www.cad-journal.net
Computer-Aided Design AND APPLICATIONS

Exploration of Adaptive Environment Design Strategy Based on Reinforcement Learning in CAD Environment

Xiaolong Chen[1] and Lin Chen[2]

[1]International School of Design, University of Sanya, Sanya 572022, China, xiaolongchen@sanyau.edu.cn
[2]International School of Design, University of Sanya, Sanya 572022, China, linchen@sanyau.edu.cn

Corresponding author: Lin Chen, linchen@sanyau.edu.cn

Abstract. This article tackles the challenge of designing CAD (Computer-Aided Design) environments and introduces an adaptive design strategy rooted in Reinforcement Learning (RL). We delve into the intricacies and variations inherent in CAD environment design, emphasizing the need for adaptability and exploring RL's potential to enhance CAD intelligence. To validate our strategy's efficacy, we've constructed a simulation platform that mimics real-world CAD design and interaction processes. Our findings reveal that the RL-driven adaptive design approach seamlessly adjusts CAD environment parameters and configurations to align with evolving design tasks, offering optimal design support. In contrast to traditional CAD setups, this adaptive approach notably boosts design efficiency, minimizes errors, and elevates user satisfaction. This strategy heralds a new era of intelligent CAD environment development, paving the way for technological advancements in engineering design. Its insights offer valuable guidance to scholars and practitioners alike, fostering continuous innovation in CAD environmental design technology.

Keywords: Computer-Aided Design; Reinforcement Learning; Adaptive; Environmental design
DOI: https://doi.org/10.14733/cadaps.2024.S23.175-190

1 INTRODUCTION

CAD, a design methodology enhanced by computer technology, assists designers throughout the entire process - from initial conceptualization to intricate detailing. This encompasses modelling, analysis, optimization, and documentation production. With the rapid development of artificial intelligence technology, visual navigation has become one of the key technologies for autonomous robot navigation. Target-driven visual navigation requires robots to autonomously find designated target positions in unknown environments through visual perception and reinforcement learning algorithms. However, in practical applications, the diversity and complexity of the environment pose significant challenges to visual navigation. In order to achieve the generalization of goal-driven visual navigation, deep reinforcement learning has become an important research direction. Devo et al. [1] utilized deep learning algorithms such as Convolutional Neural Networks (CNN) to extract features and learn representations from images, enabling robots to accurately perceive and understand environmental information. Integrate the visual perception information, position information, and other sensor data of the robot to form a state representation, which serves as input for reinforcement learning algorithms. Learn path planning strategies from the current state to the target state through reinforcement learning algorithms such as deep Q-networks (DQN) or policy gradient methods. CAD environment refers to the software system supporting CAD operation and its running environment, including hardware platform, operating system, CAD software and various auxiliary tools. With the continuous progress of technology, computer-aided design (CAD) has become an indispensable part of the modern design field. Among them, input technology and interaction technology, as the core components of CAD, have a significant impact on design efficiency and quality. In order to meet diverse design requirements, Erdolu [2] proposed an adaptive environment design framework aimed at optimizing CAD input and interaction experience and improving design efficiency. Traditional CAD systems often rely on fixed input devices and interactive interfaces, which greatly limit the flexibility and innovation of design. Therefore, developing a CAD system that can adapt to different user needs and device characteristics is particularly important. By collecting and analyzing user operation data during the design process, the system can understand the user's habits, preferences, and skill levels. These data provide an important basis for subsequent environmental perception and adaptive adjustment. Based on user behaviour analysis and environmental perception, the system can intelligently adjust input methods and interaction interfaces to adapt to different user needs and device characteristics.

An excellent CAD environment should be efficient, stable, easy to use, and extensible to meet the design needs of designers in different stages and fields. Deep reinforcement learning has become a key means for robots to make autonomous decisions and perform complex tasks. In the field of additive manufacturing, by combining deep reinforcement learning technology, the autonomous additive manufacturing capability of robots can be further improved, achieving efficient and high-quality manufacturing processes. Felbrich et al. [3] explored how to achieve additive manufacturing of autonomous robots in a computational design environment through distributed model-free deep reinforcement learning. Traditional robot additive manufacturing methods often rely on precise mathematical models and preset trajectory planning, which limits the flexibility of robots when facing complex environments and tasks. To this end, it proposes a method based on distributed model-free deep reinforcement learning aimed at enabling robots to autonomously learn and optimize additive manufacturing processes in a computational design environment. Distributed model-free deep reinforcement learning combines the advantages of distributed computing, model-free learning, and deep reinforcement learning. It does not require the establishment of precise mathematical models in advance but rather learns and optimizes decision-making strategies through interaction with the environment. CAD environment provides an efficient and convenient design platform for designers, which enables complex design tasks to be completed quickly. However, with the constant change of design requirements and the increasingly complex design problems, the traditional CAD environment has made it difficult to meet the needs of designers for adaptive and intelligent design. Therefore, it is particularly important to explore the adaptive environment design strategy in the CAD environment. In recent years, the filtering and informing design space method based on reinforcement learning has become a hot research topic. It can help designers quickly find design solutions that meet the requirements in the design space, thereby improving design efficiency and quality. Halskov and Lundqvist [4] explored a reinforcement learning-based filtering and informing design space method in CAD environments, as well as how it drives designers toward design space thinking. Design space refers to the collection of all possible design solutions. In a CAD environment, design space is often a high-dimensional and complex space that contains a large number of design solutions. Traditional design methods often struggle to effectively explore and utilize the entire design space, leading designers to miss out on some excellent design solutions. The filtering and informing design space method based on reinforcement learning can automatically filter out design solutions that do not meet the requirements by introducing intelligent algorithms. Simultaneously, providing designers with useful information about the design space to help them better understand and utilize it.

RL is a pivotal subfield of machine learning that explores how agents can learn optimal strategies to attain specific objectives through interaction with their environment. Robot assembly has become a key link in automated production lines. However, facing complex and ever-changing assembly environments and task requirements, achieving efficient and accurate assembly operations remains a challenge. Traditional robot assembly methods are usually based on fixed time scales for prediction and control, making it difficult to adapt to the needs of different assembly tasks. Therefore, Hou et al. [5] proposed a robot assembly reinforcement learning method based on fuzzy logic-driven variable time scale prediction, aiming to improve the flexibility and adaptability of robots in the assembly process. Robot assembly involves multiple stages, including part recognition, positioning, gripping, and assembly. In these stages, robots need to make corresponding decisions and adjustments based on different assembly tasks and environmental changes. Traditional prediction methods based on fixed time scales are difficult to adapt to such changes, which may lead to low assembly efficiency or unstable assembly quality. Therefore, studying a prediction method that can adaptively adjust the time scale according to different assembly tasks and environmental changes is of great significance. The fundamental premise of RL involves agents refining their behavioural strategies based on reward or punishment signals provided by the environment. This refinement occurs through iterative trial-and-error learning aimed at maximizing cumulative rewards or minimizing cumulative losses. Hu et al. [6] proposed a simulation-to-reality pipeline method based on deep reinforcement learning, aiming to train robots for autonomous navigation in chaotic terrain environments. Autonomous navigation requires robots to plan the optimal path in unknown environments in real-time based on perceptual information and drive autonomously. In a chaotic terrain environment, autonomous navigation of robots becomes particularly difficult due to the complexity and uncertainty of the terrain, as well as the possible presence of dynamic obstacles. Traditional navigation methods often rely on accurate environmental models and prior knowledge, but in practical applications, these conditions are often difficult to meet. To address this issue, it proposes a simulation-to-reality pipeline method based on deep reinforcement learning. This method utilizes deep reinforcement learning algorithms to train the navigation strategy of robots in a simulated environment and transfers the trained strategy to a real environment. By simulating real pipelines, we can fully explore and learn in the simulated environment, avoiding the risk of extensive trial and error in the real environment. Markov Decision Processes are often employed in RL to depict the agent-environment interaction, while value functions or policy gradients are utilized to determine the most effective strategy.

In this study, we aim to integrate RL, a cutting-edge machine learning technique, into the design of CAD environments. By leveraging intelligent algorithm learning and optimization, we can achieve adaptive adjustments within the CAD environment, thereby enhancing design efficiency and quality. This research not only fosters innovation in CAD technology but also provides robust support for the intelligent transformation of engineering design. Our article delves into the requisites and challenges associated with adaptive environment design in CAD settings. Furthermore, we construct an adaptive environment design model grounded in RL, devise and implement simulation experiments to validate the model's efficacy and superiority and present a comprehensive analysis and discussion of the experimental outcomes.

The innovations presented in this article are threefold: (1) The introduction of RL into CAD environment design, culminating in a novel adaptive environment design strategy. (2) The realization of adaptive adjustment and optimization within the CAD environment through the development of an intelligent algorithm model. (3) The verification of the proposed strategy's effectiveness and feasibility through simulation experiments paving the way for future practical applications.

The article comprises six sections. Section 1 outlines the research background, significance, content, innovations, and the article's organizational structure. Section 2 delves into the current research landscape and future trends. Sections 3 through 5 focus on the analysis of adaptive environment design within CAD settings, the RL-based adaptive environment design strategy, simulation experiment design and execution, as well as experimental results and discussions. Finally, Section 6 summarizes the study's key achievements and contributions, highlighting future research directions and potential trends.

2 RESEARCH STATUS AND DEVELOPMENT TREND

In the field of engineering design, deep reinforcement learning provides new ideas and methods for adaptive environment design. Lee et al. [7] explored the application of deep reinforcement learning in adaptive environment design in engineering design through case studies. Traditional engineering design methods often rely on the experience and professional knowledge of engineers, making it difficult to cope with complex and ever-changing design requirements. Deep reinforcement learning can automatically learn and optimize design schemes through interaction with the environment, thereby achieving adaptive environment design. By using deep neural networks to perceive and model the environment, key information related to design is extracted, providing a foundation for subsequent decision-making and optimization. Based on reinforcement learning algorithms, select the optimal design action based on the current environmental state and design goals and continuously optimize the design scheme through interaction with the environment. With changes in the environment and adjustments in design requirements, deep reinforcement learning can adaptively adjust design solutions to meet new requirements. WebBIM (Web Building Information Modeling) technology, as a bridge connecting virtual and real worlds, is receiving increasing attention. However, as the scale of WebBIM scenarios continues to expand, how to effectively load and present this data has become a huge challenge. To address this issue, Li et al. [8] proposed a collaborative multi-granularity interest scheduling algorithm called CEB (Collaborative, Evolutionary, and Behavioral) aimed at improving the loading efficiency and user experience of large-scale WebBIM scenarios. In the WebBIM scenario, traditional loading methods often adopt a single granularity data loading strategy, resulting in long loading time, high resource consumption, and difficulty in meeting the real-time interaction needs of users. Therefore, it proposes a CEB algorithm based on collaborative multi-granularity interest scheduling. This algorithm achieves efficient loading and intelligent optimization of WebBIM scenarios through collaborative work, dynamic evolution, and behaviour analysis. With the widespread application of machine learning (ML) in computer-aided design (CAD), its security issues are gradually becoming prominent. Especially adversarial perturbation attacks, as a new type of network attack, pose a serious threat to the stability and reliability of CAD systems. Liu et al. [9] used adaptive environment detection of lithography points as an example to explore the impact of adversarial disturbance attacks and defense strategies in CAD systems based on convolutional neural networks (CNN). In ML-based CAD systems, deep learning models such as CNN are widely used in tasks such as image recognition and pattern classification. However, these models are often susceptible to adversarial perturbation attacks during the training process, leading to a decrease in model performance or even failure. As an important part of CAD systems, the accuracy and stability of adaptive environment detection for lithography points are crucial for the operation of the entire system. Therefore, studying the impact of adversarial perturbation attacks on lithography point detection and its defense strategies is of great significance.

Liu et al. [10] proposed a dynamic stochastic human-machine collaborative task-level decision-making method based on dual agent deep reinforcement learning. Human-machine collaboration refers to the process of completing tasks through collaboration between humans and intelligent agents. In practical applications, due to the complexity and uncertainty of the environment, as well as the differences between humans and intelligent agents, human-machine collaborative decision-making becomes very difficult. Traditional decision-making methods often rely on fixed rules and models, making it difficult to adapt to dynamic and changing environments and the uncertainty of human behaviour. Therefore, it proposes a dynamic stochastic human-machine collaborative task-level decision-making method based on dual agent deep reinforcement learning, aiming to improve the efficiency and stability of human-machine collaboration. Dual agent, deep reinforcement learning, refers to the process in which two agents complete tasks through mutual learning and collaboration. In dual-agent deep reinforcement learning, each agent has its own neural network and decision strategy, which is optimized through interaction with the environment. Traditional path-planning methods often rely on precise environmental models and complex algorithms, but in actual industrial environments, it is often very difficult to establish accurate models due to factors such as uncertainty and dynamic changes. Therefore, studying how to perform path planning without a model has important theoretical and practical application value. In recent years, deep reinforcement learning has made significant progress in the field of path planning, providing new ideas for solving path-planning problems in complex industrial environments. Path planning refers to planning an optimal or feasible path from the starting point to the endpoint for a robot or other mobile entity in a specific environment. In complex industrial environments, path planning faces various challenges, such as the uncertainty of obstacles, the emergence of dynamic obstacles, and environmental changes. Traditional path-planning methods are usually based on precise environmental models, but in the absence of models, these methods are often difficult to apply. Deep reinforcement learning combines the perceptual ability of deep learning with the decision-making ability of reinforcement learning, providing a new solution for model-free path planning [11].

In recommendation systems, partial observability of the environment is a common issue. Due to the limited interaction between users and the system and the fact that user preferences and behaviour patterns may change over time, the system cannot fully and accurately understand the user's state and intention. To address this issue, Shang et al. [12] utilized Reinforcement Learning (RL) to estimate partially observable environments and enhance the inference ability of recommendation systems. A recommendation system is an intelligent system that can predict and recommend content that users may be interested in based on their historical behaviour and preferences. However, in practical applications, recommendation systems often face challenges from partially observable environments due to limited interaction data between users and the system, as well as uncertainty in user behaviour. A partially observable environment refers to a system that cannot directly observe the true state and intention of users and can only indirectly infer through user feedback and behaviour. With the rapid development of artificial intelligence technology, the navigation problem of mobile robots has become a highly concerning research field. Traditional navigation methods often rely on manually designed features and complex planning algorithms, while deep reinforcement learning provides new possibilities for the navigation of mobile robots. The end-to-end navigation strategy directly maps perception data to robot actions through deep neural networks, simplifying the navigation process and achieving more flexible and efficient navigation. Shi et al. [13] explored an end-to-end navigation environment strategy for mobile robots based on deep reinforcement learning. The navigation task of mobile robots involves multiple aspects, such as perception, decision-making, and execution. Traditional navigation methods often require the manual design of feature extractors and planning algorithms, which not only increases development difficulty but also makes it difficult to cope with complex and ever-changing environments. The end-to-end navigation strategy maps perception data directly to robot actions through deep neural networks without the need for explicit feature extraction and path planning, thus simplifying the navigation process.

Traditional motion planning methods are usually based on fixed rules and models, which make it difficult to cope with the dynamic changes and uncertainties of the environment. Machine learning technologies, especially deep learning and reinforcement learning, provide new solutions for adaptive environmental motion planning and control in mobile navigation. Xiao et al. [14] utilized deep learning algorithms such as Convolutional Neural Networks (CNN) and Recurrent Neural Networks (RNN) to process and analyze sensor data of robots, extract environmental features and information, and provide a foundation for subsequent motion planning and control. Based on machine learning algorithms such as reinforcement learning and deep learning, learn and plan the optimal motion path based on the current environmental state and goals. These methods can adaptively adjust planning strategies to cope with dynamic changes and uncertainties in the environment. In the development process of CAD, collaborative design and traditional design methods have each occupied important positions. Zhou et al. [15] compared and analyzed the design characteristics and advantages and disadvantages of adaptive environments in collaborative design and traditional computer-aided design, in order to reveal their different values and application prospects in modern design practice.

Collaborative design emphasizes collaboration and communication among experts from multiple disciplines and fields, focusing on real-time integration and sharing of design information during the design process to optimize design solutions. Traditional computer-aided design focuses more on a single designer or design team independently completing design tasks and emphasizes the use of computer technology and algorithms to assist design decisions. Adaptive environment refers to the ability of a design system to intelligently adjust design tools and interfaces based on different design needs and scenarios, providing users with a more personalized design experience.

In the past few years, the emergence of artificial intelligence technology has sparked a growing interest among scholars in exploring the use of machine learning techniques in CAD environment design. Their aim is to enhance adaptability and intelligence to an advanced level.

3 ANALYSIS OF ADAPTIVE ENVIRONMENT DESIGN IN CAD ENVIRONMENT

3.1 Concept and Application of Adaptive Environment Design

Adaptive environment design entails dynamically adjusting the parameters and setup of the CAD environment in response to evolving design tasks and designer preferences, ensuring optimal design support. It mandates that the CAD environment is capable of detecting external changes and reacting accordingly, enabling an intelligent and self-adjusting design process. This approach holds significant promise in engineering design disciplines like architecture, mechanics, electronics, and more. By flexibly adapting the CAD environment's parameters and configuration, design efficiency and quality are enhanced while designer workload and errors are minimized.

3.2 Challenges Faced by CAD Environment Design

When designing CAD environments, designers encounter numerous obstacles. Firstly, the vast array of complex design tasks necessitates a highly adaptable and customizable CAD environment to cater to the varying needs of different fields and stages. Unfortunately, conventional CAD systems often lack the necessary flexibility to adjust parameters and configurations responsively, leading to subpar efficiency and inconsistent quality. Secondly, managing data and facilitating collaboration within the CAD environment poses significant challenges. Designers must grapple with extensive design data and information while ensuring seamless team collaboration. Traditional CAD systems, however, often fall short in these areas, struggling to provide the real-time accuracy and consistency designers require. Lastly, the intelligence level of CAD environments remains a pressing concern. As artificial intelligence technology continues to evolve, designers increasingly demand more intelligent CAD environments. However, the traditional CAD environment still needs to be improved in intelligence, and it can't provide enough intelligent support and assistance for designers. Given the above challenges, adaptive environment design is particularly important (as shown in Table 1).

Challenge	*Advantages of adaptive environment design*
Design task change	Automatically adjust the parameters and configuration of the CAD environment according to the changes in design tasks.
	Provide the best design support and improve the design efficiency and quality.
	Reduce the workload and error rate of designers.
Data management and collaboration	Real-time perception of changes in design data and collaboration needs of team members.
	Automatically adjust data management and collaboration policies. Ensure the real-time accuracy and consistency of design data.
Intelligent level improvement	Introduce advanced machine learning and artificial intelligence technology.
	Learn the designer's behaviour and preferences.
	Provide personalized intelligent support and assistance to help

	designers complete design tasks more efficiently.

Table 1: Advantages of Adaptive Environment Design.

3.3 Application Potential of RL in Adaptive Environment Design

As an advanced machine learning method, RL has great application potential in adaptive environment design. First of all, RL can learn the optimal strategy through the interaction between agent and environment, which is very suitable for dealing with the adaptive problem in a CAD environment. An agent can automatically adjust the parameters and configuration of the CAD environment according to the changes in design tasks and the needs of designers to achieve optimal design support. Furthermore, RL can deal with complex decision-making problems. In the design of a CAD environment, it is often needed to make decisions based on many factors, such as the nature of design tasks, designers' preferences, and the characteristics of design data. RL can provide effective decision support for adaptive environment design by learning the mapping relationship between these factors and decision results. In addition, RL can also support online learning and real-time adjustment. In the process of using a CAD environment, the designer's behavior and preference may change, resulting in the original design strategy is no longer applicable. RL can adapt to these changes through online learning and real-time adjustment to ensure that the adaptive environment always provides the best design support.

4 RL-BASED ADAPTIVE ENVIRONMENT DESIGN STRATEGY

4.1 RL Model Construction

The goal of an RL-based adaptive environment design strategy is to improve the adaptive ability and intelligence level of the CAD environment to better support designers' design tasks. When designing a strategy, we need to follow the following principles: first, the strategy should be flexible and customizable enough to meet the design needs of different fields and stages; Secondly, the strategy should have efficient learning and optimization ability to adapt to the changes of design tasks quickly; Finally, the strategy should have stable performance and reliable robustness to ensure the feasibility and effectiveness in practical application.

In this article, an appropriate RL model is carefully designed when constructing an adaptive environment design strategy based on RL. This model is the core component of the adaptive environment design strategy. It can guide the adaptive environment in taking appropriate actions to optimize the design process by sensing and learning the state of the CAD environment. The proposed model comprises a state space, action space, and reward function. The state, denoted as $s \in S$, encapsulates the status details of the CAD environment, encompassing attributes φ on the design task alongside designer behaviours and preferences, designated as ψ.

$$s = \varphi, \psi \tag{1}$$

Among them:

φ It can be further refined into task types, complexity, requirements, etc.

ψ It can include designers' operating habits, design styles, historical behaviour data, etc.

In addition, the state space also contains designers' behaviours and preferences, such as designers' operating habits, design styles and historical behaviour data. By capturing this state information, the RL model can understand the current CAD environment more comprehensively and provide the basis for subsequent decision-making.

Action $a \in A$ is an action that can be taken by an adaptive environment, aiming at adjusting the parameters and configuration of the CAD environment according to the current state s. Action space can be defined as a set of operations such as adjusting interface layout, optimizing algorithm parameters, and providing personalized design tools.

$$A = \{ \text{adjust\_layout}, \text{optimize\_params}, \text{personalize\_tools}, \ldots \} \tag{2}$$

By defining rich action space, the RL model can flexibly respond to various design scenarios and realize adaptive environment adjustment.

The reward function gives the corresponding reward value to the action according to the feedback from the environment to guide the learning process of the RL model. In adaptive environment design, reward function can be defined based on design efficiency, design quality, user satisfaction and other indicators. When the actions taken by the adaptive environment can improve the design efficiency or reduce the design error rate, the reward function will give a positive reward. On the other hand, if the action leads to a decrease in design efficiency or user satisfaction, negative punishment will be given:

$$r = R(s,a) \tag{3}$$

Where r it can be positive, negative or zero. Through this reward mechanism, the RL model can gradually learn the optimal action strategy to achieve the goal of adaptive environment design. Strategy $\pi(s)$ is the core of RL model learning, which defines the action a that should be taken in the current state s:

$$a = \pi(s) \tag{4}$$

The strategy can be deterministic; that is, an action is determined for each state. It can also be random; that is, the probability distribution of an action is assigned to each state. The value function is used to evaluate the long-term return of adopting strategies in a given state.

$V(s)$ represents the state value function, that is, the expected return obtained by following the strategy π in the state s; $Q(s,a)$ represents the function of action value, that is, the expected return obtained by taking action a and following strategy π in the state s.

$$V(s) = E_\pi\left[R(s_0,a_0) + \gamma R(s_1,a_1) + \gamma^2 R(s_2,a_2) + \ldots \middle| s_0 = s\right] \tag{5}$$

$$Q(s,a) = E_\pi\left[R(s,a) + \gamma V(s')\right] \tag{6}$$

Among them $\gamma \in [0,1]$ is a discount factor, which is used to weigh the importance of immediate rewards and future rewards.

4.2 Implementation Process of Adaptive Environment Design Strategy

The implementation process of the RL-based adaptive environment design strategy includes the following steps: first, collect and process the state information of the CAD environment, including the characteristics of design tasks, designers' behaviours and preferences, etc. Then, according to the state information, choose appropriate actions to adjust, such as adjusting the parameters and configuration of the CAD environment. Then, observe the feedback from the environment and update the RL model. Finally, repeat the above steps until the termination conditions are met or the design requirements are met. See Figure 1 for details.

In the implementation process, we need to pay attention to the following points: (1) to ensure the accuracy and real-time performance of state information; Appropriate actions should be selected for adjustment to avoid interference with the design process; It is needed to reasonably set termination conditions and assessment indicators to measure the performance and effect of the strategy.

When constructing the RL model, we need to choose appropriate algorithms and tools to implement it. Commonly used RL algorithms include Q-learning and SARSA. Q-learning is an offline learning strategy, which seeks the optimal strategy by estimating the expected return of each state-action pair.

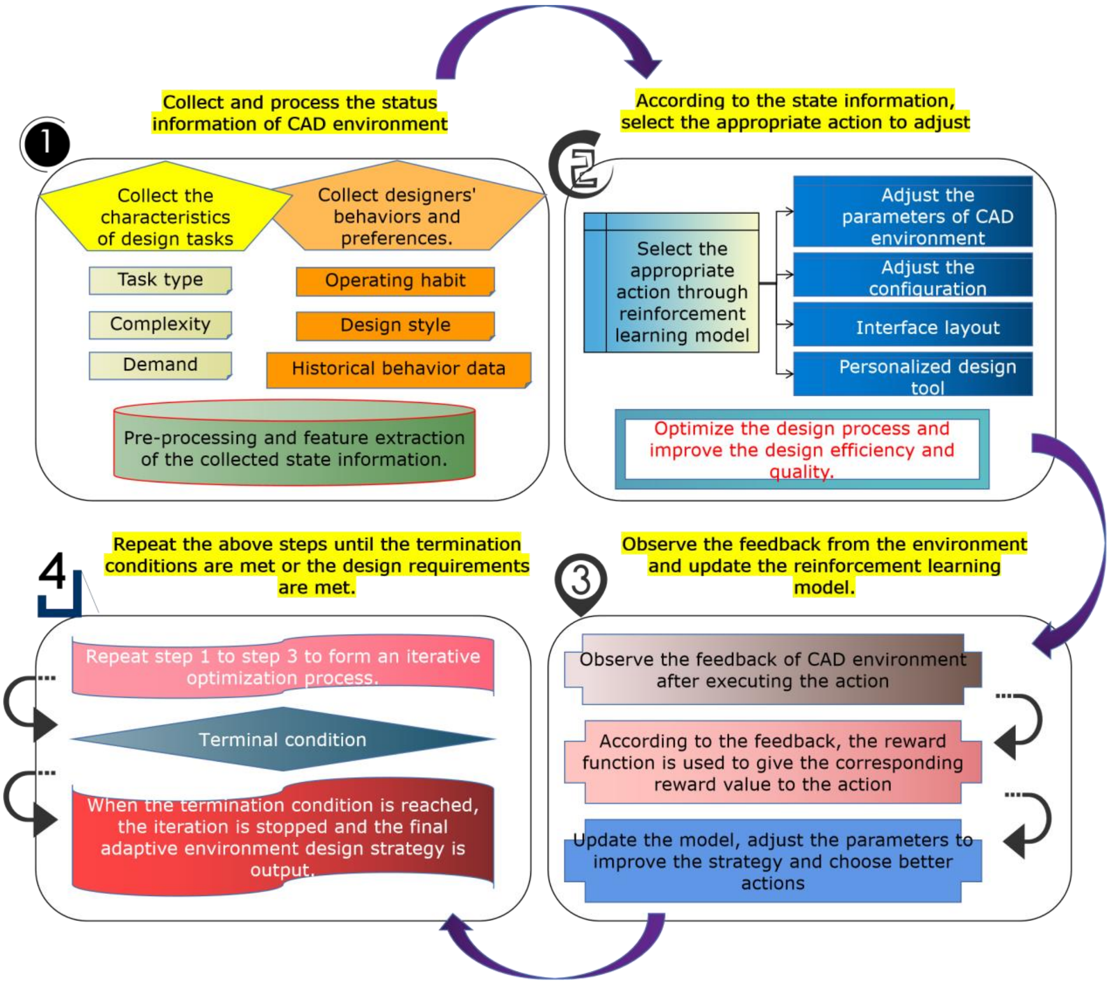

Figure 1: Implementation process of adaptive environment design strategy.

SARSA, on the other hand, is an online learning strategy, which actually performs the selected action in each iteration and updates the Q value according to the actual reward. These two algorithms have their own advantages, and the specific choice depends on the requirements and characteristics of the task. In this article, the Q-learning algorithm is adopted, and the Q value updating rule is the core of Q-learning, which is used to update the Q value of the state-action pair in each iteration. The Q value update formula is as follows:

$$Q(s,a) \leftarrow Q(s,a) + \alpha\left[r + \gamma \max_{a'} Q(s',a') - Q(s,a)\right] \tag{7}$$

Among them:

$Q(s,a)$ Represents the Q value of the action a taken in the state s.

α It is the learning rate that determines the range of change of Q value at each update.

r It is an immediate reward from the environment.

γ It is a discount factor, which determines the importance of future awards.

s' is a new state reached after taking action a in state s.

$\max_{a'} Q(s',a')$ represents the largest Q value among all possible actions in the new state s', which represents the estimation of the best action in the future.

Before the algorithm starts, the Q table needs to be initialized. Generally, the values of all state-action pairs Q are initialized to a smaller value or set based on some prior knowledge. The initialization formula is as follows:

$$Q(s,a) \leftarrow \text{Initialization value}, \forall_s \in S, a \in A(s) \tag{8}$$

Among them:

S It is a set of state spaces.

$A(s)$ Is a set of actions that can be taken in state s.

Initialization value can be a fixed number or a function value based on state or action.

Commonly used tools include deep learning frameworks such as Python programming language and TensorFlow. Choosing the appropriate programming language and deep learning framework is also very important for the realization and training of the model. Python, as a powerful and easy-to-learn programming language, is widely used in machine learning and RL fields. It provides a wealth of libraries and tools that can handle data, build models and visualize easily. TensorFlow is a popular deep learning framework that supports efficient numerical calculation and automatic differentiation and can easily construct and train complex neural network models. By combining Python and TensorFlow, this article can efficiently realize the construction, training and application of the RL model.

4.3 Policy Performance Assessment Index and Case Display

In order to evaluate the performance and effect of RL-based adaptive environment design strategy, it is needed to define some appropriate assessment indicators. Commonly used assessment indicators include design efficiency, design quality and user satisfaction. Design efficiency can be measured by comparing the difference between the adaptive environment and the traditional environment in the completion time of design tasks. The design quality can be measured by comparing the accuracy and consistency of design results between the adaptive environment and the traditional environment. User satisfaction can obtain the designer's overall assessment and use experience of the adaptive environment through questionnaires or user feedback. Through the application of these assessment indicators, the performance and effect of the strategy can be comprehensively and objectively evaluated, which will be explained in detail later.

This article chooses a commercial space as an experimental case, which needs indoor environment design, including layout planning, lighting design, furniture placement and so on. Firstly, a group of professional designers carry out environmental design according to traditional methods and experience without using an adaptive environmental design strategy. Then, in the same commercial space case, the adaptive environment design strategy is introduced. Figure 2 shows the design case before using the adaptive environment design strategy, and Figure 3 shows the design case after using the adaptive environment design strategy. It can be seen that the design case after using the adaptive environment design strategy is better.

In this article, a unified assessment standard is formulated to evaluate the quality of two groups of design schemes. The assessment criteria include space utilization, functionality, aesthetics, comfort and so on. By comparing the two groups of experimental data, it is found that the experimental group using an adaptive environment design strategy has advantages in design efficiency and quality. The specific results are shown in Table 2.

Figure 2: Design case before using adaptive environment design strategy.

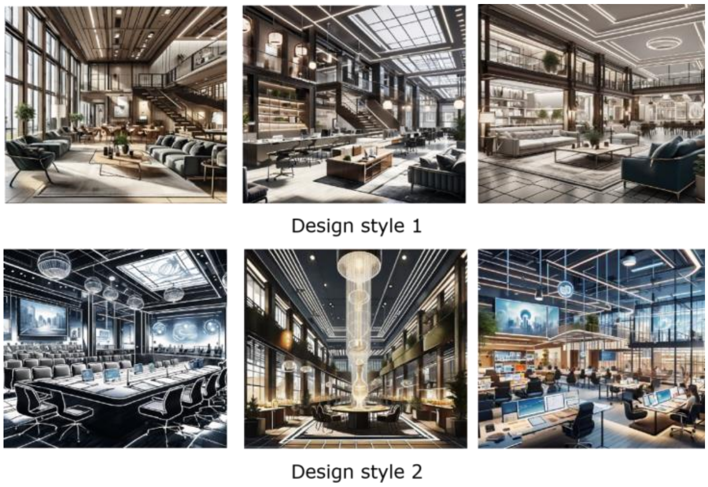

Figure 3: Design case after using adaptive environment design strategy.

Assessment index	*Control group*	*Experimental group*	*Ascending proportion*
Design time	Reference time	Reference time ×70%	Save about 30%
Number of modifications	Number of benchmark modifications	Number of benchmark revisions: ×60%	Reduce by about 40%
Designer satisfaction	Benchmark satisfaction	Benchmark satisfaction+benchmark satisfaction ×20%	Increase by about 20%
Quality of design scheme	Benchmark quality score	Benchmark quality score+benchmark quality score ×15%	Increase by about 15%

Table 2: Comparison of design results between experimental group and control group.

Design time: The time required for the experimental group to complete the design task is obviously less than that of the control group, saving about 30% time on average.

Revision times: Compared with the control group, the revision times of the experimental group in the design process are significantly reduced, with an average reduction of about 40%.

Designers' satisfaction: According to the questionnaire survey, the designers in the experimental group are significantly more satisfied with the adaptive environment design strategy than the control group, with an average increase of about 20%.

Quality of design scheme: According to the established assessment criteria, the design scheme of the experimental group is superior to that of the control group in many aspects, and the overall quality is improved by about 15%.

The experimental results show that the adaptive environmental design strategy shows significant advantages in actual environmental design cases. By introducing the RL model, an adaptive environment design strategy can automatically adjust design parameters and configuration according to the characteristics of design tasks and designers' behaviour preferences, thus improving design efficiency and quality. This is mainly due to the iterative optimization ability of the RL model, which enables the design strategy to gradually approach the optimal solution.

5 SIMULATION EXPERIMENT DESIGN AND RESULT DISCUSSION

5.1 Construction and Experimental Analysis of Simulation Experiment Platform

In order to verify the effectiveness of an RL-based adaptive environment design strategy, it is needed to build a simulation experiment platform. This platform should be able to simulate the design process and interactive process of a real CAD environment and provide the necessary interfaces and tools to support the training and application of the RL model. In this article, Python programming language and framework are selected to build the simulation model of the CAD environment and realize the interaction with the RL algorithm. Furthermore, suitable user interface and data visualization tools are designed to facilitate the display and analysis of experimental results.

During the experiment, this article collected and recorded relevant experimental data, including the characteristics of design tasks, designers' behaviours and preferences, parameters and configuration of CAD environment, quality and efficiency of design results, etc. These data are mainly obtained through the log files and databases of the simulation experimental platform. After collecting the data, the necessary pretreatment and analysis are carried out, including data cleaning, feature extraction, statistical analysis, etc., in order to extract useful information and rules. After data collection and processing, the experimental results are displayed visually. Figure 4 shows the design efficiency of the model.

The results show that in the adaptive environment, the average design time of designers is about 20% shorter than that in the baseline environment. This shows that the adaptive environment design strategy can significantly improve the design efficiency. This is because the strategy can automatically adjust the design parameters and configuration according to the characteristics of design tasks and designers' behavior preferences, thus reducing the time consumption of designers in manual adjustment and optimization.

Figure 5 shows the design quality of the model. The results show that the error rate of designers in an adaptive environment is about 30% lower than that in a baseline environment. This means that the adaptive environment design strategy not only improves the design efficiency but also significantly improves the design quality. The reduction of error rate is attributed to the iterative optimization ability of the RL model, which enables the design strategy to gradually learn the optimal design decision, thus reducing the errors and irrationalities in design. Figure 6 shows the user satisfaction before using the adaptive environment design strategy, and Figure 7 shows the user satisfaction after using the adaptive environment design strategy.

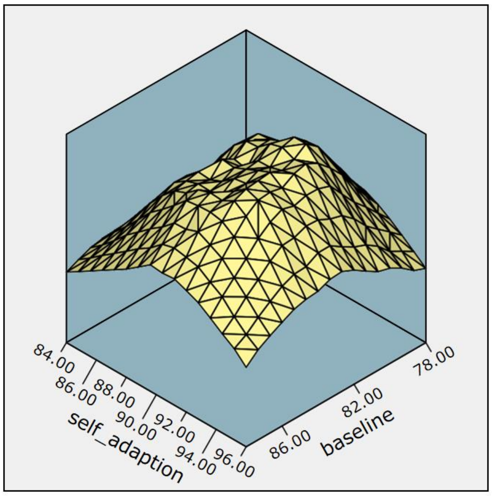

Figure 4: Design efficiency of the model.

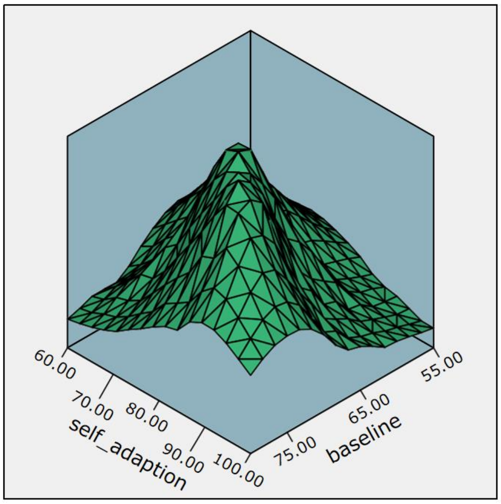

Figure 5: Design quality of the model.

By comparing Figure 6 and Figure 7, it can be clearly seen that after using the adaptive environment design strategy, user satisfaction has been significantly improved, with an average increase of about 25%. This shows that the adaptive environment design strategy can better meet the needs and expectations of users. The improvement of user satisfaction comes from the improvement of design efficiency and design quality. These factors work together to make users get a better design experience after using an adaptive environment design strategy.

To sum up, the results fully prove the effectiveness and superiority of the adaptive environment design strategy in practical application. By introducing the RL model, an adaptive environment design strategy can automatically adjust design parameters and configuration according to the characteristics of design tasks and designers' behaviour preferences, thus significantly improving design efficiency and quality. Furthermore, the improvement in user satisfaction further verifies the value and potential of this strategy in practical application. These results provide strong support for the wider application of adaptive environment design strategy in the future.

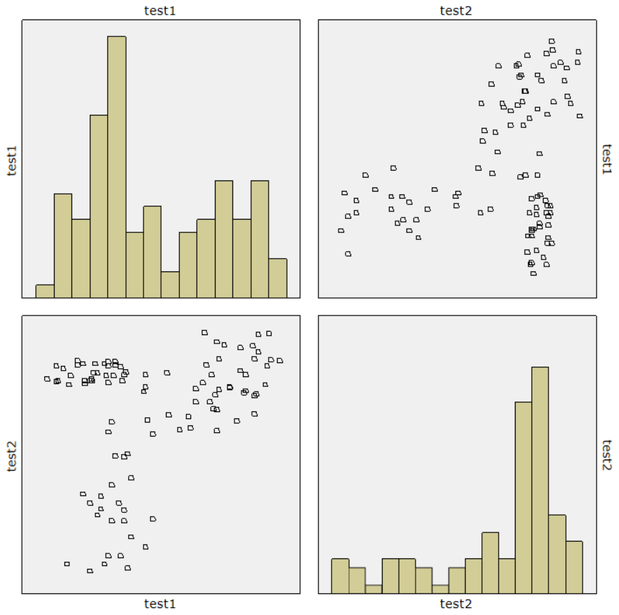

Figure 6: User satisfaction before using the adaptive environment to design strategy.

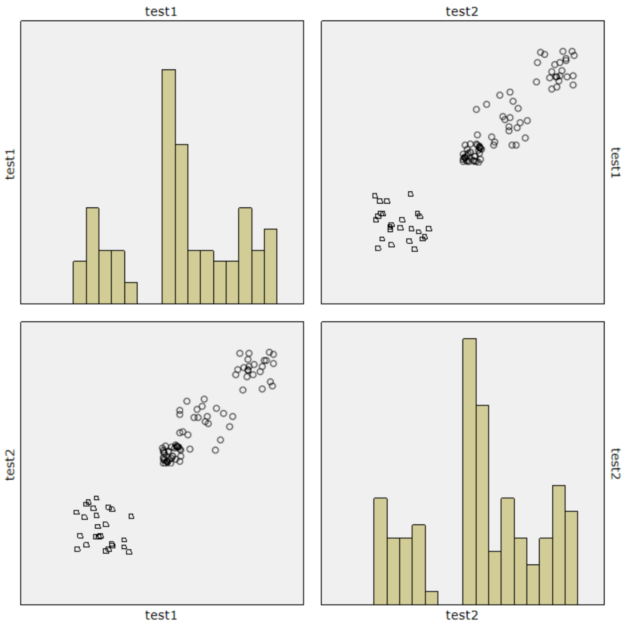

Figure 7: User satisfaction after using the adaptive environment to design strategies.

5.2 Direction and suggestions for strategy optimization

After analyzing and discussing the experimental results, several optimization strategies and suggestions can be proposed. Firstly, optimizing the algorithm and parameter settings of the RL model can enhance its learning capabilities and generalization performance. Secondly, refining the implementation process and decision-making mechanisms of the adaptive environment design strategy can improve its adaptability and intelligence. Additionally, incorporating other cutting-edge machine learning and artificial intelligence technologies can aid in realizing and optimizing these adaptive environment design strategies.

6 CONCLUSIONS

This study delves into the intricacies of adaptive environment design within CAD settings and introduces an innovative strategy grounded in RL. It delves into the challenges inherent in CAD environment design, emphasizing the need for adaptability and exploring RL's potential. Successfully, an RL model is established, outlining the implementation process for the adaptive design strategy and validating its efficacy through simulation experiments.

Our key accomplishments are twofold: first, we've crafted a self-adaptive environment design strategy powered by RL. This strategy dynamically adjusts CAD environment parameters and configurations in response to evolving design tasks, offering optimal design support. Its effectiveness is substantiated through simulation experiments, demonstrating significant improvements in design efficiency and quality. Second, our approach paves the way for CAD environment intelligence, introducing new perspectives and methodologies that catalyze technical advancements in related domains.

The study's contributions are multifaceted: it overcomes the constraints of traditional CAD environments, enhancing design process flexibility and intelligence. Furthermore, by integrating RL into adaptive environment design, we've broadened RL's applicability in engineering design. Ultimately, our findings offer valuable insights to scholars and practitioners, propelling CAD environmental design technology forward. Looking ahead, we remain attuned to emerging technologies and methodologies, poised to forge further breakthroughs and innovations in our future research endeavours.

7 ACKNOWLEDGEMENT

This work was supported by Hainan Provincial Natural Science Foundation of China, Study on the strategy of cluster protection and reuse of Hainan traditional settlement heritage from the perspective of rural revitalization (No. 721RC604).

Xiaolong Chen, https://orcid.org/0000-0002-2767-7017
Lin Chen, https://orcid.org/0000-0002-8933-7798

REFERENCES

[1] Devo, A.; Mezzetti, G.; Costante, G.; Fravolini, M.-L.; Valigi, P.: Towards generalization in target-driven visual navigation by using deep reinforcement learning, IEEE Transactions on Robotics, 36(5), 2020, 1546-1561. https://doi.org/10.1109/TRO.2020.2994002
[2] Erdolu, E.: Lines, triangles, and nets: A framework for designing input technologies and interaction techniques for computer-aided design, International Journal of Architectural Computing, 17(4), 2019, 357-381. https://doi.org/10.1177/1478077119887360
[3] Felbrich, B.; Schork, T.; Menges, A.: Autonomous robotic additive manufacturing through distributed model-free deep reinforcement learning in computational design environments, Construction Robotics, 6(1), 2022, 15-37. https://doi.org/10.1007/s41693-022-00069-0

[4] Halskov, K.; Lundqvist, C.: Filtering and informing the design space: towards design-space thinking, ACM Transactions on Computer-Human Interaction, 28(1), 2021, 1-28. https://doi.org/10.1145/3434462

[5] Hou, Z.; Li, Z.; Hsu, C.; Zhang, K.; Xu, J.: Fuzzy logic-driven variable time-scale prediction-based reinforcement learning for robotic multiple peg-in-hole assembly, IEEE Transactions on Automation Science and Engineering, 19(1), 2020, 218-229. https://doi.org/10.1109/TASE.2020.3024725

[6] Hu, H.; Zhang, K.; Tan, A.-H.; Ruan, M.; Agia, C.; Nejat, G.: A sim-to-real pipeline for deep reinforcement learning for autonomous robot navigation in cluttered rough terrain, IEEE Robotics and Automation Letters, 6(4), 2021, 6569-6576. https://doi.org/10.1109/LRA.2021.3093551

[7] Lee, X.-Y.; Balu, A.; Stoecklein, D.; Ganapathysubramanian, B.; Sarkar, S.: A case study of deep reinforcement learning for engineering design: Application to microfluidic devices for flow sculpting, Journal of Mechanical Design, 141(11), 2019, 111401. https://doi.org/10.1115/1.4044397

[8] Li, K.; Zhang, Q.; Jia, J.-Y.: CEB-collaboratively multi-granularity interest scheduling algorithm for loading large webbim scene, Journal of Computer-Aided Design & Computer Graphics, 33(9), 2021, 1388-1397. https://doi.org/10.3724/SP.J.1089.2021.18718

[9] Liu, K.; Yang, H.; Ma, Y.: Adversarial perturbation attacks on ml-based CAD: a case study on CNN-based lithographic hotspot detection, ACM Transactions on Design Automation of Electronic Systems, 25(5), 2020, 1-31. https://doi.org/10.1145/3408288

[10] Liu, Z.; Liu, Q.; Wang, L.; Xu, W.; Zhou, Z.: Task-level decision-making for dynamic and stochastic human-robot collaboration based on dual agents deep reinforcement learning, The International Journal of Advanced Manufacturing Technology, 115(11-12), 2021, 3533-3552. https://doi.org/10.1007/s00170-021-07265-2

[11] Rahul, M.-R.; Chiddarwar, S.-S.: Deep reinforcement learning with inverse jacobian based model-free path planning for deburring in complex industrial environment, Journal of Intelligent & Robotic Systems, 110(1), 2024, 4. https://doi.org/10.1007/s10846-023-02030-x

[12] Shang, W.; Li, Q.; Qin, Z.; Yu, Y.; Meng, Y.; Ye, J.: Partially observable environment estimation with uplift inference for reinforcement learning based recommendation, Machine Learning, 110(9), 2021, 2603-2640. https://doi.org/10.1007/s10994-021-05969-w

[13] Shi, H.; Shi, L.; Xu, M.; Hwang, K.-S.: End-to-end navigation strategy with deep reinforcement learning for mobile robots, IEEE Transactions on Industrial Informatics, 16(4), 2019, 2393-2402. https://doi.org/10.1109/TII.2019.2936167

[14] Xiao, X.; Liu, B.; Warnell, G.; Stone, P.: Motion planning and control for mobile robot navigation using machine learning: a survey, Autonomous Robots, 46(5), 2022, 569-597. https://doi.org/10.1007/s10514-022-10039-8

[15] Zhou, J.-J.; Phadnis, V.; Olechowski, A.: Analysis of designer emotions in collaborative and traditional computer-aided design, Journal of Mechanical Design, 143(2), 2020, 1-18. https://doi.org/10.1115/1.4047685