Research on Engineering Structures & Materials
www.jresm.org
MIM Research Group

Research Article

LLM-driven multi-agent collaborative DRL framework and its application to industrial quality inspection decision optimization

Yueguang Wang, Yongfeng Zhang [*,a]

Department of Software Engineering, Hebei Software Institute, Baoding, 071000, China

Article Info

Article History:

Received 21 July 2026

Accepted 21 Sep 2026

Keywords:

Large Language Models (LLM);
Deep Reinforcement Learning (DRL);
QMIX algorithm;
YOLOv8;
Ant Colony Optimization (ACO);
Defect classification and grading;
Multi-agent collaboration;
Flexible manufacturing lines

Abstract

Industrial quality inspection decision optimization faces critical limitations, including loose inter-module coupling and weak multi-agent collaboration when deploying Large Language Models (LLMs) and Deep Reinforcement Learning (DRL). To achieve accurate inspection, intelligent decision-making, and efficient scheduling in intelligent manufacturing scenarios, this study puts forward an improved Large Language Model-driven multi-agent deep reinforcement learning decision optimization framework. The framework uses YOLOv8 to build a perception layer for defect detection and feature extraction. It uses LLaMA to build a semantic layer for defect analysis and knowledge injection. It utilizes the QMIX algorithm for developing the decision layer of many agents for making decisions in collaboration with each other. Furthermore, the framework incorporates Temporal Convolutional Networks (TCN) for temporal feature extraction, K Means clustering for defect grading, and Ant Colony Optimization (ACO) for dynamic workstation scheduling to construct the optimization layer collectively and develops a combined collaborative system encompassing perception, semantics, decision, and optimization. Experimental results show that the proposed framework achieves a mAP@0.5 of 97.82% on the fused dataset combining the public MVTec AD dataset and the private automotive parts quality inspection dataset. The quality inspection decision accuracy and defect grading accuracy reach 96.35% and 95.78%, respectively, outperforming conventional DQN and VDN based multi agent DRL baselines by 13.00% and 9.97%, respectively. The scheduling path length decreases by 23.60%. The framework also shows better convergence speed and load balancing performance than the comparison methods. Although the overall computational overhead imposes challenges for direct deployment on resource-constrained industrial edge terminals, the proposed YLQTAK framework provides an effective technical solution for edge side near real time collaborative decision making (under cloud edge collaborative mode) of multi-workstation quality inspection in flexible production lines and points out a feasible lightweight edge-cloud collaborative deployment direction for follow-up research.

© 2026 MIM Research Group. All rights reserved.

1. Introduction

Industrial quality inspection is a key process for ensuring product quality, reducing production cost, and improving production efficiency. Its intelligence level directly determines the core competitiveness of the manufacturing industry [1]. However, traditional industrial quality inspection relies heavily on manual labor or standalone computer vision systems. These conventional inspection mechanisms exhibit severe limitations, including high false-positive rates under varying operational environments, poor generalization across diverse defect geometries,

*Corresponding author: zhangyongfeng0922@126.com
[a]https://orcid.org/0009-0005-3421-0596
DOI: http://dx.doi.org/10.17515/resm2026-1965vk0721rs

and a total lack of inter-station communication and dynamic decision scheduling. Consequently, such inspection mechanisms cannot meet modern production requirements for high-precision quality inspection [2-3]. With the rapid development of Artificial Intelligence (AI), the integrated application of Deep Reinforcement Learning (DRL), Large Language Model (LLM), and computer vision provides a new technical path for industrial quality inspection decision optimization [4-5]. For instance, LLM possesses robust semantic understanding, knowledge reasoning, and cross-modal interaction capabilities. By converting visual feature vectors into standardized text prompt representations, it can achieve domain-specific process knowledge injection, defect textual description, explainable decision-making, and semantic communication among distributed multi-agent systems. However, the integrated application of LLM and DRL in industrial quality inspection still has the following two limitations: Firstly, most of the existing research has combined LLM and DRL into a loose sequential pipeline, resulting in low coupling between modules, information transmission bottlenecks, and insufficient knowledge-decision coupling [11]. Secondly, the current multi-agent DRL mechanism research is not mature enough, mainly focusing on single-agent decision-making or simple multi-agent task allocation, and it is difficult to achieve global collaborative decision-making [12].

To address the aforementioned industrial issues and technical limitations, a modified LLM-driven multi-agent DRL integrated decision-making framework was designed and named YLQTAK. This framework uses YOLOv8 as the perception layer for extracting defect features, employs the Large Language Model Meta AI (LLaMA) for constructing the semantic layer to achieve defect knowledge reasoning, and utilizes the Q-Mixing (QMIX) algorithm as the decision-making layer to complete multi-agent collaborative decision-making. The temporal convolutional network (TCN), K-Means clustering algorithm (K-Means), and ant colony optimization algorithm (ACO) together form the optimization layer, which is responsible for temporal feature learning, defect automatic classification, and workstation scheduling. The innovation and main contributions of the research lie in the following three points:

- Special architecture design: A hierarchical YLQTAK framework is proposed, integrating the semantic understanding module based on LLaMA with QMIX multi-agent, solving the problem of loose coupling and independent usage of LLM and DRL modules in existing research.
- Optimization layer mechanism innovation: A combined optimization mechanism of TCN+K-Means+ACO is designed, introducing production line temporal prior and defect perception clustering prior into the multi-agent decision-making process, achieving automatic defect classification and dynamic workstation scheduling for complex industrial quality inspection scenarios.
- Experimental verification and performance quantification: Experiments were conducted based on the fused dataset composed of the public MVTec AD dataset and real automotive component quality inspection data. The YLQTAK framework outperformed multiple baseline comparison algorithms in terms of detection accuracy (mAP@0.5=97.82%), decision performance (Decision precision=96.35%), convergence speed, and load balancing capability.

2. Literature Review

2.1 LLM Applications in Industrial Quality Inspection

LLMs, with their powerful capabilities in semantic understanding, knowledge reasoning, and instruction generation, can enable semantic parsing of industrial quality inspection, textual description of defect information, and semantic communication among multiple agents, thereby improving model interpretability and convergence speed. However, there are still several practical challenges when applying LLM to industrial quality inspection scenarios: there is a risk of hallucination when processing industrial visual defect data, high inference latency hinders real-time response, and the ability to transmit semantic information across intelligent agents is insufficient. Based on this, numerous scholars have conducted research in this area. For example, since there is no domain-specific knowledge in LLMs, as well as their low capability of

understanding the local information of objects, Gu Z et al. designed an anomaly detection model in the industry using an improved LLM and discovered that the precision rate on MVTec-AD was 86.1% [6]. Fan H et al. designed a three-layer detection framework based on LLM that achieved a success rate of 81.88% in task planning by matching manufacturing tasks with process parameters and autonomously designing tool paths [7]. Liu P F et al. designed a knowledge-enhanced joint model by adding an aviation assembly knowledge graph into LLM to simplify computation and found that the model had a maximum accuracy rate of 98.5% in industrial fault detection [8]. Thanks to the rapid development of multimodal visual-language large models, existing research has bridged the gap between visual defect observation and text reasoning: VLM can directly receive defect image input and output structured defect descriptions and grading suggestions, eliminating the intermediate step of manually designing prompt words. However, Jiang X et al. demonstrated in the benchmark evaluation of industrial anomaly detection covering all tasks that the average accuracy of the current strongest commercial multimodal large model is only 74.9%. Open-source VLM faces problems such as high inference costs and insufficient adaptation to small-scale domain defect datasets [9].

2.2 DRL and Multi-Agent Systems in Manufacturing Decision Optimization

DRL, with its advantages in autonomous exploration and decision optimization in dynamic environments, has been preliminarily applied in the field of industrial quality inspection and has become an important technical means for solving quality inspection decision optimization problems. For instance, Chen M proposed a novel decision model and a DRL-based maintenance optimization method for industrial component production systems, which reduced the interdependencies among multiple components, and the effectiveness of the method was validated through numerical experiments [10]. To enhance the flexibility, decision-making efficiency, and scheduling performance of the workshop, Xing J et al. proposed an intelligent workshop distributed scheduling method based on multi-agent DRL (QMIX), effectively achieving the collaboration between the workshop and production goals [11]. Zhang J et al. analyzed the scheduling tasks of automatic guided vehicles (AGVs) in the workshop, modeled the workshop as a node network, and used the improved multi-agent reinforcement learning (the improved QMIX model) to solve the AGV scheduling problem. The research found that this method outperformed other common methods in terms of the maximum completion cycle of AGVs under different environmental conditions [12]. Wan L et al. proposed a multi-agent graph reinforcement learning method using AEA-MAPPO, effectively solving the scheduling problem of flexible operation workshops and optimizing the workshop operation sequence and machine scheduling allocation decisions [13].

2.3 Summary of Existing Gaps and Motivations

In summary, even though some achievements have been made by researchers through studying LLMs and DRLs in the field of industrial quality inspection, some limitations exist in the application of these two technologies in the actual process of optimizing industrial quality inspection decision making:

- The existing research on the integration of LLM and DRL generally adopts a loose serial pipeline architecture. This issue is reflected in the studies by Gu Z et al., Jiang X et al., and Xing J et al. Among them, the visual-language model proposed by Jiang X et al. can generate defect text descriptions, but it does not establish an explicit feature-level interface to inject semantic knowledge into the reinforcement learning state space, resulting in the semantic output being regarded merely as an auxiliary text log rather than an observable state input by the agent. This forms an information bottleneck, leading to insufficient coupling between knowledge and decision-making.
- Currently, the research on multi-agent reinforcement learning scheduling mostly focuses on the task allocation of workshop logistics and AGV scheduling in fully observable environments. There is a lack of extending the multi-agent value decomposition mechanism to industrial quality inspection tasks with heterogeneous multimodal inputs, and the existing research fails to achieve the collaborative joint optimization of perception-semantic-scheduling, resulting in limited global collaborative decision-making ability among distributed quality inspection stations.

- The existing research on manufacturing optimization mostly adopts isolated single-algorithm solutions: visual inspection, defect clustering, and workstation scheduling are all implemented independently, lacking a unified feedback fusion mechanism and an overall framework that integrates perception, semantic analysis, decision-making, and scheduling. This has restricted the decision accuracy and engineering implementation performance in real scenarios.

To address the aforementioned research gaps, the study integrates YOLOv8, LLaMA, QMIX, TCN, ACO and K-Means, and designs an industrial quality inspection decision optimization framework that combines data perception, semantic parsing, decision optimization and intelligent scheduling. YOLOv8 provides fine-grained visual defect feature extraction, supporting downstream semantic encoding and clustering. LLaMA converts visual features into semantic vectors with domain knowledge, compensating for the deficiency of pure reinforcement learning in decision interpretability. QMIX realizes centralized training - distributed execution multi-agent collaboration, compensating for the limitation of single-agent DRL in handling multi-station joint decision-making. TCN is used to extract production line temporal dependency, alleviating environmental non-stationarity. K-Means is used for unsupervised defect classification, outputting task priority priors, and ACO provides heuristic global scheduling optimization. These six modules together form a mutually compensating feature flow chain: the visual perception output flows into the semantic module and clustering module, the semantic features and temporal features are concatenated to form the QMIX joint state space, the defect classification results guide the ACO scheduling, and the scheduling feedback reversely corrects the multi-agent reward signal. The research aims to achieve an optimized synergy through the collaborative use of multiple algorithms, specifically addressing the issues of insufficient coordination, low accuracy, and poor implementation in industrial quality inspection decision-making.

3. Framework Design For LLM-driven Industrial Quality Inspection Decision-making

3.1 Overall Framework

Optimization of industrial quality inspection decisions involves efficient decision-making and global optimization of a sequence of industrial quality inspection operations. This can be done by leveraging intelligent manufacturing. The intention is to ensure that there is better inspection accuracy and reduced cost of operations. Therefore, to realize full-process intelligence from perception and understanding to decision-making and optimization in industrial quality inspection, this study uses YOLOv8 as the perception layer, LLaMA as the semantic layer, QMIX as the decision layer, and TCN, ACO, and K-Means as the optimization layer. Then, this study designs the overall framework shown in Figure 1 and names it the YLQTAK framework. Figure 1 shows that the perception layer takes product images collected from industrial sites as input. YOLOv8 completes defect target detection, location positioning, and feature extraction, and outputs defect categories, coordinates, confidence scores, and visual feature vectors. The semantic layer uses LLaMA to perform defect semantic analysis and multi-agent semantic communication. It converts visual features into interpretable and inferable standardized semantic knowledge. The decision-making layer incorporates perception attributes and semantics into the joint state space, implementing the QMIX approach for central learning and producing globally optimal decisions for quality inspections. The optimization layer implements the TCN algorithm, along with K-Means clustering and ACO algorithms, to extract temporal dependency attributes between the operating conditions, automatically classify defect attributes, and optimize quality inspection paths and workstation scheduling. In this work, the defect visual feature vector is explicitly extracted from the 1×1 convolutional feature map prior to the final classification and regression branches of the decoupled detection head, instead of using classification or regression output logits directly.

The defect visual feature vectors output by the perception layer of YOLOv8 are transformed in dimension through a learnable projection layer: first, the visual features are projected to a unified latent space, and then the projected visual representations are sent to LLaMA and QMIX respectively. LLaMA, combined with domain knowledge, further encodes the projected visual

representations to generate defect semantic feature vectors, while TCN extracts the time series features of the production line conditions through convolution. Finally, the perception attributes (projected visual features), semantic features, and time series features are concatenated to jointly construct the joint state space for multi-agent decision-making, enabling the compatibility of multi-source heterogeneous features.

Furthermore, considering the real-time constraints of the on-site flexible production line and the inherent inference delay brought by the large-parameter LLaMA model, the research did not adopt an end-to-end serial blocking execution for each input product sample. Instead, it adopted a synchronous-asynchronous hybrid execution mechanism combined with cloud-edge collaborative inference to alleviate the pressure of latency. The details are as follows: ① The lightweight modules YOLOv8, TCN, and K-Means are deployed and run synchronously at the industrial edge end to complete defect detection, temporal feature extraction, and defect clustering; ② Enhanced retrieval generates pre-cached industrial defect knowledge fragments, and the LoRA-optimized LLaMA only performs semantic encoding when a new defect category or unseen defect feature is encountered; ③ For known defect samples that occur repeatedly, the semantic feature vectors pre-computed in the cache are directly reused without calling the large model for inference at each step. ④ The ACO scheduling heuristic calculation and QMIX multi-agent decision-making are triggered based on the visual semantic temporal combined state after caching, ensuring that the output at the edge end is immediately executable.

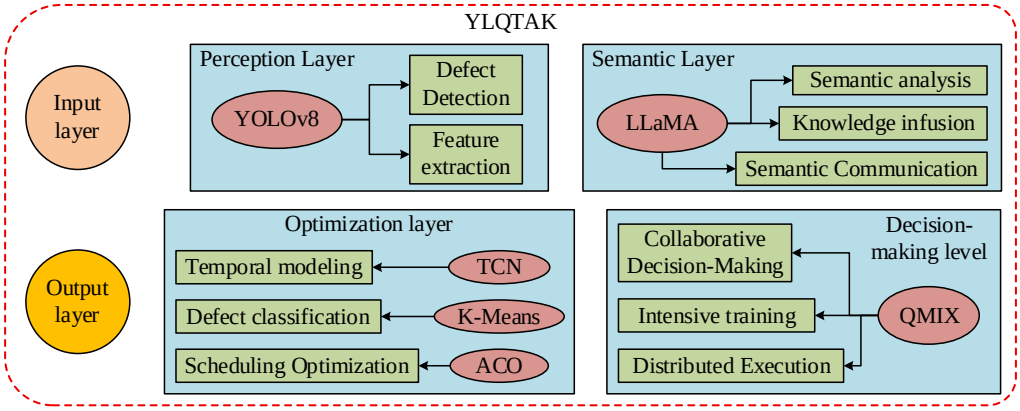


Fig. 1. Overall architecture of the YLQTAK framework

3.2 YOLOv8-based Industrial Defect Perception Model

Excellent defect features lay the foundation for further semantic understanding and decision-making. Therefore, the first step involved in the current study involves creating the perception part using the YOLOv8 technique to enable efficient detection and feature extraction of industrial image defects. The specific detection process is shown in Figure 2.

As depicted in Figure 2, YOLOv8 uses industry defect images as its inputs. Following pre-processing and augmentation, the images will be fed into the backbone that comprises of multiple C2f modules that perform multi-scale feature extraction. Next, the neck is tasked with the fusion of semantic information from various levels. Finally, the decoupled detection head completes defect image classification and bounding box regression in parallel and outputs defect categories, coordinates, and confidence scores. It also generates defect visual feature vectors for subsequent semantic analysis and decision optimization. The total loss function of YOLOv8 consists of classification loss, confidence loss, and localization loss with weighted summation. The function expression is shown in Equation (1) [14].

$$L = \lambda_1 L_{\text{cls}} + \lambda_2 L_{\text{box}} \tag{1}$$

In Equation (1), L represents the total loss function of YOLOv8. L_{cls} , and L_{box} represent classification loss, and localization loss, respectively. λ_1, λ_2, and λ_3 are both weight coefficients, used to balance the contribution ratios of the three losses during the training process. They are set

to 0.5, 1.0 and 0.8 respectively. Unlike the earlier YOLO series, YOLOv8 has removed the separate target confidence loss branch, and the confidence information is implicitly learned within the classification head. YOLOv8 usually uses Complete Intersection over Union (CIoU) as the localization loss function to improve defect localization accuracy and convergence stability. The mathematical expression is shown in Equation (2).

$$L_{\mathrm{CIoU}} = 1 - \mathrm{IoU} + \frac{\rho^2(b, b^{gt})}{c^2} + \alpha v \tag{2}$$

In Equation (2), L_{CIoU} is defined as the CIoU loss function. $\rho(\cdot)$is the Euclidean distance between center points. c represents the diagonal length of the minimum enclosing rectangle. v denotes width-height ratio consistency. α is the balance coefficient.

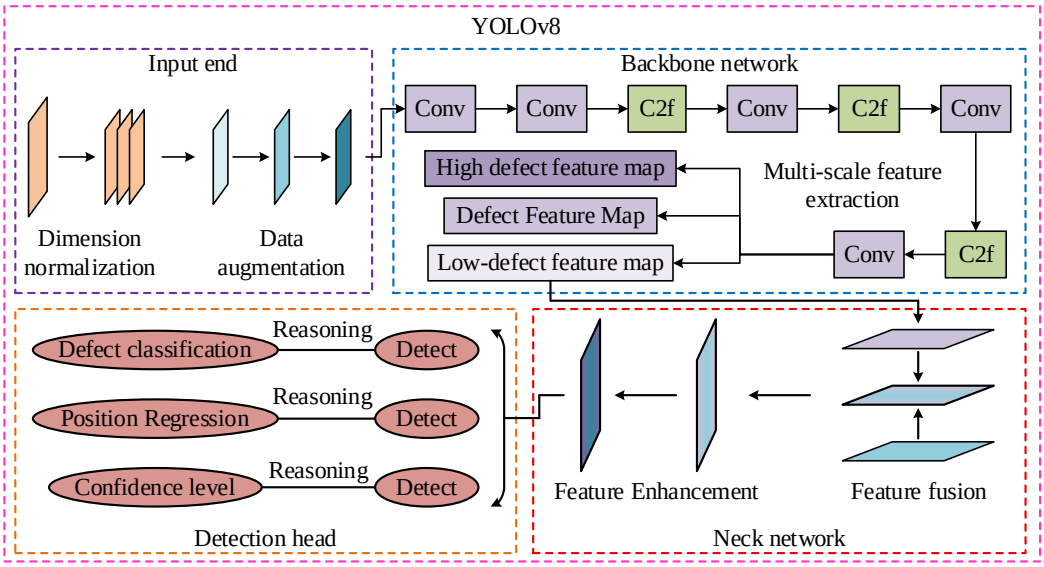


Fig. 2. Front-end perception layer based on YOLOv8

3.3 LLaMA Quality Inspection Semantic Analysis and Knowledge-Driven Mechanism

To convert visual detection results into inferable and communicable structured knowledge, this study uses the lightweight LLaMA model to conduct semantic analysis and domain knowledge injection of defect information. The specific process is shown in Figure 3.

In Figure 3, LLaMA applies structured processing on visual defect characteristics to create texts. In accordance with the limitations of domain knowledge in industry, it transforms visual results to vector forms that can be recognized by the model. Thereafter, multi-layer multi-head self-attention techniques and feedforward networks conduct the task of semantic coding and feature fusion of defect characteristics to produce the text description of defects. Finally, semantic features are mapped into standardized decision probabilities, and defect semantic feature vectors are output.

The research introduces a lightweight visual encoder module to establish a bridge between the defect visual feature vectors and the purely text LLaMA decoder: The defect visual feature vectors output by the perception layer of YOLOv8 are mapped to the embedding dimension through a learnable linear projection layer, and then concatenated with the manually designed defect description template prompts to construct a multimodal prompt sequence. This enables implicit image-to-text conversion without performing full-scale weight fine-tuning of visual-language.

Furthermore, the study employed LoRA fine-tuning combined with the bitsandbytes library to achieve 4-bit quantization, adapting the basic LLaMA-7B model for tasks in industrial quality inspection: The basic model weights were loaded and frozen in a 4-bit quantized form, and only the attention weight matrix of the decoder was updated through low-rank adaptation. This significantly reduced the memory usage during both fine-tuning and inference stages. The injection of domain knowledge adopted a hybrid paradigm of context prompts and retrieval enhancement

for generating: a knowledge base of industrial defect processing was pre-built, and corresponding manufacturing process rules were retrieved based on the defect category and geometric attributes. The retrieved knowledge fragments were appended to the prompt context as constraints for the LLaMA inference. The specific process of semantic encoding carried out by LLaMA is accomplished by a multi-layer independent decoder Transformer module composed of multiple heads of self-attention and feedforward networks, as shown in Equation (3) [15].

$$\begin{cases} h_t = TransformerDecoderBlocks(W_e x; \Theta_{LLaMA}) \\ f = Pool(h_t[t,;]) \end{cases} \tag{3}$$

In Equation (3), h_t represents the complete hidden state sequence output by the Transformer layer, f is defined as the output defect semantic feature vector, x represents the multimodal structured input sequence composed of visual embeddings from the projection, template prompts, and RAG retrieval of domain knowledge segments, W_e and Θ_{LLaMA} respectively represent the feature embedding matrix and the network parameters of LLaMA, and $Pool(\cdot)$ represents the sequence pooling operation. LLaMA realizes defect text description, process rule matching, and defect grade reasoning through iterative updating of hidden layer states. The process is shown in Equation (4) [16].

$$p_{\text{semantic}} = \text{Softmax}(W_o \cdot h_t + b_o) \tag{4}$$

In Equation (4), $p_{semantic}$ represents the probability distribution of defect grades or the confidence of process rule matching. W_o is the weight matrix. $Softmax(\cdot)$ represents the normalized exponential function and is used to output standardized decision probabilities.

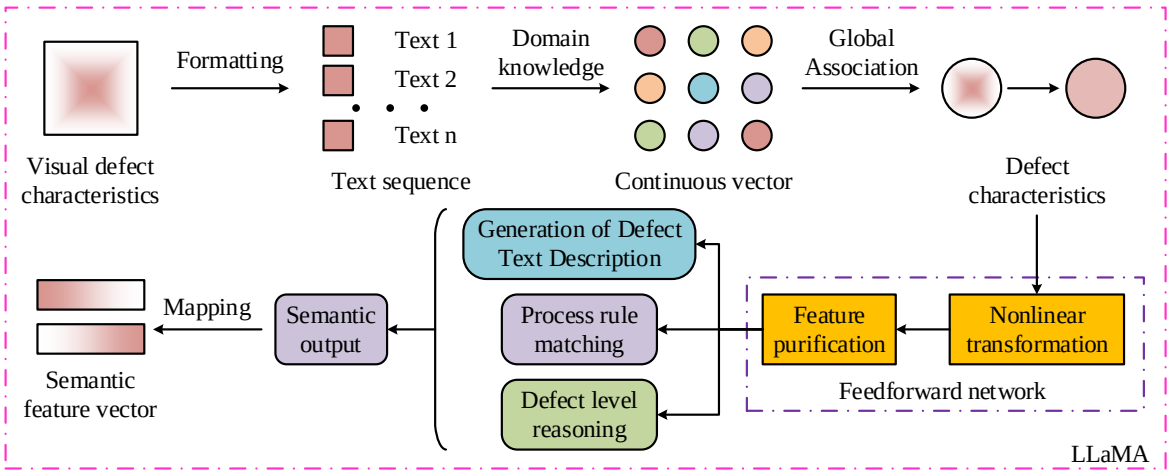


Fig. 3. Process of LLaMA semantic analysis and knowledge injection

3.4 QMIX Multi-agent DRL Decision Model

In the context of the dual inputs of visual features and semantic information, this study develops a multi-agent collaborative decision-making framework using the QMIX algorithm to achieve distributed collaboration in the task of quality inspection across multiple workstations. This multi-station quality inspection collaboration problem is formally modeled under the decentralized partially observable Markov decision process framework. The relevant definitions are as follows:

- Agent: Each physical quality inspection station deployed on the flexible production line is an independent decision-making agent. Cameras and mechanical actuators belong to the perception-execution peripheral devices and do not serve as decision-making agents.
- Global state space: The global state input to the QMIX hybrid network is composed of the following three types of features: the defect visual feature vector output by the YOLOv8 perception layer, the defect semantic feature vector obtained by LLaMA encoding, and the production line temporal working condition features extracted by TCN.

- Local observation space: It includes the local defect visual embedding vector of the station, the defect semantic vector obtained by station analysis, and the temporal working condition features of the adjacent production line part.
- Action Space of Intelligent Agents: The discrete action space of each intelligent agent includes release, scrap, rework, transfer to the next workstation, and reducing the production line speed. Release represents that the current product passes the quality inspection, scrap indicates that defective products are scrapped, rework means initiating on-site repair operations, transferring to the next workstation means handing over the product to the downstream quality inspection workstation, and reducing the production line speed is for high-risk defect samples to lower the operating rate of the production line.
- Joint Reward Function: The global joint reward optimized by QMIX is a weighted composite function that integrates decision accuracy, quality inspection delay, and workstation load balance penalties, and its mathematical expression is as follows: $R_{joint} = \omega_1 \cdot R_a cc - \omega_2 \cdot R_{latency} - \omega_3 \cdot R_{load}$ the formula, R_{acc} represents the binary reward for correct quality inspection decisions, $R_{latency}$ is the normalized penalty term for task quality inspection delay, R_{load} indicates the penalty term for load imbalance among all workstations, and ω is defined as a non-negative weight parameter used to balance the three optimization objectives.

The specific operating mechanism is shown in Figure 4. As shown in Figure 4, the multi-agent decision-making model takes defect visual features, semantic features, and other input information as joint input. Each quality inspection agent calculates its individual action value, and the mixing network combines the individual values into a global joint action value. The weight parameters of this mixing network are not fixed; they are dynamically generated by hypernetworks conditioned on the global state, and constrained to be non-negative to realize QMIX's core monotonic value-decomposition property. At the same time, the network parameters are iteratively refined in such a way that their global value is maximized. This model generates collaborative actions for quality control inspections at various stations and constantly undergoes iterative refinement by interacting with the environment and obtaining feedback. Finally, this model achieves globally optimal multi-agent collaborative quality inspection decision-making. The process in which QMIX outputs the action value function of a single agent is shown in Equation (5) [17].

$$Q_a(\tau_a, u_a; \theta_a) = f_{agent}(\tau_a, u_a, \theta_a) \tag{5}$$

In Equation (5), Q_a represents the action value function of the a -th agent. τ_a denotes local observation. u_a is the action taken by the agent. θ_a represents the parameters of the individual value network. f_{agent} denotes the mapping function of the agent network. QMIX outputs the function expression of the global joint value through a mixing network with monotonic constraints, as shown in Equation (6) [18].

$$Q_{\text{tot}}(\tau, u; \theta) = f_{\text{mix}}(Q_1, Q_2, \ldots, Q_n, s; \theta') \tag{6}$$

In Equation (6), Q_{tot} is the global joint action value, Q_n is the individual action value of each agent. s is the global state, and θ' is the parameters of the mixing network. f_{mix} is used to ensure the monotonic increase relationship between global value and individual value so that distributed decision-making can approximate the global optimum. Compared with ordinary feedforward networks, QMIX introduces an auxiliary super-network mechanism: it takes the global state as input and dynamically generates the weight parameters of the hybrid network. It also applies ReLU activation or absolute value operation to the output of the super-network to ensure that the weights of the hybrid network are non-negative, thereby satisfying the monotonicity condition between the global joint value and the individual action values of each agent at the mathematical level.

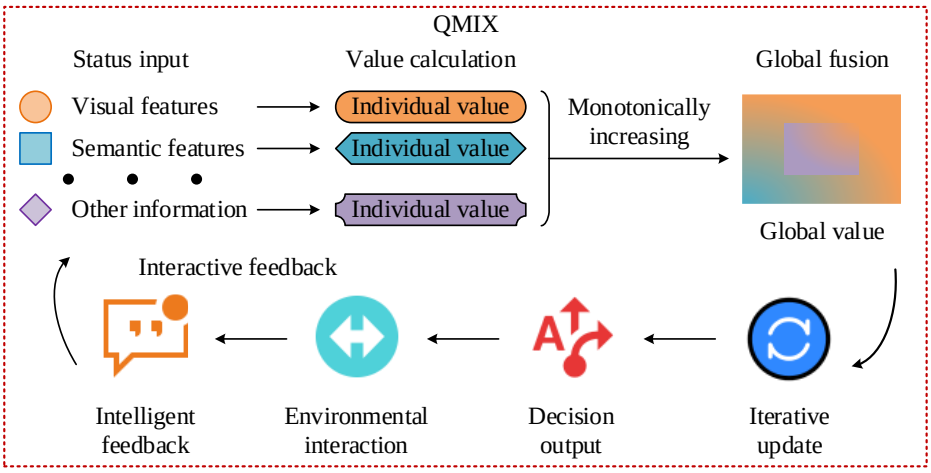


Fig. 4. Decision model based on the QMIX algorithm

3.5 TCN-K-Means-ACO Joint Optimization Mechanism

In order to optimize the decision-making effectiveness of industrial quality inspections, improve convergence effects, and achieve automatic grading of defects, a joint optimization layer combining TCN, K-Means, and ACO is proposed in this study for providing the decision model with temporal information, clustering priors, and scheduling priors. Its specific architecture and main functions are shown in Figure 5.

In Figure 5, the joint optimization layer chooses a three-branched operational architecture of TCN, K-Means, and ACO with unified feedback fusion. The first branch processes data from the production line using TCN. After preprocessing, dilated convolution, and temporal encoding, the output will be the features of long-term operation conditions. The second branch uses defect visual features as input. The process of initializing centers, calculating distances, clustering samples, and iterative updates will automatically classify defects. The ACO branch abstracts quality inspection tasks and workstations into a scheduling path graph. Through pheromone initialization, probability transition, and pheromone updating, it realizes quality inspection scheduling path optimization. The process in which TCN extracts temporal operating condition features of industrial production lines through dilated convolution and residual structures is shown in Equation (7) [19].

$$X_l = \sigma(W_l * D(X_{l-1}) + b_l) + X_{l-1} \tag{7}$$

In Equation (7), X_l is the output feature of the l -th layer. $*_d$ represents the expansion convolution operator with an expansion factor of d. W_r is the convolution kernel for the residual branch that can learn a 1×1 projection, which is used to map the output features to the same output channel number as the main convolution branch, avoiding the problem of mismatched matrix dimensions when performing residual addition. The function expression of the final temporal feature output by TCN is shown in Equation (8).

$$h_t = TCN(x_{t-S+1}, \cdots, x_t) \tag{8}$$

In Equation (8), h_t is defined as the temporal feature. $O_{1:S}$ represents an industrial quality inspection scenario sequence with a length of S, including the real-time speed of the production line, equipment temperature, and historical task completion records. The mathematical expression in which K-Means realizes automatic grading of defect features by minimizing the sum of squared errors is shown in Equation (9) [20].

$$J = \sum_{i=1}^{K} \sum_{x \in Ci} \|x - \mu_i\|^2 \tag{9}$$

In Equation (9), J represents the total clustering error. K is the number of defect grades. C_i denotes the i -th defect sample dataset. μ_i represents the i -th clustering center. Each defect sample is assigned to the cluster center closest to it to obtain the discrete defect level index. This index is further encoded to generate a low-dimensional defect level embedding vector, which serves as the input for the subsequent multi-agent decision-making process.

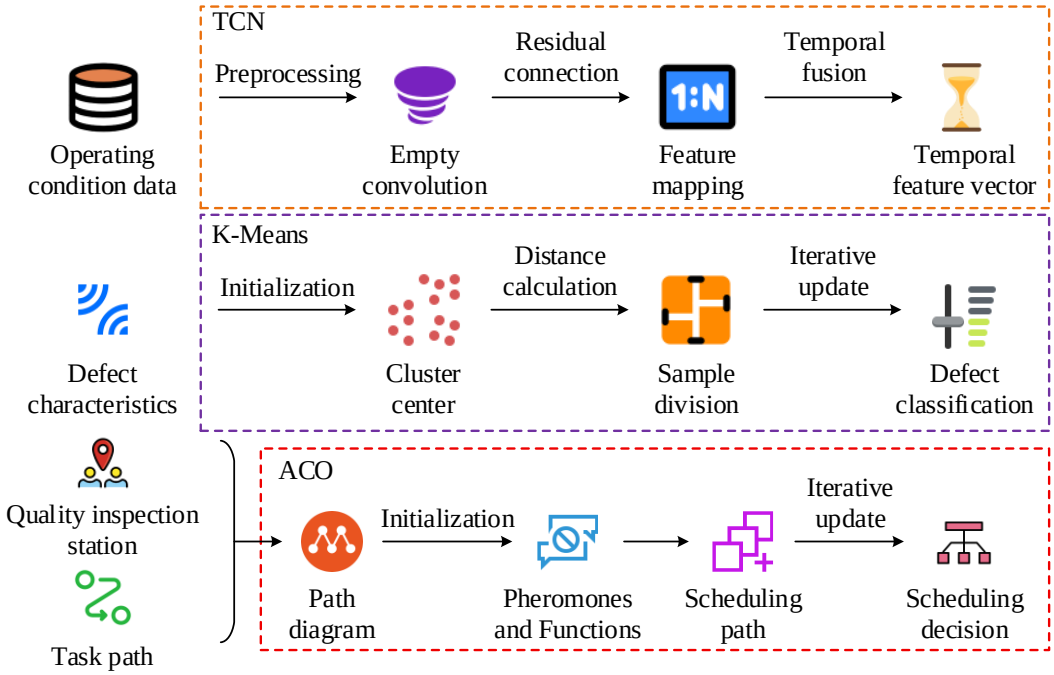


Fig. 5. Architecture of the joint optimization layer

The iterative update rule of the K-Means clustering center is shown in Equation (10).

$$\mu_i^{'} = \frac{1}{|C_i|} \sum_{x \in C_i} x \tag{10}$$

In Equation (10), $\mu_i^{'}$ represents the updated clustering center. ACO simulates ant foraging behavior to realize path and workstation scheduling optimization for industrial quality inspection tasks. The state transition probability is shown in Equation (11).

$$P_{i'j}^{k} = \frac{\gamma_{i'j}^{\alpha} \cdot \eta_{i'j}^{\beta}}{\sum_{l' \in allowed_k} \gamma_{i'l'}^{\alpha} \cdot \eta_{i'l'}^{\beta}} \tag{11}$$

In Equation (11), $P_{i'j}^{k}$ is the transition probability of ant k from workstation i' to j'. $\gamma_{i',j}$ represents pheromone. α and β both are weight coefficients. And $\eta_{i',j}$ represents the heuristic function for customizing workstation scheduling tasks: $\eta_{i',j} = \frac{1}{(\lambda_1 \cdot D_{i',j} + \lambda_2 \cdot T_{i',j} + \lambda_3 \cdot Y_{i',j})}$; among which $D_{i',j}$ is the physical space distance between workstations, $T_{i',j}$ is the processing time for the workstation tasks, $Y_{i',j}$ is the current real-time load intensity of the workstation, and λ represents a non-negative weight coefficient, used to balance the influence of distance, processing time and load status.

The global pheromone update mechanism of ACO is shown in Equation (12) [21].

$$\gamma_{i'j} = (1-\rho)\gamma_{i'j} + \sum_{k=1}^{m} \Delta\gamma_{i'j}^{k} \tag{12}$$

In Equation (12), ρ is the pheromone evaporation coefficient. m is the number of ants. $\Delta\gamma_{i'j}^{k}$ represents the pheromone released by the k -th ant on path (i', j). Extract the candidate priority encoding of the workstations from the converged ACO scheduling solution, and obtain the

scheduling priority embedding vector, which is used to represent the scheduling priority order of each candidate workstation for the current defective task.

Furthermore, in order to send the multi-branch outputs of the optimization layer into the global state space of QMIX, the study adopted vector concatenation as the feature fusion operator, and the augmented global state input to the QMIX hybrid network is shown in Equation (13).

$$s^* = \text{Concat}\left(h^*_{\text{vis}}, h^*_{\text{sem}}, h_t^{\text{TCN}}, e_f, e_{\text{ACO}}\right) \tag{13}$$

In Equation (13), h^*_{Vis} represents the visual feature of the defect after projection, h^*_{sem} is the semantic feature vector output by LLaMA, h_t^{TCN} is the temporal feature vector output by TCN, e_J is the defect grade embedding obtained by K-Means clustering, and e_{ACO} is the scheduling priority embedding obtained by ACO for optimizing the sequence of positions. All these feature vectors are first mapped to a unified latent space dimension through a learnable linear projection layer to ensure the dimension alignment when splicing heterogeneous features from multiple sources.

4. Experimental Results and Analysis

4.1 Environment and Dataset

For the verification of the efficiency of the YLQTAK architecture, the following equipment was used in this experiment: the Intel Xeon Gold 6330 processor, the NVIDIA RTX 4090 graphics card, and the memory capacity of 1 TB. In addition, the following software stack was used for testing: the operating system is Ubuntu 22.04, the programming language is Python 3.9, and the deep learning framework is PyTorch 2.0, with additional libraries including OpenCV, Transformers, and Stable-Baselines3.

The datasets selected were the MVTec AD public industrial defect dataset and quality inspection data of automobile parts collected from actual production sites. The former contained over 5000 high-resolution images, covering 5 texture categories such as carpet and leather, and 10 object categories such as cables, screws, and springs, making it suitable for industrial quality inspection scenarios. This data contained 12 kinds of defects found in industries, namely cracks, scratch, pits, deformation, among others, with 18,600 images altogether.

To make the MVTec-AD dataset compatible with the supervised defect detection training of YOLOv8, the study generates defect bounding boxes from pixel-level anomaly segmentation masks: For each defect sample in the MVTec-AD test set, the minimum enclosing rectangle of the connected domain anomaly mask is calculated, a YOLO format bounding box annotation is generated, and the transformed defect image-label pairs are fused with the private automotive component dataset. The original defect-free training images of MVTec-AD are retained as negative samples. The automotive component dataset exhibits a class imbalance issue: crack and scratch samples account for 41.2% of all images, while samples of some rare defect categories only account for 2.1% to 4.7%. To alleviate the bias caused by the class imbalance, during the training stage, Focal Loss is used to calculate the detection loss, and online random data augmentation methods such as random flipping, rotation, and brightness perturbation are combined to increase the proportion of samples of the defect categories. The two datasets were split up into train, validation, and testing datasets in the ratio of 7:2:1, respectively. Moreover, 15,000 sequences of time-series operational data, consisting of temperature, speed, and cycle times in the industrial production line, as well as 500 rules for the process, were collected.

To ensure the reproducibility of the experimental results, the study adjusted the hyperparameters of each module in the YLQTAK framework. For detailed information, please refer to Table 1. Furthermore, to avoid the problem of memory overflow (OOM) in the 24GB RTX 4090 graphics card, the research adopted several memory-saving strategies: ① The LLaMA 7B base model was loaded in a 4-bit quantization manner, combined with efficient fine-tuning of LoRA parameters, with the original pre-trained weights frozen and only the LoRA adapter parameters updated; ② YOLOv8, QMIX, and TCN were trained in FP16 mixed precision; ③ Gradient checkpointing was enabled for LLaMA and TCN to reduce the memory overhead of activation values; ④ High-load

LLaMA inference and multi-agent gradient updates were not performed simultaneously in the same GPU iteration.

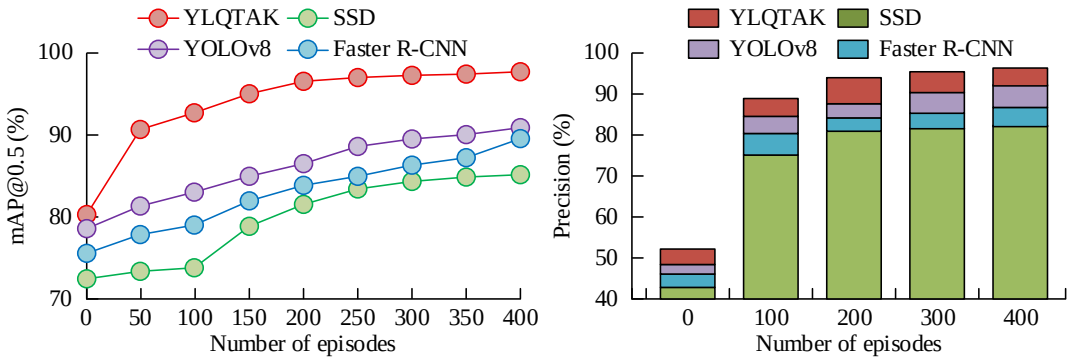
Table 1. Key hyperparameter configuration table

Module	Hyperparameter	Value
YOLOv8	Initial learning rate	0.001
	Batch size	16
	Total training epochs	400
	Weight decay	0.0005
LLaMA	Fine-tuning learning rate	2×10^{-5}
	LoRA rank	8
	Maximum sequence length	256
QMIX	Discount factor	0.95
	Replay buffer size	100000
	Greedy initial value	0.9
	Greedy decay rate	1×10^{-5}
	Learning rate	0.0005
TCN	Dilated convolution dilation rates	[1,2,4,8]
	Kernel size	3
	Dropout rate	0.1
K-Means	Number of defect clusters	4
	Maximum iteration	300
ACO	Pheromone evaporation coefficient	0.1
	Heuristic weight	2.0
	Pheromone weight	1.0
	Ant population size	20

4.2 Industrial Defect Perception Accuracy Evaluation

To test the defect detection and feature extraction performance of the perception layer in the YLQTAK framework, this study used single YOLOv8, Faster Region-based Convolutional Neural Network (Faster R-CNN), and Single Shot MultiBox Detector (SSD) as comparison models. The mean Average Precision at IoU threshold 0.5 (mAP@0.5) and precision were used as evaluation metrics, and the detection results are shown in Figure 6.

As shown in Figure 6(a), the detection model in the YLQTAK framework approached convergence after 200 episodes, and the mAP@0.5 reached 97.82% at 400 episodes, significantly higher than the three comparison models. This indicated that the detection model in the perception layer of the YLQTAK framework, with multi-scale features and semantic constraints, achieved more stable training and higher detection accuracy. According to the results presented in Figure 6(b), the precision of YLQTAK achieved 89.67% at 100 episodes, and then increased to 96.15% at 400 episodes, outperforming the compared models in the same training process. It can be concluded from above that the proposed joint learning perception model effectively lowered the false alarm rate in practical industry defect detection applications. Compared with the single YOLOv8 that only optimizes based on loss detection, the perception module in YLQTAK introduces semantic alignment auxiliary loss during the model training stage: it uses the semantic features of LLaMA as a soft supervision signal, and the gradient is back-propagated from the high-level semantic constraints through a learnable projection layer to the YOLOv8 backbone network and detection head, to update the parameters of YOLOv8. This joint loss training paradigm imposes regular constraints on the expression of visual defect features, especially for blurry and indistinct defect samples, thereby avoiding false detections of samples that are prone to confusion in industrial defects.



(a) The mAP@0.5 of the four models in industrial quality inspection

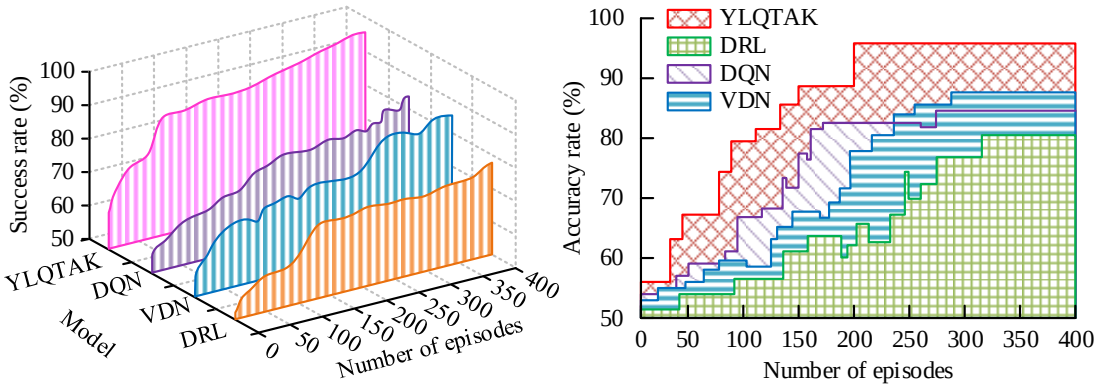
(b) The precision rates of the four models in industrial quality inspection

Fig. 6. mAP@0.5 and precision

4.3 Quality Inspection Intelligent Decision Performance and Convergence Evaluation

To verify the effectiveness of the LLaMA-enabled QMIX multi-agent decision model in the YLQTAK framework for collaborative decision-making in multi-station quality inspection, this study conducted multiple comparison experiments. The comparison models were Deep Q-Network (DQN), Value-Decomposition Network (VDN), and single-agent DRL. This study first compared the decision execution success rate and accuracy of the four models in industrial quality inspection, and the results are shown in Figure 7. The decision execution success rate represents the proportion of complete quality inspection task instances that can complete the entire process of multiple workstations without any illegal actions, without task termination or process deadlock. The decision accuracy rate indicates the proportion of samples for which the output actions of each detected sample are completely consistent with the true reference actions.

In Figure 7(a), the success rate of YLQTAK in the industrial quality inspection decision-making process became 95.10% within 400 episodes, which was significantly higher than the 82.35% achieved by DQN, 85.78% obtained by VDN, and 78.65% acquired by single-agent DRL. The reason for this performance is that YLQTAK used LLaMA semantic knowledge injection and the QMIX algorithm. In Figure 7(b), the decision execution accuracy of YLQTAK was consistently higher than the comparison models throughout, with a maximum accuracy of 96.35%. This was because single-agent DRL lacked multi-agent collaboration and semantic constraints, while DQN and VDN did not incorporate LLM semantic priors, resulting in weaker execution stability than YLQTAK in complex industrial quality inspection scenarios. Furthermore, in the process of setting the conditions and working environment, the semantic prior knowledge obtained through LLaMA encoding provided high-level defect-related guidance for the QMIX joint state space, which helped significantly compress the multi-agent exploration space and alleviate the sparse reward problem, enabling PLQTAK to achieve stable convergence within 400 episodes.

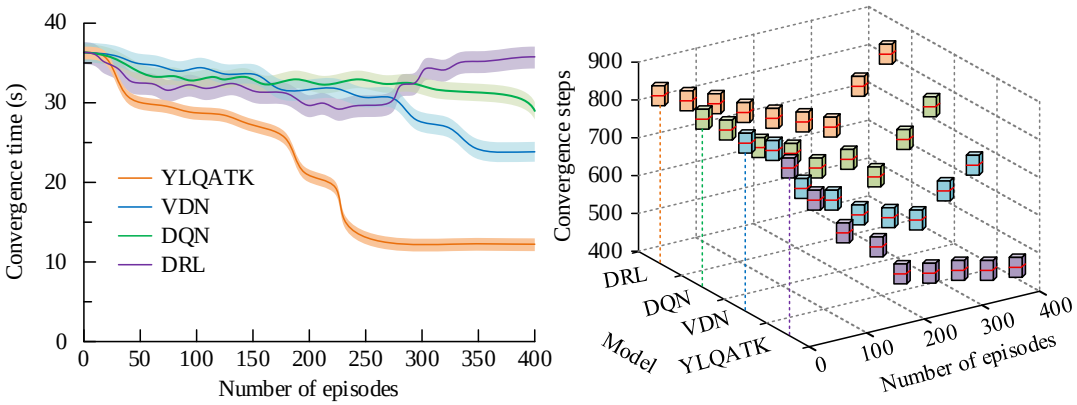


(a) The success rate of quality inspection decision implementation for the four models

(b) The accuracy rate of quality inspection decision execution for the four models

Fig. 7. Success rate and accuracy of decision execution

The results of station scheduling convergence time and steps for each model are shown in Figure 8. In Figure 8, the convergent steps and convergent time of YLQTAK in task scheduling both exhibited a declining pattern as iteration number increased and then became stable at about 200 episodes. After reaching 400 episodes, the convergent steps reduced to just 426 steps, while the convergent time dropped to only 12.3 s. On the other hand, DQN and VDN without LLaMA semantic prior convergent time of DQN and VDN took 28.3 s and 24.1 s respectively. Meanwhile, single-agent DRL had the highest convergence steps and time due to the lack of multi-agent collaboration mechanisms. These experimental results indicated that LLaMA semantic knowledge injection and the QMIX multi-agent value mixing mechanism significantly accelerated model convergence and better met the real-time requirements of industrial quality inspection.



(a) The convergence time for the scheduling of industrial quality inspection tasks in each model

(b) The convergence steps for the industrial quality inspection task scheduling of each model

Fig. 8. Comparison of convergence time and steps

4.4 Production Line Scheduling Optimization Efficiency Verification

To evaluate the scheduling optimization performance of the joint optimization mechanism of TCN, K-Means, and ACO in the YLQTAK framework under dynamic production line scenarios, this study conducted comparison experiments with Genetic Algorithm (GA), Particle Swarm Optimization (PSO), and Simulated Annealing (SA). To ensure fair comparison, the three meta-heuristic baselines were all configured under the same task scheduling instances and the same computing budget: the population size of GA and PSO was set to 50, the maximum number of evolutionary generations/iterations was set to 100, the crossover and mutation probabilities of GA were set to 0.8 and 0.05 respectively, the inertia weight of PSO was 0.7, the cognitive and social coefficients were set to 1.494. The initial temperature of SA was set to a cooling coefficient of 100 and 0.95 respectively, the termination temperature was $10\times^{-3}$, and the Markov chain length at each temperature was set to 50. All baselines were run on the same software and hardware platform.

The total task scheduling time (The total time consumed for completing an entire batch of multiple quality inspection tasks) and throughput of the four optimization algorithms are shown in Figure 9.

In Figure 9, the quality inspection task scheduling time of the YLQTAK framework, under ACO scheduling optimization and TCN temporal feature optimization, decreased from an initial 170.1 s to 23.6 s, a reduction of up to 86.13%. This indicated that the ACO algorithm with temporal operational priors enabled YLQTAK to complete path optimization more efficiently and significantly reduced scheduling time. Meanwhile, the throughput rate of YLQTAK after running for 400 episodes was 260 tasks, which is much more than the throughput rate of GA, which was 185 tasks, PSO, which had 210 tasks, and SA, which only managed 155 tasks. This was due to the fact that YLQTAK could successfully accomplish dynamic allocation of priority levels through K-Means defect classification.

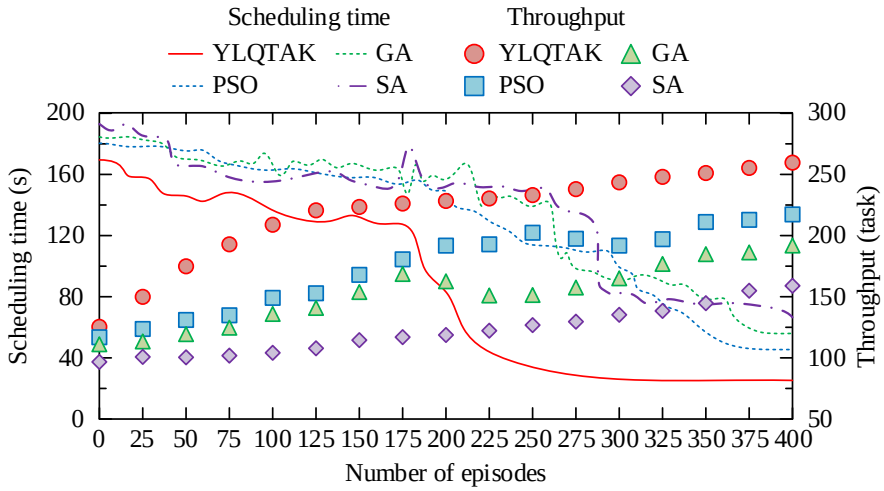
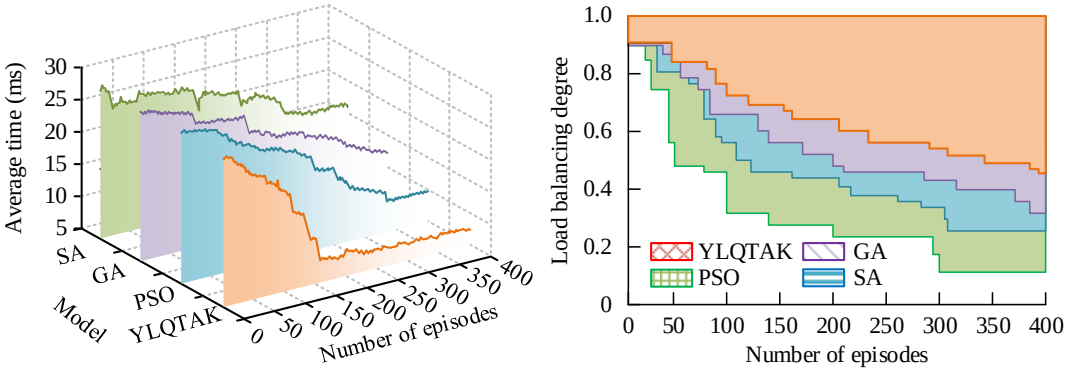


Fig. 9. Task scheduling time and throughput

The average time per task (Only the edge local computing delay of forward reasoning for a single sample is counted) and load balancing degree of different algorithms in industrial quality inspection are shown in Figure 10.

Note: The load balancing degree is defined as the normalized standard deviation of the workload among all quality inspection workstations, used to quantify the uniformity of task allocation scheduling. Its mathematical expression is: $LBD = \frac{1}{n} \cdot \sqrt{\frac{1}{N}\sum_{q=1}^{N}(n_q - \bar{n})^2}$; LBD represents the load balancing degree, N is the total number of quality inspection workstations, n_q represents the workload assigned to the q workstation, and $\bar{n}$ is the average workload of all workstations.



(a) The average time consumption for single-task quality inspection using different algorithms

(b) Load balancing degree of single-task quality inspection by different algorithms

Fig. 10. Time per task and load balancing degree

In Figure 10(a), the average time per task of YLQTAK remained stable at 7.8 ms across 400 episodes, lower than GA's 12.4 ms, PSO's 9.8 ms, and SA's 16.4 ms. This was because YLQTAK integrated TCN temporal features with the global optimization capability of ACO, enabling rapid matching of optimal scheduling strategies and significantly reducing single-task processing latency. In Figure 10(b), the load balancing degree of YLQTAK has been reduced dramatically from 0.89 to 0.11 after 400 episodes, with much better performance compared with the other three algorithms. This was mainly because of the combination of K-Means defect detection method and multiple agents' collaborative allocation algorithm that helped YLQTAK dynamically allocate workloads according to defect grades and workload distribution at each station. Furthermore, the 7.8 ms average single-task latency only represents the local computing latency at the edge, encompassing the YOLOv8 defect detection, TCN temporal feature extraction, K-Means clustering, ACO heuristic scheduling computations, and QMIX multi-agent decision-making inference running on the edge hardware, but excluding the LLaMA semantic reasoning overhead that is offloaded to the cloud.

4.5 Ablation Experiment

In order to quantitatively study the individual roles and synergies between the perception, semantics, decision-making, and optimization modules in the YLQTAK framework, a series of ablation experiments were conducted in this study. Through the process of eliminating each module one by one and examining its performance on critical metrics, the significance of each algorithmic element was verified for scientific accuracy and efficiency. The experimental results are shown in Table 2.

Table 2. Ablation experiment results

Model	Experimental indicators (%)				
	mAP@0.5	Accuracy 1 (%)	Accuracy 2 (%)	Reduction rate (%)	Average reward value
No YOLOv8 (Replaced with baseline CNN)	76.34	81.22	72.51	8.65	18.62
No LLaMA	96.14	94.63	84.12	22.31	29.74
No QMIX	97.80	94.71	71.36	21.89	17.53
No TCN-K-Means-ACO	97.76	82.35	85.67	9.24	30.12
YLQTAK	97.82	95.78	96.35	23.60	38.45

Note: Accuracy 1 and Accuracy 2 represent defect classification accuracy and quality inspection decision accuracy respectively. Reduction rate is the scheduling path length reduction rate.

As shown in Table 2, removing any key module from the YLQTAK framework caused varying degrees of decline in core metrics. For example, after removing the YOLOv8 perception module, the mAP@0.5 for defect detection in the framework dropped sharply from 97.82% to 76.34%, a decrease of 21.58%. Removing the LLaMA module means that the semantic alignment soft supervision loss during the training stage will be disabled. The mAP@0.5 drops to 96.14%, proving that LLaMA brings quantifiable positive constraints on the learning of visual features during the training stage. While QMIX and TCN-K-Means-ACO belong to the modules for decision-making and scheduling, they do not modify the parameters of the visual detection backbone network, thus having almost no impact on mAP@0.5. Their main performance gains are reflected in defect classification accuracy, quality inspection decision accuracy, workstation scheduling optimization, and cumulative rewards: After removing the QMIX decision layer, the multi-agent collaboration capability of the framework failed, resulting in a decision accuracy of only 71.36%, unable to form globally optimal decisions. Without joint optimization, the accuracy for defect classification lowered from 95.78% to 82.35%, and the percentage of improvement in scheduling route decreased from 23.60% to 9.24%. The findings suggested that lacking temporal features and scheduling precedence led to lower efficiency in inspection of manufacturing quality. The findings confirmed that the algorithm collaboration between perception, semantics, decision-making, and optimization was essential for enhancing efficiency and accuracy in industrial quality inspection.

5. Conclusions

In response to the problems existing in the current industrial quality inspection research, such as the separation of LLM and DRL, insufficient multi-agent collaborative decision-making ability, and the lack of a complete global optimization workflow, the research proposes a hierarchical YLQTAK framework, and constructs a complete technical link that includes visual perception, semantic knowledge reasoning, multi-agent collaborative decision-making, and joint optimization of workstation scheduling. Compared with the independent solution schemes of each algorithm in existing research, this framework effectively realizes cross-module feature-level information interaction through explicit feature projection and joint loss regularization training. Comprehensive experimental results show that the YLQTAK framework has achieved excellent overall performance when integrating industrial defect datasets. Thanks to semantic-assisted feature regularization and multi-agent value decomposition optimization, compared with the baseline method YLQTAK, it has better defect detection accuracy, defect classification ability, and

multi-workstation decision reliability. Moreover, the embedded TCN-K-Means-ACO optimization module effectively improves the scheduling efficiency and workstation load balance.

From the application perspective, the YLQTAK framework provides a feasible system-level idea for the intelligent quality inspection workflow of flexible manufacturing lines. Its modular design features enable it to have the potential to interface with mainstream industrial infrastructures such as SCADA and PLC. At the same time, through multi-station collaborative quality inspection and reasonable task redistribution, the YLQTAK framework helps reduce the frequency of human intervention and the risk of product scrapping. In actual deployment, it brings indirect benefits to the overall manufacturing cost control. However, due to the influence of large parameter quantity modules, the overall framework has a high computational cost and the inference speed and hardware adaptability may not fully meet the extreme real-time requirements of the production line. Moreover, it is difficult to directly complete the full model inference at resource-constrained industrial edge terminals.

Future research will focus on conducting targeted domain knowledge transfer for industrial quality inspection semantics, promoting the engineering implementation of multi-agent semantic-enhanced quality inspection systems:

- Transfer the defect reasoning, defect classification, and process matching capabilities of the LLaMA teacher model to a small-scale student model that can be deployed on the edge, retaining only the semantic representations related to the defect task and eliminating the redundant overhead of the general large-text model.
- Build a defect semantic vector quantization and offline caching library, enabling the vast majority of high-frequency defect categories to complete semantic queries locally on the device without calling the cloud-based large model.
- Explore lightweight feature alignment fine-tuning strategies, constraining the output feature space of the edge student model to align with the original LLaMA semantic embeddings, ensuring that the defect understanding ability of the model does not decrease after compression.

Funding Statement

The research is supported by: S&T Program of Baoding, Research on Industrial Quality Inspection Systems Based on Large Language Models and Multi-Agent Collaboration, No.: 2511ZG017.

References

[1] Reis AM, Dall-Orsoletta A, Nunes E, Costa L, Sousa S. Quality costs and Industry 4.0: inspection strategy modelling and reviewing. *Int J Adv Manuf Technol*. 2025;136(9):3883-3897. https://doi.org/10.1007/s00170-024-13184-9

[2] Ait El Cadi A, Gharbi A, Dhouib K, Artiba A. Joint production, maintenance, and quality control in manufacturing systems with imperfect inspection. *J Manuf Syst*. 2024;77:848-858. https://doi.org/10.1016/j.jmsy.2024.10.020

[3] Guo H, Bao Y, Hu S, Luan C, Fu J, Li L, et al. AFCN: An attention-directed feature-fusion ConvNeXt network for low-voltage apparatus assembly quality inspection. *IET Image Process*. 2024;18(8):2093-2104. https://doi.org/10.1049/ipr2.13085

[4] Yan Q, Ouyang H, Tao Z, Shen M, Du S, Zhang J, et al. Multi-wavelength optical information processing with deep reinforcement learning. *Light Sci Appl*. 2025;14(1):160-171. https://doi.org/10.1038/s41377-025-01846-6

[5] Wang Z, Xuan J. Intelligent fault recognition framework by using deep reinforcement learning with one dimension convolution and improved actor-critic algorithm. *Adv Eng Inform*. 2021;49:101315. https://doi.org/10.1016/j.aei.2021.101315

[6] Gu Z, Zhu B, Zhu G, Chen Y, Tang M, Wang J. AnomalyGPT: Detecting industrial anomalies using large vision-language models. In: *Proceedings of the AAAI Conference on Artificial Intelligence*; 2024 Feb 20-27; Vancouver, Canada. 2024;38(3):1932-1940. https://doi.org/10.1609/aaai.v38i3.27963

[7] Fan H, Liu X, Fuh JYH, Lu WF, Li B. Embodied intelligence in manufacturing: leveraging large language models for autonomous industrial robotics. *J Intell Manuf*. 2025;36(2):1141-1157. https://doi.org/10.1007/s10845-023-02294-y

[8] Liu PF, Lu Q, Zhao XW, Tao B. Joint knowledge graph and large language model for fault diagnosis and its application in aviation assembly. *IEEE Trans Ind Inform*. 2024;20(6):8160-8169. https://doi.org/10.1109/TII.2024.3366977
[9] Jiang X, Li J, Deng HQ, Liu Y, Gao BB, Zhou YF, et al. MMAD: A comprehensive benchmark for multimodal large language models in industrial anomaly detection. In: *International Conference on Learning Representations*; 2025 May 24-28; Singapore. 2025. https://doi.org/10.48550/arXiv.2410.09453
[10] Chen M, Kang Y, Li K, Li P, Zhao YB. Deep reinforcement learning for maintenance optimization of multi-component production systems considering quality and production plan. *Qual Eng*. 2025;37(2):219-230. https://doi.org/10.1080/08982112.2024.2373362
[11] Xing J, Ma Y, Cai J, Shi J, Liu J. Distributed scheduling method for smart shop floor based on QMIX. In: *2023 IEEE 19th International Conference on Automation Science and Engineering*; 2023 Aug 26-30; Auckland, New Zealand. IEEE; 2023. p. 1-6. https://doi.org/10.1109/CASE56687.2023.10260396
[12] Zhang J, Lv Y, Li Y, Liu J. An improved QMIX-based AGV scheduling approach for material handling towards intelligent manufacturing. In: *2022 IEEE 20th International Conference on Embedded and Ubiquitous Computing*; 2022 Oct 27-29; Bournemouth, UK. IEEE; 2022. p. 54-59. https://doi.org/10.1109/EUC57774.2022.00018
[13] Wan L, Fu L, Li C, Li K. An effective multi-agent-based graph reinforcement learning method for solving flexible job shop scheduling problem. *Eng Appl Artif Intell*. 2025;139:109557. https://doi.org/10.1016/j.engappai.2024.109557
[14] Elsharkawy ZF, Kasban H, Abbass MY. Efficient surface crack segmentation for industrial and civil applications based on an enhanced YOLOv8 model. *J Big Data*. 2025;12(1):16-35. https://doi.org/10.1186/s40537-025-01065-1
[15] Touvron H, Lavallee M, Izacard G, Martinet X, Lachaux MA, Lacroix T, et al. LLaMA: Open and efficient foundation language models. arXiv preprint arXiv:2302.13971; 2023. https://doi.org/10.48550/arXiv.2302.13971
[16] Radford A, Kim JW, Hallacy C, Ramesh A, Goh G, Agarwal S, et al. Learning transferable visual models from natural language supervision. In: *Proceedings of the 38th International Conference on Machine Learning*; 2021 Jul 18-24; Virtual. PMLR; 2021. p. 8748-8763. https://doi.org/10.48550/arXiv.2103.00020
[17] Rashid TS, Samvelyan M, Witt CS, Farquhar G, Foerster J, Whiteson S. QMIX: Monotonic value function factorisation for deep multi-agent reinforcement learning. In: *Proceedings of the 35th International Conference on Machine Learning*; 2018 Jul 10-15; Stockholm, Sweden. PMLR; 2018. p. 4295-4304. https://doi.org/10.24963/ijcai.2019/661
[18] Zhang M, Tong W, Zhu G, Xu X, Wu EQ. SQIX: QMIX algorithm activated by general softmax operator for cooperative multi-agent reinforcement learning. *IEEE Trans Syst Man Cybern Syst*. 2024;54(11):6550-6560. https://doi.org/10.1109/TSMC.2024.3370186
[19] Al-qaness MAA, Dahou A, Trouba NT, Abd Elaziz M, Helmi AM. TCN-inception: temporal convolutional network and inception modules for sensor-based human activity recognition. *Future Gener Comput Syst*. 2024;160:375-388. https://doi.org/10.1016/j.future.2024.06.016
[20] Suyal M, Sharma S. A review on analysis of k-means clustering machine learning algorithm based on unsupervised learning. *J Artif Intell Syst*. 2024;6(1):85-95. https://doi.org/10.33969/AIS.2024060106
[21] Awadallah MA, Makhadmeh SN, Al-Betar MA, Dalbah LM, Al-Redhaei A, Kouka S, et al. Multi-objective ant colony optimization: review. *Arch Comput Methods Eng*. 2025;32(2):995-1037. https://doi.org/10.1007/s11831-024-10178-4