/9783527816392.ch11.

[6] Ibanescu D, Cailean (Gavrilescu) D, Teodosiu C, Fiore S. Assessment of the waste electrical and electronic equipment management systems profile and sustainability in developed and developing European Union countries. Waste Manag 2018;73:39–53. https://doi.org/10.1016/j.wasman.2017.12.022.

[7] Duflou JR, Boudewijn A, Cattrysse D, Wagner F, Accili A, Dimitrova G, et al. Product clustering as a strategy for enhanced plastics recycling from WEEE. CIRP Ann 2020;69:29–32. https://doi.org/10.1016/j.cirp.2020.04.080.

[8] Umeda Y, Takata S, Kimura F, Tomiyama T, Sutherland JW, Kara S, et al. Toward integrated product and process life cycle planning—an environmental perspective. CIRP Ann 2012;61:681–702. https://doi.org/10.1016/j.cirp.2012.05.004.

[9] Wang L, Wang XV, Gao L, Váncza J. A cloud-based approach for WEEE remanufacturing. CIRP Ann 2014;63:409–12. https://doi.org/10.1016/j.cirp.2014.03.114.

[10] Shittu OS, Williams ID, Shaw PJ. Global E-waste management: can WEEE make a difference? A review of e-waste trends, legislation, contemporary issues and future challenges. Waste Manag 2021;120:549–63. https://doi.org/10.1016/j.wasman.2020.10.016.

[11] Mohsin M., Zeng X., Rovetta S., Masulli F. 2024. Meas Recycl Electron Compon Assist Autom Disassembly Sorting Waste Print Circuit Boards.

[12] He Y, Hao C, Wang Y, Li Y, Wang Y, Huang L, et al. An ontology-based method of knowledge modelling for remanufacturing process planning. J Clean Prod 2020; 258:120952. https://doi.org/10.1016/j.jclepro.2020.120952.
[13] Hauschild MZ, Jeswiet J, Alting L. Design for environment — do we get the focus right? CIRP Ann 2004;53:1–4. https://doi.org/10.1016/S0007-8506(07)60631-3.
[14] Stamer F, Rouven J, Puttero S, Verna E, Galetto M. Integrative inspection methodology for enhanced PCB remanufacturing using artificial intelligence. Procedia CIRP 2024:1–6.
[15] Huang J, Su J, Chang Q. Graph neural network and multi-agent reinforcement learning for machine-process-system integrated control to optimize production yield. J Manuf Syst 2022;64:81–93. https://doi.org/10.1016/j.jmsy.2022.05.018.
[16] Tolio T, Bernard A, Colledani M, Kara S, Seliger G, Duflou J, et al. Design, management and control of demanufacturing and remanufacturing systems. CIRP Ann 2017;66:585–609. https://doi.org/10.1016/j.cirp.2017.05.001.
[17] Zhu T, Liu X, Yu Y, Fu L. Matrix manufacturing system layout and scheduling via graph neural network and multi-action deep reinforcement learning. J Manuf Syst 2025;82:239–53. https://doi.org/10.1016/j.jmsy.2025.06.005.
[18] Gong G, Deng Q, Chiong R, Gong X, Huang H, Han W. Remanufacturing-oriented process planning and scheduling: mathematical modelling and evolutionary optimisation. Int J Prod Res 2020;58:3781–99. https://doi.org/10.1080/00207543.2019.1634848.
[19] Aryan P, Sampath S, Sohn H. An overview of non-destructive testing methods for integrated circuit packaging inspection. Sensors 2018;18:1981. https://doi.org/10.3390/s18071981.
[20] Shafiee Roudbari E, Fatemi Ghomi SMT, Sajadieh MS. Reverse logistics network design for product reuse, remanufacturing, recycling and refurbishing under uncertainty. J Manuf Syst 2021;60:473–86. https://doi.org/10.1016/j.jmsy.2021.06.012.
[21] Ahsan M, Stoyanov S, Bailey C, Albarbar A. Developing computational intelligence for smart qualification testing of electronic products. IEEE Access 2020;8:16922–33. https://doi.org/10.1109/ACCESS.2020.2967858.
[22] Bao Z, Li W, Gao M, Liu C, Zhang X, Cai W. An integrated optimization decision method for remanufacturing process based on conditional evidence theory under uncertainty. IEEE Access 2020;8:221119–26. https://doi.org/10.1109/ACCESS.2020.3042533.
[23] Mejía G, Pereira J. Multiobjective scheduling algorithm for flexible manufacturing systems with Petri nets. J Manuf Syst 2020;54:272–84. https://doi.org/10.1016/j.jmsy.2020.01.003.
[24] Walter G, Flapper SD. Condition-based maintenance for complex systems based on current component status and Bayesian updating of component reliability. Reliab Eng Syst Saf 2017;168:227–39. https://doi.org/10.1016/j.ress.2017.06.015.
[25] Rezaei-Malek M, Mohammadi M, Dantan J-Y, Siadat A, Tavakkoli-Moghaddam R. A review on optimisation of part quality inspection planning in a multi-stage manufacturing system. Int J Prod Res 2019;57:4880–97. https://doi.org/10.1080/00207543.2018.1464231.
[26] Li M, Yao N, Liu S, Li S, Zhao Y, Kong SG. Multisensor image fusion for automated detection of defects in printed circuit boards. IEEE Sens J 2021;21:23390–9. https://doi.org/10.1109/JSEN.2021.3106057.
[27] Wright RG. Multiresolution sensor fusion approach to PCB fault detection and isolation. 2008 IEEE AUTOTESTCON. IEEE; 2008. p. 41–6. https://doi.org/10.1109/AUTEST.2008.4662581.
[28] Puterman ML. Markov Decision Processes. Wiley; 1994. https://doi.org/10.1002/9780470316887.
[29] Stricker N, Kuhnle A, Sturm R, Friess S. Reinforcement learning for adaptive order dispatching in the semiconductor industry. CIRP Ann 2018;67:511–4. https://doi.org/10.1016/j.cirp.2018.04.041.
[30] Magnanini MC, Demir O, Colledani M, Tolio T. Performance evaluation of multi-stage manufacturing systems operating under feedback and feedforward quality control loops. CIRP Ann 2024;73:349–52. https://doi.org/10.1016/j.cirp.2024.04.015.
[31] Magnanini MC, Tolio T. A Markovian model of asynchronous multi-stage manufacturing lines fabricating discrete parts. J Manuf Syst 2023;68:325–37. https://doi.org/10.1016/j.jmsy.2023.04.006.
[32] Stamer F, Lanza G. Dynamic pricing of product and delivery time in multi-variant production using an actor critic reinforcement learning. CIRP Ann 2023;72:405–8. https://doi.org/10.1016/j.cirp.2023.04.019.
[33] Asif ME, Rastegarpanah A, Stolkin R. Robotic disassembly for end-of-life products focusing on task and motion planning: a comprehensive survey. J Manuf Syst 2024;77:483–524. https://doi.org/10.1016/j.jmsy.2024.09.010.
[34] Kristensen CB, Sørensen FA, Nielsen HB, Andersen MS, Bendtsen SP, Bøgh S. Towards a robot simulation framework for e-waste disassembly using reinforcement learning. Procedia Manuf 2019;38:225–32. https://doi.org/10.1016/j.promfg.2020.01.030.
[35] Liu A, Zhang D, Wang Y, Xu X. Knowledge graph with machine learning for product design. CIRP Ann 2022;71:117–20. https://doi.org/10.1016/j.cirp.2022.03.025.
[36] Wu Z, Pan S, Chen F, Long G, Zhang C, Yu PS. A comprehensive survey on graph neural networks. IEEE Trans Neural Netw Learn Syst 2021;32:4–24. https://doi.org/10.1109/TNNLS.2020.2978386.
[37] Hussong M, Ruediger-Flore P, Klar M, Kloft M, Aurich JC. Selection of manufacturing processes using graph neural networks. J Manuf Syst 2025;80:176–93. https://doi.org/10.1016/j.jmsy.2025.02.016.
[38] Subramaniyan M, Skoogh A, Sheikh Muhammad A, Bokrantz J, Turanoğlu Bekar E. A prognostic algorithm to prescribe improvement measures on throughput bottlenecks. J Manuf Syst 2019;53:271–81. https://doi.org/10.1016/j.jmsy.2019.07.004.
[39] Kaelbling LP, Littman ML, Cassandra AR. Planning and acting in partially observable stochastic domains. Artif Intell 1998;101:99–134. https://doi.org/10.1016/S0004-3702(98)00023-X.
[40] Rizova MI, Wong TC, Ijomah W. A systematic review of decision-making in remanufacturing. Comput Ind Eng 2020;147:106681. https://doi.org/10.1016/j.cie.2020.106681.
[41] Sutton RS, Barto AG. Reinforcement Learning: An Introduction. MIT press; 2018.
[42] Orfanoudakis S, Robu V, Salazar EM, Palensky P, Vergara PP. Scalable reinforcement learning for large-scale coordination of electric vehicles using graph neural networks. Commun Eng 2025;4:118. https://doi.org/10.1038/s44172-025-00457-8.
[43] Zhang S, Yao L, Sun A, Tay Y. Deep learning based recommender system. ACM Comput Surv 2020;52:1–38. https://doi.org/10.1145/3285029.
[44] Watkins CJCH, Dayan P. Q-learning. Mach Learn 1992;8:279–92. https://doi.org/10.1007/BF00992698.
[45] Heynicke R, Krush D, Cammin C, Scholl G, Kaercher B, Ritter J, et al. IO-Link Wireless enhanced factory automation communication for Industry 4.0 applications. J Sens Sens Syst 2018;7:131–42. https://doi.org/10.5194/jsss-7-131-2018.
[46] Yin S, Li X, Gao H, Kaynak O. Data-based techniques focused on modern industry: an overview. IEEE Trans Ind Electron 2015;62:657–67. https://doi.org/10.1109/TIE.2014.2308133.
[47] Zhou Y, Yuan M, Zhang J, Ding G, Qin S. Review of vision-based defect detection research and its perspectives for printed circuit board. J Manuf Syst 2023;70:557–78. https://doi.org/10.1016/j.jmsy.2023.08.019.
[48] Ionel R, Mâțiu-Iovan L. Flying probe measurement accuracy improvement by external LCR integration. Measurement 2022;190:110703. https://doi.org/10.1016/j.measurement.2022.110703.