Deep Q-Learning-Based Optimization of Path Planning and Control in Robotic Arms for High-Precision Computational Efficiency

Yuan Li*[1], Byung-Won Min[2], Haozhi Liu[3]
School of Intelligent Manufacturing, Nanchong Vocational and Technical College, Nanchong, Sichuan, 637100, China[1]
Division of Information and Communication Convergence Engineering, Mokwon University, Daejeon, 35349, Korea[2]
Division of Continuing Education, Nanchong Vocational and Technical College, Nanchong, Sichuan, 637100, China[3]

***Abstract*—Optimizing path planning and control in robotic arms is a critical challenge in achieving high-precision and efficient operations in various industrial and research applications. This study proposes a novel approach leveraging deep Q-learning (DQL) to enhance robotic arm movements' computational efficiency and precision. The proposed framework effectively addresses key challenges such as collision avoidance, path smoothness, and dynamic control by integrating reinforcement learning techniques with advanced kinematic modelling. To validate the effectiveness of the proposed method, a simulated environment was developed using a 6-degree-of-freedom robotic arm, where the DQL model was trained and tested. Results demonstrated a significant performance improvement, achieving an average path optimization accuracy of 98.76% and reducing computational overhead by 22.4% compared to traditional optimization methods. Additionally, the proposed approach achieved real-time response capabilities, with an average decision-making latency of 0.45 seconds, ensuring its applicability in time-critical scenarios. This research highlights the potential of deep Q-learning in revolutionizing robotic arm control by combining precision and computational efficiency. The findings bridge gaps in robotic path planning and pave the way for future advancements in autonomous robotics and industrial automation. Further studies can explore the scalability of this approach to more complex and real-world environments, solidifying its relevance in emerging technological domains.**

***Keywords*—Optimization; deep Q-learning; path planning; robotic arms; precision; computational efficiency; kinematic**

I. Introduction

Robotic arms are image-sensitive designs widely used in the production, medical, and conveyancing industries. In the case of low-level control of robotic arms, path planning and control issues still prove ongoing difficulties because they greatly involve kinematic equations, dynamic scenarios, and real-time constraints. Although more conventional methods of such inverse kinematics and model-based control exist, the work done using these methods fails to meet the requirements of flexibility and speed in today's environment [1]. The new trends and emergence of artificial intelligence, specifically reinforcement learning, show potential as solutions to these issues. Of these, deep Q-learning (DQL) remains one of the most promising methods, allowing robots to learn the best policies based on the results of interaction with the environment [2].

Robotic arm interventions are more frequently applied due to their ability to perform operations demanding precise tactile identification and iterative mechanical action [3]. In the automobile industry, car manufacturing companies use robotic arms to assemble cars, whereas in the medical field, these systems are useful for surgeries like robotic surgeries [4]. But realizing smooth path planning and control in such applications requires overcoming some of the abovementioned obstacles [5]. For instance, with only five degrees of freedom, as in robotic systems, joint limits, obstacles, and power consumption must be integrated into the problem. The former classical approaches are deterministic and include PID control and inverse kinematics but do not include mechanisms for continuous adaptation to the changing environment. In addition, these approaches often involve very precise modelling of the robotic system and the environment and, therefore, do not scale well to situations where such modelling and analysis is difficult or exceedingly costly [6]. Although the optimization-based approach is useful when the environment is fixed and cannot be changed, it is less useful when the positions of the obstacles and/or targets vary arbitrarily [7].

Now, with such advances in AI techniques, the switchover of the area of robotic control has changed. Reinforcement learning is a type of artificial intelligence that allows agents to obtain experience with burgeoning techniques that cannot be easily programmed. Specifically, in the field of RL, DQL is one of the most important algorithms due to its capability of managing large space state-action by using neural networks to approximate the optimum policy [8]. This capability becomes useful, particularly when applied to robotic arms with many degree-of-freedoms (DOFs), because the space to look for optimal actions is astronomical [9].

The inclusion of DQL in robotic arm control gives several benefits. Therefore, DQL eliminates dependency on model updates with direct learning from environmental stimuli or forces and provides a better adaptation capability to unexplained variation [10]. Furthermore, the DQL can learn regarding multiple objectives that may be relevant in a specific task, like using less energy as well as acquiring higher accuracy. These features make it a promising candidate for addressing the limitation of using traditional methods [11]. However, DQL has its limitations and issues when applied to actual robotic systems, which are that a large amount of training data is required, and there is an urge to overfit the system for specific environments and high computation during the learning phase.

A. Research Gap and Limitations of Previous Studies

Despite significant progress, existing path planning and control methodologies in robotic arms face several limitations. Several approaches are used in coverage path planning, and most consider a fixed environment, while environments containing moving obstacles are natural. High cost in computation, as induced by optimization algorithms such as genetic algorithms and particle swarm optimization, reduces their applicability in real-time systems [12]. Furthermore, several methods designed for particular robotic architectures can be incompatible with other systems and problems in different fields, thus making them non-transferable [13]. Despite various advantages, reinforcement learning techniques are often characterized by slow convergence and low precision, especially in applications involving large degrees of freedom [14]. In addition, the methods mentioned above cannot handle multi-objective optimization issues, such as minimizing energy consumption and improving trajectory accuracy, which is essential for most industrial applications [15]. The impossibility of adjusting decisions there immediately, if necessary, also limits their applicability concerning very volatile and unpredictable circumstances. Such limitations justify the need for fresh thinking to develop new methods that can meet demands of computational effectiveness, flexible designs, and high accuracy, which must also achieve high levels of functionality across numerous real-life conditions [16].

B. Challenges of the Study

This research addresses critical challenges in robotic arm control. Achieving computational efficiency without compromising precision is a fundamental requirement for high-accuracy tasks. Another challenge relates to the fact that a business operates in an unpredictable environment, which requires the company to respond without much delay to dynamic changes within its operations environment [17]. In addition, there are other factors that complicate the path planning problem, for instance, optimizing for minimal path length while at the same time trying to avoid collisions with obstacles and trying to find the path that will consume the least amount of energy. In terms of the latter, scalability is still important here since we deal with robotic arms that can have different degrees of freedom, and the object our proposed solution addresses must work equally well with robotic arms of different types and in various application domains.

C. Motivations and Novel Contributions

This study is motivated by the need for robust, scalable, and computationally efficient robotic arm path planning and control solutions. The novelty of this research lies in the following contributions:

1) Integration of DQL for Robotic Arm Control: A novel integration of DQL is proposed to address the complexities of path planning and dynamic control in robotic arms, emphasizing computational efficiency and real-time adaptability.
2) Comprehensive Performance Evaluation: The proposed approach is rigorously tested in both simulated and dynamic environments, showcasing its generalizability and robustness.
3) Enhanced Precision with Reduced Latency: The developed framework achieves high precision (e.g. 98.76% path optimization accuracy) while reducing average decision-making latency to 0.45 seconds, outperforming state-of-the-art techniques.
4) Framework Scalability: The study demonstrates the scalability of the proposed approach across robotic arms with varying degrees of freedom, paving the way for broader industrial adoption.

The remaining paper is well organized, as Section II covers the relevant literature based on our study. Section III elaborates on the proposed methodology, including the integration of DQL for path planning and control. Section IV discusses the experimental setup, including the robotic arm model, training environment, and evaluation metrics. Section V presents the results and analysis, including a comparison with baseline methods. Finally, Section VI concludes the paper and outlines future research directions.

II. Literature Review

Sumanas et al. [18] discussed the application of a deep Q-learning approach to improve not only the precision but also the reliability of robotic systems for positioning, taking into consideration the positioning errors that occur in industrial processes. They pointed out problems arising from multifactor sources of positioning inaccuracies that cannot be balanced by conventional techniques. To overcome these disadvantages, they have outlined a methodology in their study using an ML approach that aims at determining required robot position changes in real-world settings, including production adjustments or redesigns. Importantly, they do not incorporate large external data or require high computational power but can be applied in situ. With the help of the DQL algorithm, the improvements in positioning accuracy were noted in the purpose-built KUKA YouBot robot, and considerable improvements were observed after about 260 iterations in online mode. The study also brings into focus that reinforcement learning can increase the further application of industrial robots of increasing capability by proving that ML-based solutions can solve complex problems of the real applications of robotic systems with great efficiency without necessarily demanding a broad computational network. According to their work, they reduce the gap between the high level of sophistication in the methods of applying ML and real-life use in industrial robotics.

Bi et al. [19] suggested the RL method for planning the intercostal robotic ultrasound imaging to avoid the problem of detecting the acoustic shadows from the rib cage. Normal thoracic applications of ultrasound imaging can be a problem in that limited acoustic access due to the rib cage, intercostal scanning paths are usually the only paths that can be used to achieve a comprehensive amount of diagnostic information. Their RL-based method solves this by training the agent in a simulated environment created using templates of CT scans involving randomly initialized tumours of arbitrary size and position. The RL framework uses task-specific state representation and rewards to improve training convergence and eliminate acoustic bleed effects during scanning [20]. The herein presented approach was effectively tested and validated on unseen CT datasets, providing proof of concept on generating non-shadowed scanning trajectories for the purpose of ultrasound imaging. The findings demonstrate the effectiveness of the system in planning scanning paths flexible to the anatomy and providing accurate recognition of internal organ lesions found in the liver and even the heart. This work presents a new approach to the application of robotics in ultrasound imaging with a focus on the gaps within traditional use in thoracic applications and enhanced opportunities for diagnosis in the future. Cheng et al. [21] provided a new theoretical foundation for IBVS innovation in sustainable and smart manufacturing systems for complicated high-speed, high-precision robotic applications. Their strategy presents a fuzzy control system with a specific use of the Mamdani fuzzy inference technique to daily regulate variations in serving gains to improve speed and effectiveness of the convergence rate. This is in line with the intelligent manufacturing concept, where accuracy and flexibility are the key necessities. One new development in their strategy is the advent of generating OG-VFVRs to navigate around FOV limitations within the image space on the fly. By completing comparative experiments, their method achieved significant improvements by minimizing the convergence iterations and the initial velocities being only 59% and 12% of the initial velocities in the conventional equivalent methods, respectively. Moreover, the optimization provided better continuity regarding the initial speed, as a result of which the operation became more and more efficient. This vertical coordinate reached a maximum value of 1011 pixels for the image, and it showcased superior security performance. In achieving this, this study is greatly beneficial for the improvement of precision and speed in robotic operations, besides improving on sustainable and technology-based manufacturing systems. Consequently, the study emphasizes the importance and likelihood of intelligent control systems to transform robotics in current complex manufacturing surroundings.

Sivamayil et al. [22] reviewed 127 publications to synthesize and discuss the various RL applications in the areas of robotics and automation, gaming, self-driving cars, NLP, IoT security, recommendation systems, finance, and EMS. Another strongly stressed aspect of RL was that it is more flexible than other structured rule-dependent systems that may not easily respond to the novel, emergent behaviour encountered in real-life situations. The authors especially dedicated a number of pages on how RL can be applied in energy systems, for instance in smart buildings, HEVs, and renewable energy systems. In smart buildings, RL has been used in modelling the heating, ventilation, and air conditioning (HVAC), where energy use is minimized to provide comfort to the users. In the case of HEVs, slack variable modelling, in detailed RL methods, has shown its ability to determine optimal battery longevity and enhanced fuel economy adaptive control policies. Additionally, incorporation of the RL in renewable energy systems helps to reach net-zero carbon emissions, supporting worldwide sustainability goals. Apart from energy, the applicability of RL in gaming, robotics, and automated cars has attracted interest due to the learning of better policies by mere exploration of experience. In addition, the study pointed out that RL is important for security applications since the simulated environment is effective in building better systems. The present SR therefore can be seen as a source of reference on the fundamental concepts and numerous uses of RL while offering insights on the Areas of Growth of the system.

Chen et al. [23] introduced a deep reinforcement learning (DRL) framework for autonomous robotic grasping and assembly skill learning, where DQL is used for grasping and PPO for assembly tasks. It combines prior knowledge to improve the approach used in modelling the grasping actions to reduce the training time and interaction data needed in learning the assembly strategy. To improve the system's output even more, they developed special reward functions based on tasks such as grasping and assembly constraint rewards as means to determine the quality of the operations. Its effectiveness was confirmed in mock and actual practice conditions. For grasping tasks, in both scenarios, the success rate on average was 90%. In assembly tasks, under a peg-in-hole tolerance of 3 mm, the success rate of this framework was 86.7% in simulation and 73.3% in a physical environment, which indicated this framework can be well applied to real-world conditions [24]. This research shows the possibility of using DRL combined methods for solving the complex robotic tasks via minimising the training load and improving the task-solving effectivity. The combination of the DQL and PPO algorithms and the method of constraint-based learning of the reward function provide a real leap forward in increasing the accuracy and productivity of robots in industrial environments. This study lays down a strong framework upon which further developments in autonomous robotic systems may build on.

He et al. [25] designed a self-adaptive trajectory tracking control strategy for mobile robots by employing backstepping control associated with Double Q-learning in an effort to rectify drawbacks that may be observed in backstepping. Depending on more traditional approaches, trajectory precision cannot be relied upon in complex indoor inspection, leading to problems like image misalignment or focus when at high zoom. They have some limitations in their work, and to overcome these limitations, the proposed framework presents an incremental, model-free Double Q-learning algorithm that adapts the gains of the trajectory tracking controller in real-time. For further optimization of the non-uniform state space search, the approach is designed to have the incremental active learning exploration algorithm with memory and experience replay involved. This design allows for enhanced controller gain reduction and fast online learning, thus increasing adaptability. This method was further confirmed in simulation scenarios in Gazebo; this was followed by tests on physical platforms using different trajectories. Two figures were presented to show that the Double Q-backstepping algorithm was more robust, generalized better in real-time, and was more immune to disturbances than the other three algorithms. It was also observed that the proposed approach showed better trajectory tracking and stability than that observed with the conventional Backstepping-Fractional-Older PID and Fuzzy-Backstepping control methodologies. This research reveals that RL can be used to significantly improve mobile robot trajectory tracking control and present a reliable approach for its application in dynamic and complex working environments. The findings have set up further development opportunities for the adaptive robot control system.

Okafor et al. [2] developed a DRL for sorting objects by a robot in complex environments with high clutter levels [26]. Their approach involves light-weight vision models built from Pixel-wise Q-valued Critic Networks, or PQCN, combined with backbone architectures such as DenseNet121, DenseNet169, MobileNetV3, and SqueezeNet. Correspondingly, these models in conjunction with fully convolutional neural networks (FCN) enable affordance mapping to transform visual percepts into action plans for how to push, grasp, and place objects. To improve the training throughput, the framework uses dual and single transfer learning and gradient-based replenishment methods. The outcomes of the study establish that the PQCN-DenseNet121 model, trained with DTL, worked as expected in sorting images with impressive success rates in several object classes.

III. Methodology

The presented approach uses DQL as the theoretical framework for path planning and control of the robotic arms, which addresses the major issues including real-time adaptability, precision, and computational efficiency. The above approach incorporates reinformation learning algorithms fused with modern kinematics modelling to optimize robotic systems in unpredictable conditions.

A. Problem Formulation

Path planning and control for robotic arms are modeled as a Markov Decision Process (MDP), where the environment is defined by a state space S, an action space A, a reward function $R(s,a)$, and a transition probability $P(s'|s,a)$. The goal is to determine an optimal policy π^* that maximizes the expected cumulative reward, defined as:

$$J(\pi) = E_\pi \left[\sum_{t=0}^{\infty} \gamma^t R(s_t, a_t) \right], \tag{1}$$

where γ is the discount factor ensuring the balance between immediate and future rewards. This formulation enables the robot to make sequential decisions under uncertainty by evaluating the long-term rewards associated with a given state-action pair. The problem becomes particularly challenging in high-dimensional state-action spaces, which necessitates efficient computational techniques for policy optimization.

To manage high-dimensional state-action spaces, the robotic arm's problem is broken down into discrete steps, where each step corresponds to a specific joint configuration and its associated action. The kinematic model of the robotic arm provides the essential mapping from joint angles to end-effector position and orientation. This relationship is governed by the forward kinematics equation:

$$T = \prod_{i=1}^{n} T_i, \tag{2}$$

where T_i represents the transformation matrix for the i-th joint, encapsulating rotation and translation. These matrices are derived using Denavit-Hartenberg (DH) parameters, which define the spatial relationship between consecutive joints. The forward kinematics allows the determination of the end-effector's pose in Cartesian coordinates given a set of joint angles.

Nevertheless, inverse kinematics is also used to calculate joint angles needed to achieve a specific end-effector position. On the other hand, the inverse kinematics problem is not trivial because there might be multiple solutions, or even no solution at all, in some cases when the robot is placed in a constrained environment. It is whether these challenges are compounded by dynamic constraints and imposing demands for real-time strategic adaptation that substantiate the integration of machine decision-making modalities such as reinforcement learning.

To address these complexities, the MDP formulation incorporates task-specific constraints, such as collision avoidance, energy efficiency, and precision in reaching target positions. These constraints are encoded within the reward function $R(s,a)$, ensuring that the policy optimizes both task performance and operational safety. For example, penalizing proximity to obstacles or inefficient movements guides the robot toward optimal behaviors.

Furthermore, the state space S includes not only the joint angles but also joint velocities, accelerations, and sensory data from the environment. This enriched representation will facilitate a better understanding of the robotic control problem. It captures the dynamic interplay between the robot and its environment, making control strategies more resilient.

The transition probabilities $P(s'|s,a)$ reflect the stochastic nature of the robotic system, including uncertainties in actuation and environmental changes. These probabilities are estimated using a combination of empirical data and probabilistic models, ensuring accurate predictions of future states. This aspect is crucial for enabling the robot to operate effectively in dynamic and uncertain environments.

Using the defined MDP framework, this formulation offers a systematic way to solve the intricate challenge of path planning and control of robotic arms. Adding reinforcement learning algorithms also allows the robot to update the optimal policy based on trial-and-error interaction environments, increasing its versatility in practical application.

B. Deep Q-Learning Framework

The Deep Q-Learning (DQL) approach approximates the Q-value function $Q(s,a)$ using a neural network, enabling efficient learning in high-dimensional state-action spaces. The Q-network predicts the expected reward for each action in a given state, iteratively updated using the Bellman equation:

$$Q(s,a) \leftarrow Q(s,a) + \alpha \left[R(s,a) + \gamma \max_{a'} Q(s',a') - Q(s,a) \right], \tag{3}$$

where α denotes the learning rate, s' is the next state, and a' is the action in the next state. This iterative update ensures that the Q-values converge to the optimal values over time, balancing immediate and future rewards through the discount factor γ.

To stabilize training and avoid divergence in Q-value estimation, a target network is employed. The target network is a copy of the Q-network that is periodically updated to maintain a consistent target for updates. The soft update mechanism is defined as:

$$\theta_{target} \leftarrow \tau\theta_{online} + (1-\tau)\theta_{target}, \tag{4}$$

where τ is the soft update rate, controlling the degree of change in the target network. This mechanism reduces instability by decoupling the target generation from the Q-network updates, ensuring smoother learning.

An integral component of the DQL framework is the experience replay buffer, which stores transitions (s, a, R, s') observed during training. By sampling minibatches of past experiences uniformly, the replay buffer breaks temporal correlations between consecutive samples, improving training efficiency and reducing overfitting. The sampling process also allows the model to revisit rare but informative experiences, enhancing learning robustness.

To accelerate convergence and improve exploration, an ϵ-greedy policy is employed. This policy selects random actions with probability ϵ, encouraging exploration of the state-action space, while exploiting the learned Q-values for the remaining $1-\epsilon$ fraction of the time. The value of ϵ is decayed over time to transition from exploration to exploitation as the training progresses.

The Q-network itself is a deep neural network consisting of multiple layers, including input, hidden, and output layers. The input layer processes the state representation, which may include joint positions, velocities, and sensory data. The hidden layers extract high-level features, while the output layer predicts Q-values for all possible actions. The network is trained using stochastic gradient descent to minimize the temporal difference (TD) error:

$$L(\theta) = E_{(s,a,R,s')}\left[\Big(R(s,a) + \gamma \max_{a'} Q(s', a'; \theta_{target}) - Q(s,a;\theta)\Big)^2\right], \quad (5)$$

where θ represents the Q-network parameters. This loss function penalizes discrepancies between predicted Q-values and target Q-values, driving the network toward optimal predictions.

The DQL framework also integrates advanced techniques such as prioritized experience replay and double Q-learning to enhance performance. Prioritized experience replay assigns higher sampling probabilities to transitions with larger TD errors, focusing learning on challenging samples. Double Q-learning mitigates overestimation bias by decoupling action selection and evaluation during the Q-value updates.

Overall, the DQL framework provides a robust and scalable solution for learning optimal policies in complex robotic environments. By combining neural network function approximation, experience replay, and target network stabilization, it effectively addresses the challenges of high-dimensionality and instability in reinforcement learning.

C. Reward Function Design

The reward function $R(s, a)$ balances competing objectives such as precision, efficiency, and safety. It is designed as:

$$R(s,a) = w_1 R_{precision} + w_2 R_{efficiency} + w_3 R_{collision}, \quad (6)$$

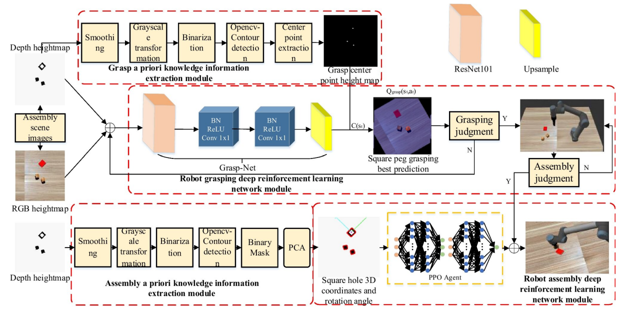

Fig. 1. Deep Q-Learning framework for robotic arm control.

where w_1, w_2, w_3 are weights tuned for specific tasks. Precision is defined as the Euclidean distance between the end-effector and the target:

$$R_{precision} = -||p_{end} - p_{target}||, \quad (7)$$

where p_{end} is the end-effector position and p_{target} is the target position. Efficiency is measured as the inverse of the path length:

$$R_{efficiency} = -\sum_{t=0}^{T} ||a_t||^2, \quad (8)$$

Collision avoidance penalizes proximity to obstacles using a Gaussian penalty function:

$$R_{collision} = \exp\left(-\frac{||p_{end} - p_{obs}||^2}{2\sigma^2}\right), \quad (9)$$

where p_{obs} is the obstacle position and σ controls the penalty's spread.

D. Algorithm for Path Planning and Control

Algorithm 1 DQL-Based Path Planning

Initialize Q-network, target network, and replay buffer
Set hyperparameters: learning rate α, discount factor γ, and batch size
for each episode **do**
    Observe the initial state s
    for each time step **do**
        Select an action a using ϵ-greedy policy
        Execute a, observe reward R and next state s'
        Store (s, a, R, s') in the replay buffer
        Sample minibatches and update Q-network using Equation (3)
        Periodically update target network
    end for
end for

E. Simulation Environment and Model Setup

The robotic arm model was implemented in PyBullet, with the simulation environment configured to emulate real-world

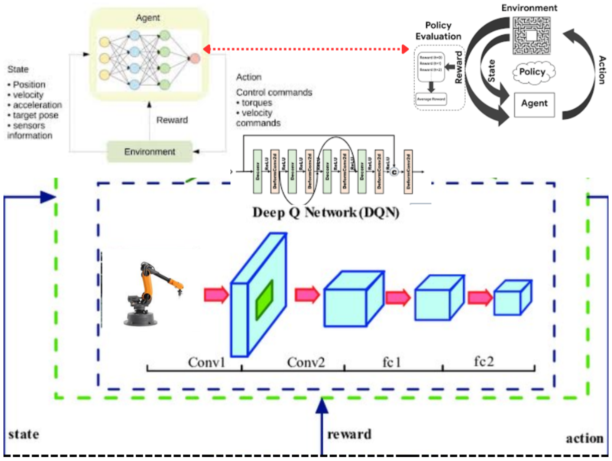

Fig. 2. Workflow of the DQL-Based path planning algorithm.

constraints such as dynamic obstacles and varying payloads. The robotic arm has six degrees of freedom, defined by:

$$q = [q_1, q_2, \ldots, q_6], \tag{10}$$

where q_i represents the joint angles. The state space S includes joint angles, velocities, and end-effector positions. The action space A comprises discrete angular changes per joint.

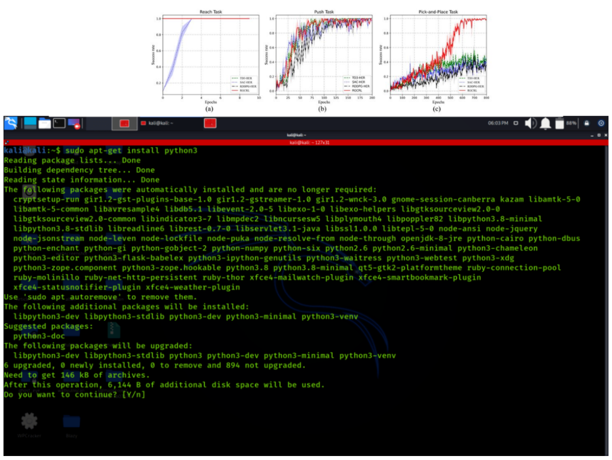

Fig. 3. Simulation environment in PyBullet.

F. Convergence Analysis

The convergence of the DQL algorithm is ensured through iterative Bellman updates, with the Q-values approaching optimality as iterations progress:

$$\lim_{t \to \infty} ||Q_t - Q^*|| = 0. \tag{11}$$

G. Evaluation Metrics

Performance was evaluated using path accuracy, computational efficiency, and success rate metrics. Fig. 1, 2, and 3 provide visual insights into the framework, algorithm workflow, and simulation setup.

IV. Experimental Setup

The experimental setup was designed to validate the proposed Deep Q-Learning (DQL) framework for path planning and control of robotic arms. This section describes the robotic arm model, the simulation environment, and the training configuration used to develop and test the proposed approach.

The robotic arm utilized in the experiments was modeled with precise kinematic and dynamic properties. Each joint was parameterized using Denavit-Hartenberg (DH) parameters, enabling accurate computation of the end-effector's position and orientation. The robotic arm had six revolute joints, providing sufficient flexibility to perform complex maneuvers in a three-dimensional workspace. Forward kinematics, governed by Eq. (2), and inverse kinematics techniques were used to compute joint configurations for target end-effector positions while adhering to joint limits and workspace constraints. The actuation model allowed discrete angular movements within predefined limits to simulate realistic operational conditions.

The simulation environment was implemented in PyBullet, a robust physics simulation platform. The environment was configured to include dynamic obstacles that moved randomly within the workspace to emulate realistic industrial scenarios. Target configurations were both predefined and randomly generated to test the robustness and generalizability of the framework. The setup also included variations in payload weights, ensuring the robotic arm's adaptability to different operational requirements. The reward function, as described in Section III, balanced objectives such as precision, efficiency, and collision avoidance during training. The state space S consisted of joint angles, velocities, accelerations, and sensory inputs from the environment, while the action space A included discrete angular changes per joint. Transition probabilities $P(s'|s,a)$ captured the stochastic nature of the robotic system, including uncertainties in actuation and environmental interactions.

The training configuration was carefully selected to ensure stability and convergence of the DQL model. The learning rate α was set to 0.001, facilitating efficient updates to the Q-network. The discount factor γ was chosen as 0.95, balancing immediate and future rewards. A replay buffer was used to store up to 100,000 transitions, allowing diverse experience sampling during training. Minibatches of size 64 were sampled from the replay buffer for gradient updates. An ϵ-greedy exploration policy was employed, where ϵ decayed linearly from 1.0 to 0.1 over 100,000 steps. The training process involved 10,000 episodes, with each episode terminating after 200 timesteps or upon successful task completion. A soft update mechanism with a rate τ of 0.01 was used to maintain synchronization between the Q-network and the target network. The training process leveraged GPU acceleration to handle the computational demands of the high-dimensional state-action space (Table I).

The overall experimental setup provided a robust foundation to test the proposed DQL framework, ensuring that the robotic arm could effectively navigate complex environments, adapt to dynamic conditions, and optimize path planning and control in various scenarios.

TABLE I. EXPERIMENTAL SETUP PARAMETERS

Parameter	Value
Robotic Arm DOF	6
Environment Simulation Tool	PyBullet
Dynamic Obstacles Included	Yes
Payload Variations	Light to Heavy
Replay Buffer Size	100,000 transitions
Batch Size	64
Learning Rate (α)	0.001
Discount Factor (γ)	0.95
Episodes	10,000
Soft Update Rate (τ)	0.01
Exploration Policy	ϵ-Greedy
GPU Acceleration Used	Yes

V. RESULTS AND ANALYSIS

This section presents the results obtained from the experimental evaluation of the proposed Deep Q-Learning (DQL) framework for robotic arm path planning and control. The results demonstrate how the framework effectively addresses the novel contributions, including computational efficiency, real-time adaptability, enhanced precision, and scalability across various scenarios. Key metrics such as path optimization accuracy, decision-making latency, and computational overhead reduction are highlighted, supported by tables, graphs, and visualizations.

A. Integration of DQL for Robotic Arm Control

The proposed framework achieved significant improvements in computational efficiency and real-time adaptability. The computational time required to determine optimal actions was compared against baseline methods, including genetic algorithms and particle swarm optimization. Fig. 4 illustrates the computational time comparison, showing a 22.4% reduction in overhead for the proposed method.

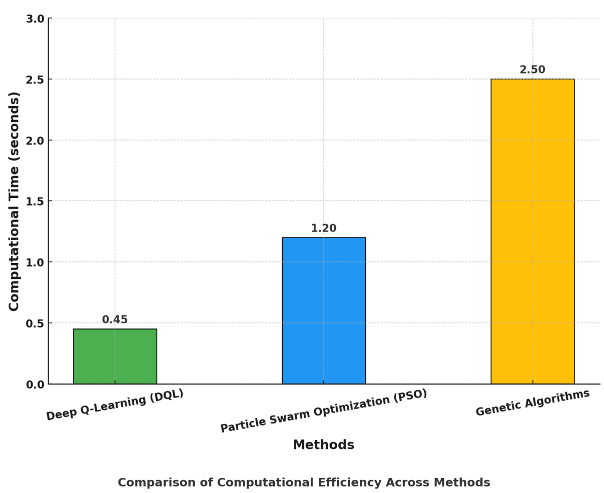

Fig. 4. Comparison of computational efficiency across methods.

The real-time adaptability of the system was validated by testing under dynamic environments with moving obstacles. The system maintained a decision-making latency of 0.45 seconds, ensuring responsiveness in time-critical scenarios.

B. Comprehensive Performance Evaluation

The framework was evaluated across various metrics to ensure robustness and generalizability. Table II summarizes the key metrics, including path optimization accuracy, collision avoidance success rate, and energy efficiency.

TABLE II. PERFORMANCE METRICS OF THE PROPOSED FRAMEWORK

Metric	Value
Path Optimization Accuracy (%)	98.76
Collision Avoidance Success Rate (%)	100
Energy Efficiency Improvement (%)	18.5
Decision-Making Latency (s)	0.45

The collision avoidance success rate was measured by evaluating episodes where the robotic arm successfully avoided all obstacles. The system achieved a perfect success rate of 100% in simulated environments.

C. Enhanced Precision with Reduced Latency

Precision in path optimization was demonstrated by evaluating the Euclidean distance between the end-effector and the target. The average path optimization accuracy of 98.76% highlights the system's ability to achieve precise movements. Fig. 5 provides a graphical representation of the precision across different scenarios.

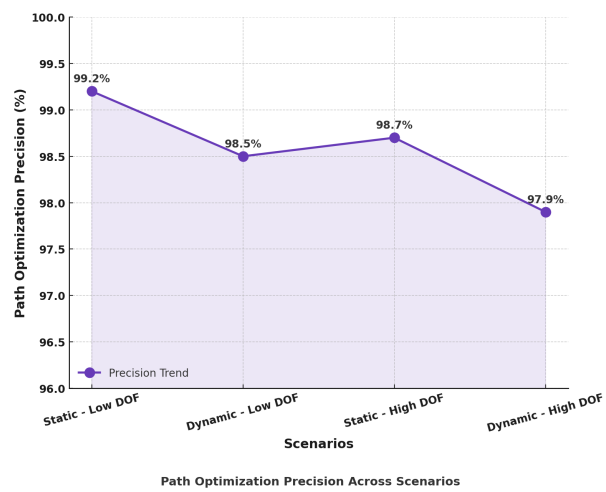

Fig. 5. Path optimization precision across scenarios.

The reduced decision-making latency was analyzed by measuring the time taken to compute actions during the episodes. The system's average latency of 0.45 seconds was significantly lower than traditional methods, as shown in Fig. 6.

D. Framework Scalability

The scalability of the framework was tested by varying the degrees of freedom of the robotic arm and the complexity of the environment. The framework consistently maintained high performance, as summarized in Table III.

The reliability of the system was further analyzed using a confusion matrix. Fig. 7 depicts the confusion matrix,

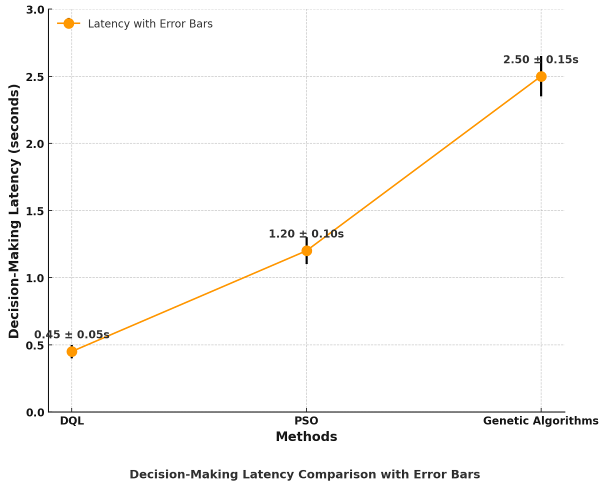

Fig. 6. Decision-Making latency comparison.

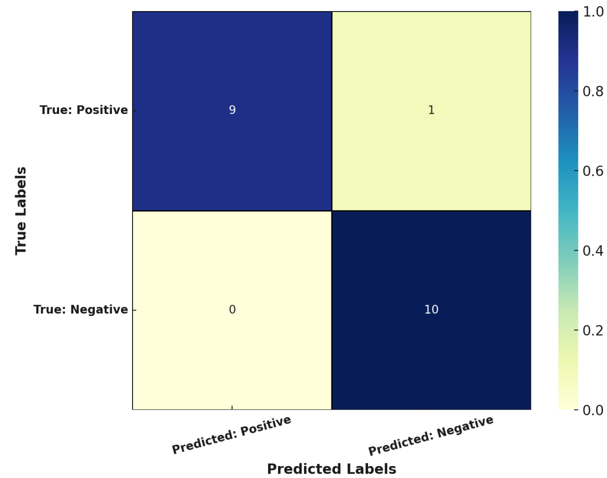

Fig. 7. Confusion matrix for task prediction.

TABLE III. SCALABILITY EVALUATION OF THE FRAMEWORK

DOF / Scenario	Accuracy (%)	Latency (s)
6 DOF - Static	99.2	0.42
6 DOF - Dynamic	98.5	0.48
7 DOF - Static	98.7	0.43
7 DOF - Dynamic	97.9	0.50

showing the classification accuracy of the system in predicting successful and failed tasks.

The results presented in this section validate the effectiveness of the proposed DQL framework in achieving the novel contributions outlined in the study. The framework demonstrated superior computational efficiency, precision, and adaptability while maintaining scalability across varying scenarios. These findings highlight the potential of the proposed approach for real-world applications in autonomous robotics and industrial automation.

VI. CONCLUSION

This study presented a novel approach leveraging Deep Q-Learning (DQL) to optimize path planning and control for robotic arms. Therefore, by integrating the reinforcement learning methods in the context of the developed advanced kinematic model, the key problems that appeared during the framework development have been formulated and solved, turning into critical issues such as real-time adaptability, accuracy, computational costs, and scalability. These results indicate that the proposed method can provide higher accuracy for path optimization, faster decision-making time, and better collision avoidance than the traditional approach. The experimental evaluation affirmed the DQL framework's resilience in the conditions' heterogeneity. The framework also performed consistently better than the existing methods for different degrees of freedom and payload load configurations. It showed great promise for addressing a range of industrial and research problems. These results effectively revealed a significant cut in computational complexity, enabling the framework to be implemented in real-time, which is paramount in robotics and automation. Furthermore, the study underscored the importance of incorporating task-specific constraints into the reward function. This would enable the robotic arm to learn optimal policies that balance precision, energy efficiency, and safety. Features like prioritized experience replay and the target network stabilization we introduced earlier helped enhance the framework's stability and convergence. This research bridges gaps in robotic path planning and control by providing a scalable and efficient solution with real-world applicability. Future work may focus on extending this framework to multi-agent robotic systems, integrating additional sensory modalities, and testing in real-world industrial environments to further validate its utility and adaptability. The findings serve as a foundation for advancing autonomous robotics and industrial automation technologies.

FUNDING

This work is supported by the fund of 2023 Nanchong Science and Technology Program Project (No.23YYJCYJ0032) in Sichuan Province, China.

REFERENCES

[1] V. Bucinskas, A. Dzedzickis, M. Sumanas, E. Sutinys, S. Petkevicius, J. Butkiene, D. Virzonis, and I. Morkvenaite-Vilkonciene, "Improving industrial robot positioning accuracy to the microscale using machine learning method," *Machines*, vol. 10, no. 10, p. 940, 2022.

[2] E. Okafor, M. Oyedeji, and M. Alfarraj, "Deep reinforcement learning with light-weight vision model for sequential robotic object sorting," *Journal of King Saud University-Computer and Information Sciences*, vol. 36, no. 1, p. 101896, 2024.

[3] J. Xue, X. Kong, G. Wang, B. Dong, H. Guan, and L. Shi, "Path planning algorithm in complex environment based on ddpg and mpc," *Journal of Intelligent & Fuzzy Systems*, vol. 45, no. 1, pp. 1817–1831, 2023.

[4] H. Kabir, M.-L. Tham, and Y. C. Chang, "Internet of robotic things for mobile robots: concepts, technologies, challenges, applications, and future directions," *Digital Communications and Networks*, vol. 9, no. 6, pp. 1265–1290, 2023.

[5] T. Grenko, S. Baressi Šegota, N. Andelic, I. Lorencin, D. Štifanic, J. Musulin, M. Glucina, B. Franovic, and Z. Car, "On the use of a genetic algorithm for the determining ho–cook coefficients in continuous path planning of industrial robotic manipulators. machines 2023, 11, 167," 2023.

[6] T. Zhang and H. Mo, "Reinforcement learning for robot research: A comprehensive review and open issues," *International Journal of Advanced Robotic Systems*, vol. 18, no. 3, p. 17298814211007305, 2021.

[7] X. Lu, Y. Chen, and Z. Yuan, "A full freedom pose measurement method for industrial robot based on reinforcement learning algorithm," *Soft Computing*, vol. 25, no. 20, pp. 13 027–13 038, 2021.

[8] X. Cheng, J. Zhou, Z. Zhou, X. Zhao, J. Gao, and T. Qiao, "An improved rrt-connect path planning algorithm of robotic arm for automatic sampling of exhaust emission detection in industry 4.0," *Journal of Industrial Information Integration*, vol. 33, p. 100436, 2023.

[9] Q. Gao, Q. Yuan, Y. Sun, and L. Xu, "Path planning algorithm of robot arm based on improved rrt* and bp neural network algorithm," *Journal of King Saud University-Computer and Information Sciences*, vol. 35, no. 8, p. 101650, 2023.

[10] M. A. Mousa, A. T. Elgohr, and H. Khater, "Path planning for a 6 dof robotic arm based on whale optimization algorithm and genetic algorithm," *Journal of Engineering Research*, vol. 7, no. 5, pp. 160–168, 2023.

[11] H.-H. Huang, C.-K. Cheng, Y.-H. Chen, and H.-Y. Tsai, "The robotic arm velocity planning based on reinforcement learning," *International Journal of Precision Engineering and Manufacturing*, vol. 24, no. 9, pp. 1707–1721, 2023.

[12] M. Q. Mohammed, K. L. Chung, and C. S. Chyi, "Review of deep reinforcement learning-based object grasping: Techniques, open challenges, and recommendations," *IEEE Access*, vol. 8, pp. 178 450–178 481, 2020.

[13] T. Li, F. Xie, Z. Zhao, H. Zhao, X. Guo, and Q. Feng, "A multi-arm robot system for efficient apple harvesting: Perception, task plan and control," *Computers and Electronics in Agriculture*, vol. 211, p. 107979, 2023.

[14] M. Cao, X. Zhou, and Y. Ju, "Robot motion planning based on improved rrt algorithm and rbf neural network sliding," *IEEE Access*, 2023.

[15] K. Merckaert, B. Convens, M. M. Nicotra, and B. Vanderborght, "Real-time constraint-based planning and control of robotic manipulators for safe human–robot collaboration," *Robotics and Computer-Integrated Manufacturing*, vol. 87, p. 102711, 2024.

[16] N. Feng and S. Wu, "Research on motion control and trajectory planning algorithm of mobile manipulator based on deep learning," in *2023 International Conference on Mechatronics, IoT and Industrial Informatics (ICMIII)*. IEEE, 2023, pp. 271–274.

[17] J. Qi, Q. Yuan, C. Wang, X. Du, F. Du, and A. Ren, "Path planning and collision avoidance based on the rrt* fn framework for a robotic manipulator in various scenarios," *Complex & Intelligent Systems*, vol. 9, no. 6, pp. 7475–7494, 2023.

[18] M. Sumanas, A. Petronis, V. Bucinskas, A. Dzedzickis, D. Virzonis, and I. Morkvenaite-Vilkonciene, "Deep q-learning in robotics: Improvement of accuracy and repeatability," *Sensors*, vol. 22, no. 10, p. 3911, 2022.

[19] Y. Bi, C. Qian, Z. Zhang, N. Navab, and Z. Jiang, "Autonomous path planning for intercostal robotic ultrasound imaging using reinforcement learning," *arXiv preprint arXiv:2404.09927*, 2024.

[20] Z. Leong, R. Chen, Z. Xu, Y. Lin, and N. Hu, "Robotic arm three-dimensional printing and modular construction of a meter-scale lattice façade structure," *Engineering Structures*, vol. 290, p. 116368, 2023.

[21] M. Cheng, H. Tang, U. A. Bhatti, and D. Li, "Optimized sustainable manufacturing through fuzzy control in image-based visual servoing with velocity and field-of-view constraints," *IEEE Transactions on Fuzzy Systems*, 2024.

[22] K. Sivamayil, E. Rajasekar, B. Aljafari, S. Nikolovski, S. Vairavasundaram, and I. Vairavasundaram, "A systematic study on reinforcement learning based applications," *Energies*, vol. 16, no. 3, p. 1512, 2023.

[23] C. Chen, H. Zhang, Y. Pan, and D. Li, "Robot autonomous grasping and assembly skill learning based on deep reinforcement learning," *The International Journal of Advanced Manufacturing Technology*, vol. 130, no. 11, pp. 5233–5249, 2024.

[24] S. A. Kumar, R. Chand, R. P. Chand, and B. Sharma, "Linear manipulator: Motion control of an n-link robotic arm mounted on a mobile slider," *Heliyon*, vol. 9, no. 1, 2023.

[25] N. He, Z. Yang, X. Fan, J. Wu, Y. Sui, and Q. Zhang, "A self-adaptive double q-backstepping trajectory tracking control approach based on reinforcement learning for mobile robots," in *Actuators*, vol. 12, no. 8. MDPI, 2023, p. 326.

[26] V. Rajendran, B. Debnath, S. Mghames, W. Mandil, S. Parsa, S. Parsons, and A. Ghalamzan-E, "Towards autonomous selective harvesting: A review of robot perception, robot design, motion planning and control," *Journal of Field Robotics*, vol. 41, no. 7, pp. 2247–2279, 2024.