Reinforcement Learning–Driven Intelligent Path Planning for Dual Tower Cranes in Modular Construction Sites

Shanshan Jiang[1], Yifan Wang[1], Bin Yang[1] and Zhaozheng Shen[1]

[1]College of Civil Engineering, Tongji University, China
2232553@tongji.edu.cn, 2410444@tongji.edu.cn, yangbin@tongji.edu.cn, 2232499@tongji.edu.cn

Abstract

The shift towards prefabricated and modular construction has increased reliance on tower cranes for handling heavier loads, making operational precision more critical than ever. However, traditional path planning methods often lack the adaptability and real-time performance required to prevent accidents in dynamic construction environments. This study develops a reinforcement learning framework for intelligent path planning of dual-cranes. A high-fidelity simulation environment was built in Unity 3D, integrating a BIM-defined worksite and a robust collision detection system that classifies risks into external and internal interactions. To address the complexity of dual-tower crane operations, a curriculum learning strategy was implemented, progressing from single-crane to cooperative dual-crane tasks. Knowledge transfer via behavior cloning and generative adversarial imitation learning further improves training efficiency. Experiments show that the proposed framework enables the trained agents to complete all curriculum stages, producing collision-free and efficient lifting paths in complex dual-crane scenarios. These findings indicate that reinforcement learning can enhance both safety and operational performance in modern crane operations.

Keywords –

Lifting path planning; Tower crane; Reinforcement learning; Construction simulation; Modular construction

1 Introduction

Driven by industrialized construction, the Architecture, Engineering, and Construction (AEC) sector is pursuing higher prefabrication rates, with modular construction representing a highly integrated approach [1]. However, the installation phase of modular projects depends critically on the precise operation of heavy lifting equipment such as tower cranes [2]. Compared with conventional precast elements, modular units are larger and heavier, which increases collision risk during lifting [3]. Dual-crane operations, which are common in modular settings, further raise the demands on safe operation. Consequently, a robust path-planning methodology is of clear engineering significance for improving safety and efficiency.

Crane lifting path planning is a complex, dynamic decision-making problem [4]. In current practice, paths are coordinated by signalers and operators using experience-based procedures, which are inadequate for increasingly complex site conditions and may introduce additional safety risks [5]. Traditional planning methods also struggle with dynamic, uncertain environments and the multi-dimensional constraints of crane operations. Dual-crane scenario intensifies these challenges and further complicates motion coordination and collision avoidance.

This study develops a digital simulation environment for tower-crane lifting operations by integrating BIM with Unity 3D. Using crane kinematic modeling and Unity’s physics engine, the simulation enables realistic lifting behavior. A collision detection mechanism with an optimized bounding-box algorithm is implemented to support safety monitoring and reduce computational cost. To address the complexity of dual-crane coordination, a staged curriculum learning strategy is adopted, decomposing the task into pathfinding, obstacle avoidance, and cooperative manipulation. The reinforcement learning framework, including state, action, and reward design, is optimized to improve agent decision-making. Finally, a modular housing project is used as a case study to validate the proposed approach.

2 Literature Review

2.1 Lifting Path Planning Methods

Lift path planning for cranes aims to compute collision-free, executable trajectories that improve safety and productivity rather than simply minimizing geometric distance [6]. Because construction sites are dynamic and cluttered, the shortest path is often impractical for operators [7]. As a result, studies usually adopt multi-objective formulations that balance time, energy use [8], safety risk, and operator difficulty [9]. Methods are commonly grouped into graph-search, sampling-based, and metaheuristic approaches.

Classical graph searches such as Dijkstra and A* [10] can find shortest paths on discretized spaces, but they scale poorly when the configuration space grows and obstacles move [11] ; finer grids raise computational cost, while coarser grids lose detail and may compromise safety margins. Sampling-based planners such as RRT and its variants [12], [13] avoid full configuration-space modeling and are efficient at finding feasible paths, yet they are not guaranteed to be optimal and often produce "zigzagged" motions [14] that require smoothing and can be awkward near people and assets. Metaheuristics search large, complex spaces and naturally support multiple objectives [15]. GAs are widely used for crane path optimization and benefit from parallel implementations [4] to speed up collision checking in complex scenes, and hybrids such as Particle Swarm Optimization and Simulated Annealing [2] can further improve path smoothness and success rates. These limits motivate learning-based policies that adapt online while respecting safety envelopes and operator needs.

2.2 Reinforcement Learning and dual-crane lifting

Several researchers have implemented and tested various Reinforcement Learning (RL) and Deep Reinforcement Learning (DRL) approaches for crane lift path planning. Cho and Han [16] firstly proposed an RL-based method incorporating the kinematic actuator system of a tower crane into spatio-temporal lift planning within 3D virtual environments. They designed and compared six training strategies, adopting the on-policy Proximal Policy Optimization (PPO) and off-policy Soft Actor-Critic (SAC) algorithms. Wang et al. [17] proposed a DRL planner for online lifting path planning in unknown dynamic environment, using local environmental data from lidar sensors mounted above the hook. They designed a new reward function specifically for path smoothness to minimize frequent operation switching which makes the path more practical. To reduce the burden of reward design, Wang et al. [18] presented a framework combining Imitation Learning and RL (specifically Behavioural Cloning, PPO, and Generative Adversarial Imitation Learning). This approach aimed to learn from expert demonstrations collected via Virtual Reality.

Collectively, these RL and DRL studies show the promise of learning-based planning for single-crane operations. However, modular construction often involves two tower cranes working simultaneously in overlapping workspaces [19]. In such settings, each crane's jib, trolley, and suspended load become time-varying obstacles for the other, so the planner must detect conflicts quickly, avoid collisions in real time, and coordinate task timing [20]. Existing methods do not yet meet these requirements in a robust, real-time manner. This gap points to the need for RL-based methods for dual-crane lifting that can handle dynamic mutual interference and real-time deconfliction while remaining practical for site operations.

3 Methodology

The proposed RL-based lifting path planning method consists of three components, as shown in Figure 1. Unity-based virtual construction site provides the environment. A collision-monitoring module checks clearances and safety. The tower crane is modeled as the RL agent.

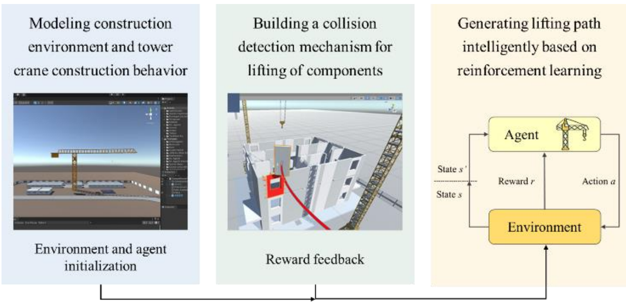

Figure 1. The framework of methodology

3.1 Tower Crane Construction Behavior Simulation Based on Unity

3.1.1 Establishment of Virtual Prototype for Construction Scene

The tower-crane lifting environment was modeled in Autodesk Revit, and the Building Information Model was exported in Industry Foundation Classes (IFC) format to enable lossless transfer from BIM to Unity. The IFC schema follows the EXPRESS standard, where each entity is assigned a unique identifier that defines its attributes and relationships, thus ensuring complete preservation of building information.

To resolve the differences between IFC and Unity in coordinate conventions, data precision, and naming rules, this study employs the IFC Importer plugin for optimized conversion. The tool performs geometric alignment and property mapping, converting the IFC model into a Unity-recognizable hierarchy of game objects while retaining all entity metadata, materials, and type information. This process, as shown in Figure 2, results in a virtual prototype of the construction site within Unity, providing a solid foundation for subsequent simulation and analysis.

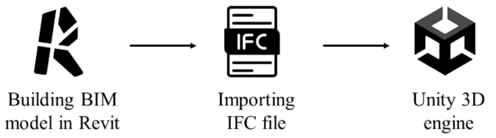

Figure 2. Establishment process for virtual construction scene

3.1.2 Realization of Tower crane construction behavior in Unity

Common lifting equipment in modular construction includes tower cranes and mobile cranes. Tower cranes are more prone to collision risk. This study focuses on flat-top tower cranes. The basic structure comprises the tower body, jib, tower top, trolley, sling, hook, base, operator cabin, and safety devices, as shown in Figure 3.

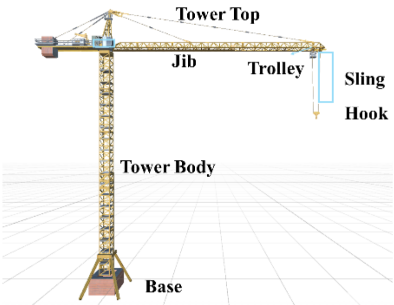

Figure 3. Tower crane rigid body model

Tower-crane lifting operations can be decomposed into three primary motions: jib rotating, trolley travel, and hook hoisting.

(1) Jib rotating and implementation

Jib rotating refers to rotation about the central axis of the tower body, with the jib and its attached components rotating around this axis. For the lifted object, the horizontal projection changes while the vertical height remains constant. In this study, the Unity function Transform.RotateAround is used, which is well suited to rotation about a fixed point and a fixed axis.

(2) Trolley travel and implementation

Trolley travel refers to the horizontal motion of the trolley along the jib driven by the luffing mechanism. For the lifted object, only the horizontal projection changes; the vertical height remains constant. In Unity, we define a parent–child hierarchy where the trolley is the child of the jib, so the trolley's pose is evaluated in the jib's local frame and updates consistently with the jib's motion.

Implementation proceeds in three steps. First, constrain the trolley's Z coordinate to the valid rail span with Mathf.Clamp to prevent boundary overruns. Second, assign the constrained value through Transform.localPosition to set the trolley's precise location along the jib. Third, execute motion along the local Z axis using Transform.Translate to achieve smooth, frame-consistent travel.

(3) Hook hoisting and implementation

Hook hoisting adjusts the load height through vertical motion of the hook. The horizontal position remains fixed while the vertical coordinate varies. Implementation follows the same logic as trolley travel.

Based on actual operating practice, the following assumptions are made for crane motion.

a. Hook rotation is not treated as a decision variable for the crane in this paper.

b. The long axis of the lifted component should align with the jib, and this adjustment is performed by workers.

c. During lifting, the bottom surface of the component remains parallel to the ground.

Under these assumptions, the crane can be simplified as a three-degree-of-freedom robot with rotating, trolley travel, and hoisting. Let the degrees of freedom be expressed in spatial coordinates.

φ is the counter-clockwise angle between the slewing jib and the X axis of the local frame; r is the distance from the trolley to the slewing centre along the jib; h is the height difference between the hook and the rotating centre. These are shown in Figure 4.

(4) Attachment and detachment of the component

In Unity, child objects inherit the motion of their parent while the local pose is preserved. Pick-and-place is implemented as a proximity-triggered reparent-and-align routine with two thresholds d_{on} for pickup and d_{off} for release. When the hook reaches a threshold, the component is reparented to the hook or to the building and oriented with Transform.RotateAround. At pickup the long axis is aligned with the jib. At release the component is rotated to the installation pose and lowered with MoveTowards function for a smooth set-down. The crane state then switches between loaded and unloaded.

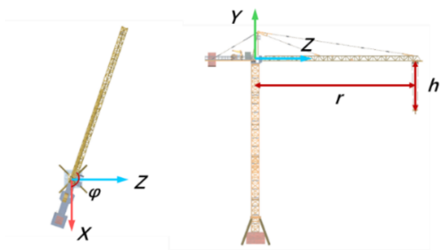

Figure 4. Parameters defining the tower crane configuration

3.2 Collision Detection Mechanism of Tower Crane Lifting

Previously, we proposed a minimum-distance method for collision risk in lifting operations [21] that separates collisions between the lifted component and the environment from collisions between cranes. Building on this work, we use a vertically aligned oriented bounding box, VA-OBB, with a minimum-distance metric tailored to modular construction. The system monitors clearances in real time. Two requirements are enforced: the operating spaces of the two cranes do not overlap and are separated by a defined safety margin, and each lifted component maintains the required clearance from surrounding entities.

3.3 Dual-crane Lifting Path Planning Based on Curriculum Learning

In this study, the tower crane is modeled as a three-degree-of-freedom robotic system. Lifting path planning is therefore treated as a motion-planning problem. Using the configuration-space concept, any crane state at a given time is a point in C-space. The task is to specify the load's pick and place positions and to compute a feasible, preferably optimal, sequence of configurations that transfers the load from pick to place. The optimality criteria typically include minimizing the lifting path length, reducing operation time, and lowering energy consumption. By encoding obstacles as constraints and optimizing an initial trajectory against these objectives, the problem becomes a constrained optimization formulation, which can be written as:

$$\begin{cases} \min f(x), x \in R_n \\ g_i(x) \le b_i, i = 1,2,3, \dots p \end{cases} \tag{1}$$

$f(x)$ is the objective function to be minimized. $g_i(x)$ are the constraint functions capturing workspace limits and collision avoidance, with p constraints in total.

This study focuses on a dual-tower crane system operating within the same construction area. To prevent collisions, the minimum distance between lifted components and surrounding obstacles, as well as the minimum clearance between the two cranes, must exceed the safety limits specified by relevant standards.

Dual-crane path planning falls under the broader category of multi-agent path planning, whose key challenge lies in coordinating the motion of two agents (cranes) in a complex environment to achieve efficient and collision-free operation. However, multi-agent reinforcement learning suffers from sparse rewards, where agents typically receive positive feedback only when all reach their goals, making effective learning difficult. To address this issue, curriculum learning is introduced. We decompose the dual-crane task into three stages: pathfinding, obstacle avoidance, and cooperation, allowing the agents to gain frequent early rewards and gradually adapt to more complex coordination.

3.3.1 Curriculum 1: single tower crane lifting path planning

The learning objective is to enable a single tower crane to navigate from the pickup point to the installation point while avoiding static obstacles. In this stage, interactions with other agents are ignored. According to the Markov Decision Process, the following elements are defined:

(1) Agent set N: Only one agent tc_0 is involved.

(2) State space Σ: Each state fully describes the planning environment, including crane's pose information, component information, construction environment information and task information.

(3) Action space A: The policy network outputs three actions $\mathcal{A} = [A_J, A_T, A_H]$ corresponding to jib rotation, trolley travel, and hook hoisting, normalized to $[-1,1]$ for training stability.

(4) Observation space: The crane perceives the distance to the target point, including angular, radial, and vertical differences, as well as the collision distance to surrounding objects.

(5) Reward function: The agent optimizes its policy through interaction with the environment based on received rewards. In this study, the agent learns with a mixed scheme of discrete events and continuous signals, as summarized in Table 1. Discrete rewards mark milestones: DR_1 for successful pickup and loading, DR_2 for successful installation and unloading, and a large negative reward DR_c for any collision. Continuous terms reflect proximity, heading, safety margin, and delay. To stabilize learning, penalties CR_1 to CR_3 reduce distance errors in rotation, trolley position, and hook height, guiding the hook toward the target. A multi-threshold positive reward CR_d is given when rotation error is below 3°, trolley offset is below 0.5 m, or hook-height difference is below 1.5 m, which accelerates precise control. Safety is further shaped by CR_c, a negative term that increases as collision distance decreases, encouraging early avoidance. Efficiency goals are encoded through penalties for time and energy: a small per-step cost promotes shorter completion time, and per-action costs encourage economical motion. All components are aggregated as the path-optimization reward CR_y.

The training and testing environment for the crane motion-planning model is illustrated in the Figure 5. Dark gray irregular cubes denote the planned building structure, the purple cube marks the target component to be lifted, and the green curve represents the lifting path. The pickup and installation positions are also indicated.

Table 1. Reward function in Curriculum 1

Reward Type	Agent Action	Reward Function / Reward Value
Task Completion Reward	Hook reaches the pickup point and completes hooking	$DR_1 = +2000$
	Hook reaches the installation point and completes unhooking	$DR_2 = +2000$
Crane Motion-related Reward	Tower rotation moves away from target point	$CR_1 = -c_1 \times (\varphi_{diff}/180°)^2, (c_1 > 0)$
	Trolley travels away from target point	$CR_2 = -c_2 \times (r_{diff}/12)^2, (c_2 > 0)$
	Hook lifting or lowering moves away from target height	$CR_3 = -c_3 \times (h_{diff}/20)^2, (c_3 > 0)$
	Tower rotation angle difference smaller than threshold	$CR_d = +0.5$
	Trolley distance difference smaller than threshold	$CR_d = +0.5$
	Hook height difference smaller than threshold	$CR_d = +0.5$
Collision-related Reward	Collision occurs	$DR_c = -50$
	Collision distance decreases	$CR_c = -c_4 \times (d_{recom} - d_c), (c_4 > 0)$
	Path optimization	$CR_y = -c_5 \times f, (c_5 > 0)$

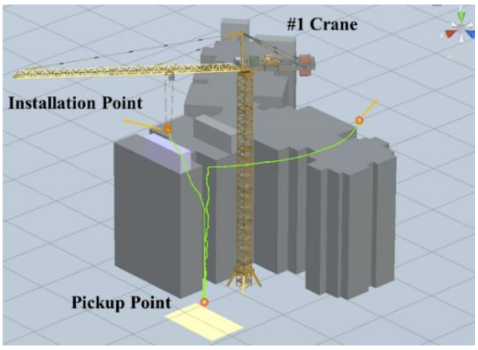

Figure 5. Training environment and lifting path in Curriculum 1

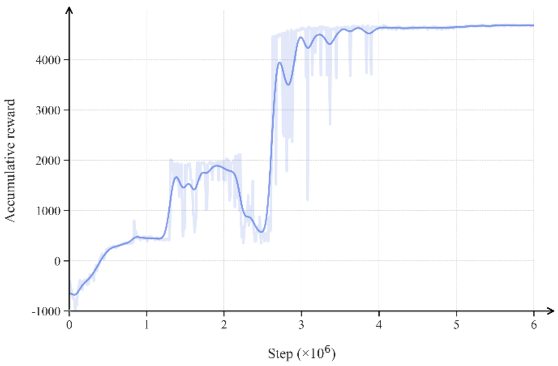

Figure 6. Learning curve of accumulative reward in Curriculum 1

During training, the Proximal Policy Optimization (PPO) algorithm is employed for 6×10^6 steps, with the maximum length of each episode set to 10,000 steps. An episode terminates under two conditions: either the maximum step limit is reached, or the collision distance during lifting falls below the predefined safety threshold. The learning curve in Figure 6 shows rewards near 2000 between 1.5×10^6 and 2×10^6 steps， indicating that basic grasping has been learned. As training proceeds, the cumulative maximum reward converges to about 4600, indicating efficient grasping and successful installation.

3.3.2 Curriculum 2: single tower crane lifting path planning in dual-crane environment

This stage focuses on optimizing the motion and obstacle-avoidance strategy of one tower crane in the presence of another fixed crane. Unlike Curriculum 1, where a single crane operates independently, this curriculum introduces static environmental constraints. The crane agent must not only optimize its own trajectory but also consider the workspace of the other crane and evaluate potential collision risks.

(1) Agent set N: Two tower-crane agents are involved, denoted as tc_0 and tc_1.

(2) Joint observation space Σ : Compared with Curriculum 1, each crane agent can additionally observe the position and target point of the other agent.

(3) Reward function: At each step, each crane receives an individual reward from its own observations and actions, consistent with Curriculum 1. To capture inter-crane risk, two safety terms are added: DR_T, a large discrete penalty for any collision, and CR_T, a continuous penalty applied when the separation between the two operating spaces falls below a set threshold, increasing with risk. The added terms are summarized in Table 2.

To transfer knowledge from Curriculum 1, we use the Behavioural Cloning (BC) and Generative Adversarial Imitation Learning (GAIL) modules in ML-Agents. At initialization, Unity demonstration trajectories for Crane 1 are recorded from start to grasp to installation. These data pretrain the agents through BC and GAIL, improving sample efficiency and accelerating convergence.

Curriculum 2 still adopts PPO. The training environment of Curriculum 2 is shown in Figure 7. After

9×10^6 training steps, the learning curve of Crane 2 in the is shown in Figure 8. Relative to Curriculum 1, behavioural-cloning pretraining markedly improves initial learning efficiency. Nevertheless, spatial interference with Crane 1 triggers collision penalties more frequently, yielding stronger reward instability than in the single-crane case.

Table 2. Added reward function in Curriculum 2

Agent Action	Reward Function / Reward Value
Collision occurs between the operating spaces of tower cranes	$DR_T = -50$
Collision distance between tower crane operating spaces decreases	$CR_T = -c_6 \times (d_{recom} - d_T),\ (c_6 > 0)$

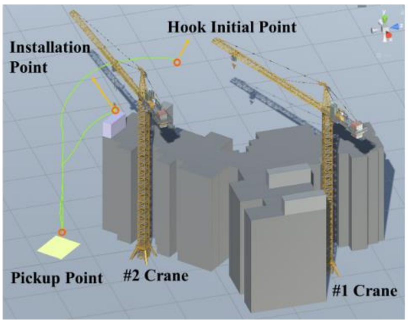

Figure 7. Training environment and lifting path in Curriculum 2

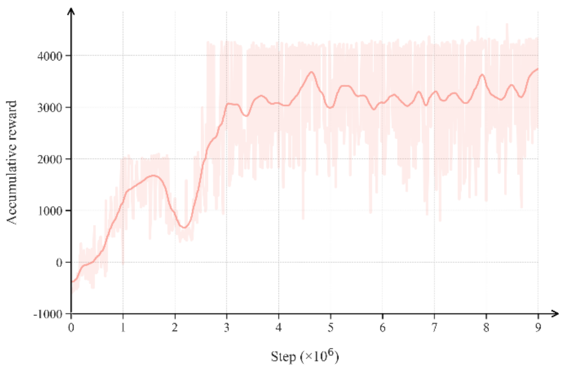

Figure 8. Learning curve of accumulative reward in Curriculum 2

3.3.3 Curriculum 3: dual tower crane lifting path planning

Curriculum 3 upgrades to dynamic dual-crane coordination, in which both cranes perform path planning and motion control simultaneously. Each crane must avoid static obstacles in the environment, perceive the other crane's motion state in real time, and adapt its own trajectory to prevent mutual interference.

To transfer knowledge from earlier curricula, training in this stage uses both BC and GAIL. Specifically, Crane 1 is pretrained with demonstrations recorded in Curriculum 1, while Crane 2 is initialized using trajectories collected in Curriculum 2.

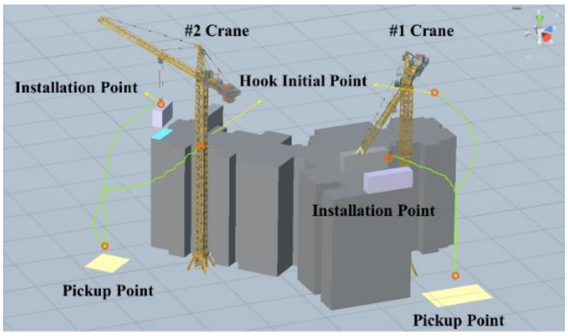

Figure 9. Training environment and lifting path in Curriculum 3

The training environment of Curriculum 2 is shown in Figure 9. Figure 10 presents the learning curves of the two crane agents and reveals distinct evolution patterns. Crane 2 learns grasping faster in the early phase but exhibits larger performance variability, whereas Crane 1 shows greater stability. After 12×10^6 steps, both agents achieve lower cumulative rewards than in their respective curriculum stages. The decline reflects the increased dynamism and complexity of the construction setting, where inter-crane collision risk induces more frequent negative rewards.

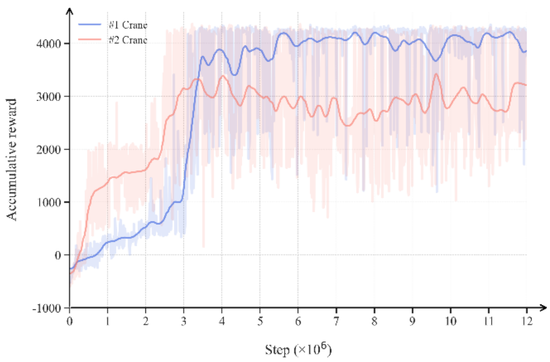

Figure 10. Learning curve of accumulative reward in Curriculum 3

During the transition between curriculum stages, the trained policy networks from earlier stages are used to initialize the networks in subsequent stages. In particular, the policy trained in Curriculum 2 is used as the initial parameter set for Curriculum 3, and the entire network is further fine-tuned during dual-crane cooperative training. This transfer mechanism allows the agent to retain previously learned obstacle-avoidance capabilities while adapting to the more complex coordination task. Although staged curriculum learning may introduce the risk of bias toward locally optimal strategies, stochastic exploration during reinforcement learning helps the agents continue searching for improved cooperative behaviors.

4 Case Study

The case study is drawn from a high-rise concrete modular housing project. The building comprises 42 stories, with 19 flats on each typical floor. Two representative modules on the seventh floor are selected for analysis. Module 9578 is assigned to Crane 1, and Module 30330 to Crane 2.

Using the deep reinforcement learning framework, cooperative policies for dual-crane operation were trained and exported as two ONNX models, TowerCrane1 and TowerCrane2. Deploying the best Curriculum 3 policies to the corresponding decision modules enabled successful lifting of the modular components in the digital simulation environment as shown in Figure 11.

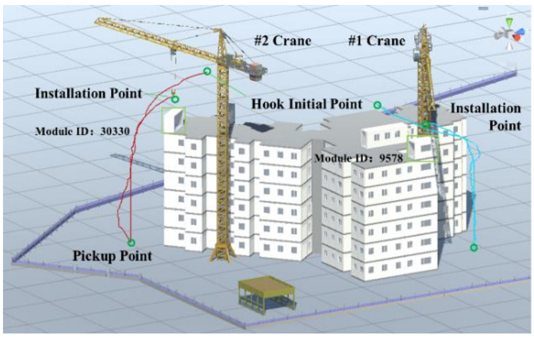

Figure 11. Lifting path of target modules

As shown in Figure 12, the coordinates along all three degrees of freedom vary smoothly throughout the lift, with no pronounced oscillations or abrupt jumps. This indicates that the learned policy satisfies the basic stability requirements of practical lifting operations.

The action space $\mathcal{A} = [A_J, A_T, A_H]$ is defined to control crane motion. Figure 13 illustrates the action outputs during the task, which map directly to crane control commands.

According to Tower Crane Safety Code GB5144-2006 in China, the clearance between the machine and surrounding buildings or facilities shall not be less than 0.6 m. We record the minimum distance between the task module and other scene entities from pickup to set-down across training curricula. The results show that the clearance remains above the safety threshold for the entire lifting process.

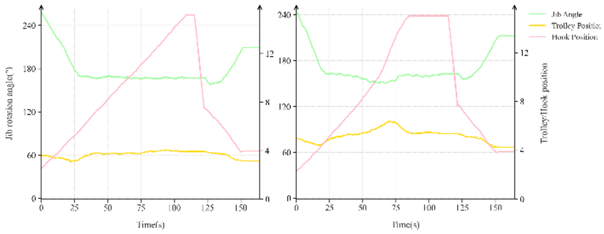

Figure 12. Configuration space coordinates of #1 Crane (left) and #2 Crane (right)

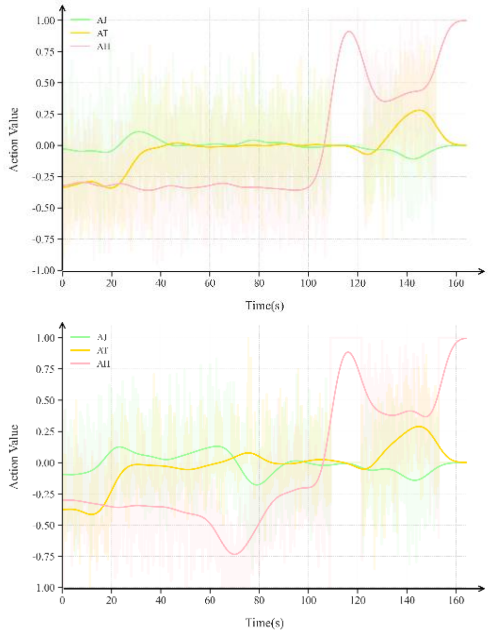

Figure 13. Action value output of #1 Crane (up) and #2 Crane (bottom)

5 Conclusion

This study developed a Unity-based simulation environment integrated with reinforcement learning for intelligent dual–tower-crane path planning. A high-fidelity construction scene was built in Unity to simulate crane operations and provide a training platform for RL agents. To address the complexity of dual-crane coordination, a curriculum learning strategy was introduced, progressing from single-crane path planning to obstacle avoidance and finally to dual-crane cooperation. Knowledge transfer through behavior cloning and generative adversarial imitation learning improved training efficiency. A modular high-rise residential project was used as a case study to evaluate path quality and safety. The results show that the learned policies generate smooth lifting trajectories while maintaining safe clearances and avoiding workspace conflicts, demonstrating the practical potential of the proposed approach.

Although the framework is evaluated using a representative modular construction case, the learned policy can generalize to similar environments because the agent learns spatial relationships rather than fixed trajectories. Moderate variations in obstacle layouts or crane initial configurations can therefore be handled without major changes. However, when site layouts differ significantly, retraining or fine-tuning may be required. In addition, the current study adopts simplified kinematic assumptions and focuses on simulation-based validation. Future work will incorporate more detailed dynamics such as load sway, compare the approach with classical planners (e.g., A*, RRT*, and GA), and investigate real-world deployment in construction sites.

References

[1] H.-T. Thai, T. Ngo, and B. Uy. A review on modular construction for high-rise buildings. Structures, 28:1265–1290, 2020.

[2] A. Zhu, Z. Zhang, and W. Pan. Crane-lift path planning for high-rise modular integrated construction through metaheuristic optimization and virtual prototyping. Automation in Construction, 141:104434, 2022.

[3] Z. Zhang, M. O. Wong, and W. Pan. Virtual reality enhanced multi-role collaboration in crane-lift training for modular construction. Automation in Construction, 150:104848, 2023.

[4] P. Cai, Y. Cai, I. Chandrasekaran, and J. Zheng. Parallel genetic algorithm based automatic path planning for crane lifting in complex environments. Automation in Construction, 62:133–147, 2016.

[5] S. Hasan, A. Bouferguene, M. Al-Hussein, P. Gillis, and A. Telyas. Productivity and CO2 emission analysis for tower crane utilization on high-rise building projects. Automation in Construction, 31:255–264, 2013.

[6] S. Dutta, Y. Cai, L. Huang, and J. Zheng. Automatic re-planning of lifting paths for robotized tower cranes in dynamic BIM environments. Automation in Construction, 110:102998, 2020.

[7] C. Zhang and A. Hammad. Improving lifting motion planning and re-planning of cranes with consideration for safety and efficiency. Advanced Engineering Informatics, 26(2):396–410, 2012.

[8] Z. Zhang, B. Zhang, W. Hu, R. Zhou, D. Cao, and H. Yin. Dynamic three-dimensional lift planning for intelligent boom cranes. IEEE/ASME Transactions on Mechatronics, 28(5):2885–2896, 2023.

[9] S. Hu. A practicality and safety-oriented approach for path planning in crane lifts. Automation in Construction, 2021.

[10] X. Lin, Y. Han, H. Guo, Z. Luo, and Z. Guo. Lift path planning for tower cranes based on environmental point clouds. Automation in Construction, 155:105046, 2023.

[11] A. R. Soltani, H. Tawfik, J. Y. Goulermas, and T. Fernando. Path planning in construction sites: performance evaluation of the Dijkstra, A* and GA search algorithms. Advanced Engineering Informatics, 16(4):291–303, 2002.

[12] S. Mohammad Langari, F. Vahdatikhaki, and A. Hammad. Improving the performance of RRT path planning of excavators by embedding heuristic rules. Advanced Engineering Informatics, 62:102724, 2024.

[13] Y. Zhou, E. Zhang, H. Guo, Y. Fang, and H. Li. Lifting path planning of mobile cranes based on an improved RRT algorithm. Advanced Engineering Informatics, 50:101376, 2021.

[14] Y. Lin, D. Wu, X. Wang, X. Wang, and S. Gao. Lift path planning for a nonholonomic crawler crane. Automation in Construction, 44:12–24, 2014.

[15] K. Chen, B. You, Y. Zhang, and Z. Chen. Automatic lift path planning of prefabricated building components using semantic BIM, improved A* and GA. Engineering, Construction and Architectural Management, 2024.

[16] S. Cho and S. Han. Reinforcement learning-based simulation and automation for tower crane 3D lift planning. Automation in Construction, 144:104620, 2022.

[17] K. Wang, J. Li, Z. Yin, J. Zhang, and X. Ma. Deep reinforcement learning based online lifting path planning for tower cranes in unknown dynamic environments. International Journal of Advanced Robotic Systems, 21(5):17298806241283176, 2024.

[18] Z. Wang, C. Huang, B. Yao, and X. Li. Integrated reinforcement and imitation learning for tower crane lift path planning. Automation in Construction, 165:105568, 2024.

[19] F. Shahnavaz, H. Taghaddos, R. S. Najafabadi, and U. Hermann. Multi crane lift simulation using Building Information Modeling. Automation in Construction, 118:103305, 2020.

[20] P. Cai, Y. Cai, I. Chandrasekaran, and J. Zheng. Parallel genetic algorithm based automatic path planning for crane lifting in complex environments. Automation in Construction, 62:133–147, 2016.

[21] B. Yang, H. Zhang, and Z. Shen. Minimum distance calculation method for collision issues in lifting construction scenarios. Journal of Computing in Civil Engineering, 38(6):04024041, 2024.