DDPG-Based Reinforcement Learning Framework for Action Path Optimization in New Media Animation Creation

Chunyu Zou [1], Peifeng Xie [2*]
[1]School of Management, University of Suffolk, Boston, Massachusetts 02108, USA
[2]School of Economy and Finance, South China University of Technology, Guangzhou 510006, China
E-mail: PeifengXiee@outlook.com
[*]Corresponding author

Keywords: optimization algorithm, reinforcement learning, new media animation, action path optimization

Received: April 13, 2025

With the rapid development of the new media industry, animation creation occupies an important position in the fields of modern film and television, advertising and games. Traditional animation creation process mostly relies on manual operation, which is inefficient and flexible, especially in the design and optimization of action path. In order to solve this problem, this study proposes an intelligent control and optimization scheme using the Deep Deterministic Policy Gradient (DDPG) algorithm to optimize the action path in new media animation creation. The method constructs a reward function considering path fluency, precision, creation cost, and user preference, and applies a continuous control strategy within a reinforcement learning framework. We collected 1000 animation scene data samples and compared the proposed method against traditional optimization techniques including Genetic Algorithm (GA), Particle Swarm Optimization (PSO), and Simulated Annealing (SA). Experimental results show that our method reduces the action path error (MSE) from 0.082 to 0.045 (a 45.1% improvement), increases fluency from 0.87 to 0.97 (a 11.5% increase), and reduces optimization time by 55% compared with GA. The DDPG-based approach also demonstrates faster convergence and better stability. These findings confirm the effectiveness and efficiency of reinforcement learning in enhancing intelligent animation production. The research results of this paper provide a new idea and method for new media animation creation, which can greatly improve the automation degree and quality of animation production, and provide theoretical support and practical guidance for the intelligent animation creation in the future.

Povzetek: Predlagana metoda z ojačitvenim učenjem (DDPG) optimizira poti animacijskih gibov ter v primerjavi z GA/PSO/SA zmanjša napako (MSE 0,082 → 0,045), poveča gladkost (0,87 → 0,97) in skrajša čas optimizacije za 55 %.

1 Introduction

With the rapid progress of digital and intelligent technology, the traditional animation creation mode faces increasingly severe challenges. With the prosperity of the new media field, animation creation is no longer restricted by the conventional restrictions of 2D and 3D technologies, and the demand for cross-platform and interactive content has significantly improved the complexity of animation creation. Especially in the creative process, the design and optimization of the movement trajectory of animation characters are always the core elements that affect the animation effect and creation speed. Animation creation depends on manual design and adjustment, which takes a long time and lacks flexibility. It is even more challenging to do so when facing large-scale animation output. This need is particularly urgent given the explosive growth in demand for digital content in film, television, advertising, virtual reality, and interactive entertainment. Production studios are under increasing pressure to deliver high-quality animations within tight timelines and budget constraints. Manual design workflows, while artistically valuable, are no longer sufficient to meet the volume, speed, and precision required. As a result, the integration of artificial intelligence techniques, especially reinforcement learning-based automation, is emerging as a transformative solution. This study directly responds to this trend by proposing a reinforcement learning framework tailored for real-world animation production challenges [1].

Optimization algorithms and Reinforcement Learning (RL) have made remarkable progress in artificial intelligence in recent years. Optimization algorithms can find the best solution under preset constraints. At the same time, reinforcement learning gradually improves the decision-making process through interactive learning with the environment, especially showing unique performance when dealing with high-dimensional and nonlinear problems. Building upon recent advancements in artificial intelligence, this study adopts a reinforcement learning strategy based on the DDPG algorithm to address the path optimization problem in new media animation creation. Rather than combining multiple optimization techniques, the focus is placed on evaluating the standalone effectiveness of

DDPG, which is applied to the intelligent control and optimization of action path in new media animation creation. This strategy achieves the adaptive optimization of the action path innovatively, improving the animation production speed and ensuring a natural and smooth animation effect [2].

The innovation of this research lies in integrating reinforcement learning and optimization algorithms and constructing an intelligent control system that can adaptively regulate the animation trajectory. The system automatically optimizes the action path parameters with the help of the reward function and learns the best action design mode. Compared with traditional manual design, the reinforcement learning approach can create a natural and plot-appropriate action path in a shorter time. Experiments have proved that this approach not only ensures the quality of creation but also significantly speeds up the animation production speed, reduces the cost, opens up a new perspective, and provides technical support for the intelligent evolution of the new media animation industry [3].

This paper aims to explore the integration and profound application of optimization algorithms and reinforcement learning and open up an intelligent path optimization method for new media animation creation. With the support of abundant experimental data, this paper shows the practical effect of this method in action path planning and confirms its potential to promote creative efficiency, reduce labor costs, and improve animation quality. In addition, the research results also contribute valuable reference and practical experience to promoting the automation and intelligence process of new media animation creation in the future [4].

This study identified three main research objectives: to evaluate whether the DDPG model can achieve a mean square error (MSE) of less than 0.05 in animation path planning, compare its improvement effect on path fluency compared to traditional algorithms (GA, PSO, SA), and verify its optimization efficiency in large-scale animation production. Three hypotheses are proposed accordingly: H1 believes that DDPG is superior to traditional algorithms in path error, H2 believes that its path fluency score is higher, and H3 expects DDPG to significantly outperform GA and SA in optimization time while maintaining or improving output quality.

2 Theoretical basis and related research

2.1 Overview of reinforcement learning algorithms

Reinforcement learning is one of the three classic branches of machine learning. Unlike the other two types of branches, reinforcement learning algorithms do not rely on external data but on the feedback of the environment to actions, spontaneously exploring and learning the optimal model [5].

The concept of reinforcement learning originates from behavioral psychology. In specific scenarios, creatures react closely or distantly to the behavior they receive. This psychological mechanism prompts organisms to get rewards or be punished in the environment according to different behaviors and then gradually learn adaptive strategies and evolve in a direction that is beneficial to themselves [6, 7].

After a long accumulation period, reinforcement learning has gradually been integrated into many engineering projects. The core purpose is to explain and solve the problem of what strategies agents adopt to maximize benefits or achieve specific goals when interacting with the environment. In a given scenario, the agent accumulates knowledge through trial-and-error learning to select the optimal action to win the greatest return. Specifically, the agent takes actions in an unknown environment, continuously accumulates experience according to the feedback of the environment, constantly improves its decision-making process, and finally establishes a behavioral decision-making system that can obtain higher rewards [8].

In the field of reinforcement learning, the Markov Decision Process (MDP) is a common model, and almost all reinforcement learning problems can be mathematically transformed into MDP. Because of its feasibility in practical application, this model has become the most widely adopted form to define reinforcement learning problems and constitutes the core cornerstone of reinforcement learning algorithms [9].

As the field of reinforcement learning continues to evolve, many complex problems increasingly rely on its solutions. However, such issues often lack suitable models, making implementing model-based methods difficult. Faced with the lack of model information, the state transition probability P is in an unknown state, and the subsequent state prediction becomes difficult. It is difficult to derive the state value and state-action value function directly from the Bellman equation to obtain the optimal strategy. This situation forces the agent to interact directly with the surrounding environment and continuously learn strategy from the interactive experience to overcome the complex problems that traditional planning methods can't overcome [10].

2.2 Overview of action path of new media animation creation under reinforcement learning of optimization algorithm

In creating new media animation, designing and optimizing the action path is the key to the natural and smooth animation effect. In traditional animation production, animators need to personally adjust the subtleties of character movements, such as character walking posture, posture conversion, rate control, etc. [11]. Faced with the increasingly complex demand of animation production, manual design has made it challenging to meet the needs of efficient creation. Especially when making large-scale and high-quality animations, manual adjustment is time-consuming, laborious, and error-prone. Therefore, optimizing the action path and improving the animation production automation level are essential topics in animation creation [12].

Optimization algorithm is a skill that uses mathematical means to find the best solution and has been widely used in many fields. New media animation production can help animation teams quickly produce up-to-standard action trajectories under established restrictions. Through the optimization algorithm, the labor of manual adjustment can be significantly reduced, and the creation speed can be improved. Under the coordination of multiple parameters, a more natural and harmonious animation effect can be achieved. Standard methods such as genetic algorithms, particle swarm optimization, and simulated annealing algorithms can provide efficient and feasible solutions to path planning, action adjustment, and other problems [13].

As an intelligent algorithm, reinforcement learning continuously optimizes the decision-making process when interacting with the environment. It has now become a powerful assistant in dealing with high-dimensional complex problems. The field of animation creation relies on the set reward mechanism to guide the model in learning and adjusting the action path, striving to present the best effect. Unlike traditional optimization algorithms, reinforcement learning does not need to preset all possible action paths but relies on continuous training and feedback to improve the action path generation strategy [14, 15]. This model can independently explore the action path that best matches the specific scene and needs, significantly improving animation creation's intelligence.

Combining the essence of optimization algorithms and reinforcement learning, this research innovatively proposes a methodological framework to optimize the action path in new media animation creation. The architecture dynamically adjusts the action path parameters with the help of a reinforcement learning mechanism. It incorporates optimization algorithms to precisely carve the details to ensure the action's fidelity, coherence, and plot fit. This intelligent control method accelerates the animation creation process and presents a more vivid and expressive visual feast for the audience. To sum up, the research results have contributed a cutting-edge intelligent control strategy to the new media animation field, laying the foundation for the future automation and intelligence of animation production [16]. The comparison of animation path optimization methods is shown in Table 1.

Table 1: Comparison of animation path optimization methods

Method Type	Representative Techniques	Evaluation Metrics
Heuristic-based	GA, PSO, Simulated Annealing	Path error
GAN-based	GANs with motion priors	Visual realism
Classical planning	A*, RRT, Dijkstra	Path length, smoothness

3 Establishment of new media animation creation model based on optimization algorithm reinforcement learning

3.1 Formulation of dynamic path optimization model based on reinforcement learning

Aiming at the dynamic path optimization model and final optimization objective function of new media animation creation, it is necessary to ensure the accuracy and smoothness of the creation process's action path and consider the creation efficiency and resource consumption. Against this background, formulating a dynamic path optimization algorithm mechanism is critical [17, 18]. The flow of the dynamic path optimization algorithm for new media animation creation is shown in Figure 1.

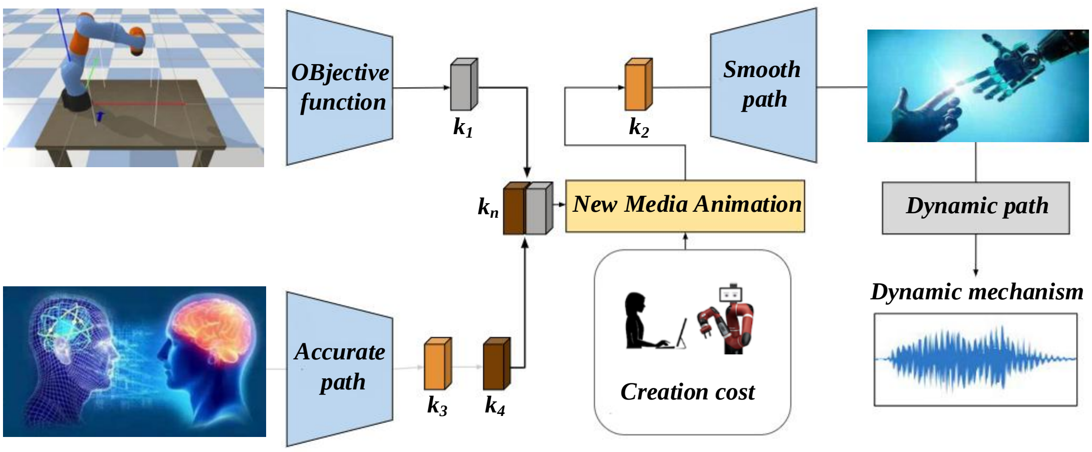

Figure 1: Dynamic path optimization algorithm process for new media animation creation

New media animation creation: Dynamic path optimization strategy construction, according to the creation cost, action path accuracy, time demand, and user smooth and natural preference adjustment, striving for a win-win situation between the animation team and users [19]. Facing the changeable new media animation environment, the existing schemes and algorithms have poor results. Reinforcement learning is progressing rapidly, exploring new directions for dynamic path optimization [20]. The objective function formula of creation cost and demand optimization is shown in (1). The various indicators in equation (1) quantify the key factors of animation path optimization, including total creation cost, path accuracy error, path smoothness, and user satisfaction. This reward design reflects the methodological integration of traditional optimization theory into the reinforcement learning framework. The key optimization goals—path accuracy, fluency, cost minimization, and user preference alignment — are incorporated as components of the reward function. This allows the RL agent to optimize the action policy while being guided by interpretable, optimization-derived objectives. In particular, the DDPG algorithm benefits from this integration by learning a deterministic policy that maximizes cumulative reward, effectively balancing exploration with precision, and ensuring alignment with classic path planning constraints while leveraging the adaptive learning capacity of deep RL.

$$Objectivefunction = \lambda_1 \cdot C_{cost} + \lambda_2 \cdot D_{precision} + \lambda_3 \cdot D_{flow} - \lambda_4 \cdot P_{user} \quad (1)$$

Among them, C_{cost} represents the creation cost, $D_{precision}$ represents the accuracy of the action path, D_{flow} represents the fluency of the action path, P_{user} represents the user's preference for fluency and naturalness, and $\lambda_1, \lambda_2, \lambda_3, \lambda_4$ represent the weight factors.

Because of the complex composition and large scale of new media animation creation, accompanied by highly creative uncertainty, it can be regarded as the research of exploring dynamic path optimization strategies in unknown fields [21]. According to the reinforcement learning theory, the agent accumulates creative experience through continuous exploration and interaction with the surrounding environment, understands the relationship between animation path and innovative action, and then uses path optimization means to adjust the fluency and accuracy of animation creation to achieve the established dynamic path optimization mechanism goal. The animation path optimization objective function formula is shown in (2).

$$Objectivefunction = \alpha_1 \cdot L_{flow} + \alpha_2 \cdot L_{precision} - \alpha_3 \cdot C_{cost} \quad (2)$$

Among them, L_{flow} represents path fluency, $L_{precision}$ represents path accuracy, C_{cost} represents creation cost, and α_1, α_2, and α_3 represent weight coefficients. The reinforcement learning path optimization reward letter formula is shown in (3).

$$R(a_t) = \gamma \cdot E[V(s_{t+1})] - V(s_t) + r_t \quad (3)$$

Where $R(a_t)$ represents the reward obtained when the action at is performed at time t, γ represents the discount factor, $V(s_t)$ represents the value of the current state st, $E[V(s_{t+1})]$ represents the expected value of the following state s_{t+1}, and r_t represents the immediate reward. Within the reinforcement learning framework, the animation creation team is regarded as an agent, its creation environment is the learning situation, and the action is defined as the path optimization strategy. This action changes state variables such as accuracy, smoothness, and audience experience of the animation path, Creative efficiency, and audience satisfaction as reward signals for learning feedback [22]. This research aims to maximize the reward through policy actions, model this as a Markov decision process, and use the deep deterministic policy gradient algorithm to solve it. The formula of the Markov decision process model of the path optimization strategy is shown in (4). Where $P(s_t, a_t)$ represents the probability of the combination of the current state s_t and the current action a_t, $P(s_{t+1}|s_t, a_t)$ represents the state transition probability, and $\pi(a_t|s_t)$ represents the policy function.

$$P(s_t, a_t) = P(s_{t+1} \mid s_t, a_t) \cdot \pi(a_t \mid s_t) \quad (4)$$

3.2 Deep deterministic policy gradient algorithm

To address the challenges of continuous and high-dimensional action path optimization in new media animation, this study adopts the DDPG algorithm as the core method within the reinforcement learning framework. DDPG is a specific type of deep reinforcement learning algorithm developed by Google DeepMind, designed for environments with continuous action spaces. Compared with discrete-action methods like DQN, DDPG provides more precise control and better scalability for complex motion path generation tasks, which uses discrete action space, DDPG performs excellently in continuous action domains and high-dimensional situations. The objective function formula of the DDPG algorithm is shown in (5).

$$J(\theta) = E_{s_t \sim \rho^{\beta}} \left[r_t + \gamma Q'(s_{t+1}, a_{t+1}) - Q(s_t, a_t) \right]^2 \quad (5)$$

Where $J(\theta)$ represents the optimization objective function, s_t represents the current state, a_t represents the action selected in state s_t, r_t represents the instant reward, γ represents the discount factor, $Q(s_t, a_t)$ represents the value of taking an action in this state, and $Q'(s_{t+1}, a_{t+1})$ represents the target network calculation. For the complex path optimization model of new media animation creation, if a stochastic strategy is adopted, such as the DON algorithm, to evaluate the probability of each action one by one, the computational complexity will be extremely high, which significantly hinders the algorithm efficiency and optimal path selection [23]. In contrast, the DDPG algorithm uses a deterministic strategy, approximates the strategy function μ with the help of a neural network, and directly obtains the unique optimal path strategy, effectively eliminating the disturbance of probabilistic factors. The pseudocode for DDPG animation path optimization is shown in Table 2.

Table 2: Pseudo code for DDPG animation path optimization

Initialize actor μ(s|θμ), critic Q(s,a|θQ), target networks μ', Q'
Initialize replay buffer B and noise process N
for each episode do
Initialize state s_0 from animation scene
for each step t do
Select action $a_t = \mu(s) + N_t$
Execute a_t, observe reward r_t and next state s_{t+1}
Store (s_t, a_t, r_t, s_{t+1}) in B
Sample minibatch from B
Compute target: $y = r + \gamma \cdot Q'(s', \mu'(s'))$
Update critic: minimize $(Q(s,a) - y)^2$
Update actor via policy gradient
Soft-update target networks
end for
end for

The DDPG deterministic strategy formula is shown in (6).

$$a_t = \mu(s_t \mid \theta_\mu) \quad (6)$$

Where a_t represents the action selected at time step t, $\mu(s_t \mid \theta_\mu)$ represents the optimal action, and θ_μ represents the parameters of the policy network. The DDPG policy update formula is shown in (7).

$$\theta_\mu = \theta_\mu + \alpha \cdot \nabla_{\theta_\mu} J(\theta_\mu) \quad (7)$$

Where θ_μ represents the parameters of the policy network, α represents the learning rate, and $\nabla_{\theta\mu} J(\theta_\mu)$ represents the gradient of the policy network parameters. At the same time, as an algorithm in the field of reinforcement learning, it needs to deepen the reinforcement of exploration and development during training to broaden the search space and explore better strategies. Therefore, in the action selection stage, the DDPG algorithm integrates random processes into the deterministic strategy architecture, samples actions with random noise, and applies them to the environment. The summary of DDPG hyperparameters is shown in Table 3.

Table 3: Summary of DDPG hyperparameters

Parameter	Value	Description
Learning rate (α)	0.0003	Actor and critic network learning rate
Discount factor (γ)	0.99	Future reward discount
Soft update rate (τ)	0.005	Target network soft update coefficient

This study constructs a reinforcement learning model for new media animation creation based on dynamic path optimization and DDPG algorithm. This model adopts a continuous control actor critic architecture, with inputs including character position, trajectory, environmental constraints, and semantic information, and outputs as path adjustment actions (such as curvature, rhythm, direction). The reward function comprehensively considers path accuracy, smoothness, cost, and user preferences. Through experience replay and soft update mechanisms, the model continuously optimizes strategies in interaction with animation samples, automatically generates action trajectories that meet performance and aesthetic requirements, and constructs an intelligent, efficient, and high-quality core mechanism for new media animation creation.

This section constructs a reinforcement learning model based on DDPG for dynamic path optimization in new media animation. The model adopts a continuous control actor critic architecture, with the state space containing character position, velocity, trajectory history, environmental constraints, and semantic labels, and the action space defining path adjustment parameters such as direction, velocity, curvature, and rhythm. The actor network adjusts the output path action, and the critic network evaluates its long-term return. The reward function integrates four indicators: path accuracy, smoothness, animation cost, and user preference, all of which are normalized. Through experience replay and soft target network updates, the model achieves stable training and gradually generates high-quality and resource efficient animation paths, enhancing the intelligence and practicality of new media animation creation.

DDPG follows the experiential playback and Q-target network mechanism in DQN and improves it on this basis. Aiming at the instability of single Q network training and the deviation of value function estimation, the Actor-Critic architecture is used to construct the Actor strategy network and the Critic value network, and the Online network and the Target network are set up respectively. Among them, the policy network is responsible for fitting and updating the policy function μ (parameter θ_μ). In contrast, the value network is used to evaluate the advantages and disadvantages of the current policy (parameter θ_Q). The parameter update strategy follows the soft update mechanism to ensure the stability and progressiveness of the training process. The Actor policy network update formula is shown in (8).

$$\theta_\mu \leftarrow \theta_\mu + \alpha \cdot \nabla_{\theta_\mu} J(\theta_\mu) \quad (8)$$

Where θ_μ represents the weight and bias of the neural network, α represents the learning rate, $\nabla_{\theta\mu} J(\theta_\mu)$ represents the gradient of the policy network parameters, and $J(\theta_\mu)$ represents the objective function of the policy network. The core of reinforcement learning lies in the agent's interactive learning with the environment to derive the optimal action strategy to maximize the reward. DDPG algorithm also follows this concept [24, 25].

Equation (9) shows the action reward obtained by the agent according to the Markov decision process, which is usually characterized by the state action-value function.

$$\theta_\mu = arg\ \underset{\theta_\mu}{max} E\left[\sum_{t=0}^{T}\gamma^t r_t\right] \quad (9)$$

Where θ_μ represents the parameters of the policy network, γ represents the discount factor, r_t represents the immediate reward at the current time step t, T represents the termination time step, and E represents the expected value. However, discrete actions and state spaces, such as formula (10), aim only at the DQN and Q-learning algorithms. DDPG uses its unique deep neural network to optimize the optimization strategy and enhance the stability and convergence ability of the algorithm [26]. The soft update formula of the target network is shown in (10).

$$\theta_Q^{'} = \tau\theta_Q + (1-\tau)\theta_Q^{'} \quad (11)$$

Where θ'_Q represents the parameters of the target network, θ_Q represents the parameters of the current network, τ represents the soft update rate, and $(1-\tau)$ represents the supplementary part. According to the above steps, the flow chart of the DDPG algorithm is shown in Figure 2.

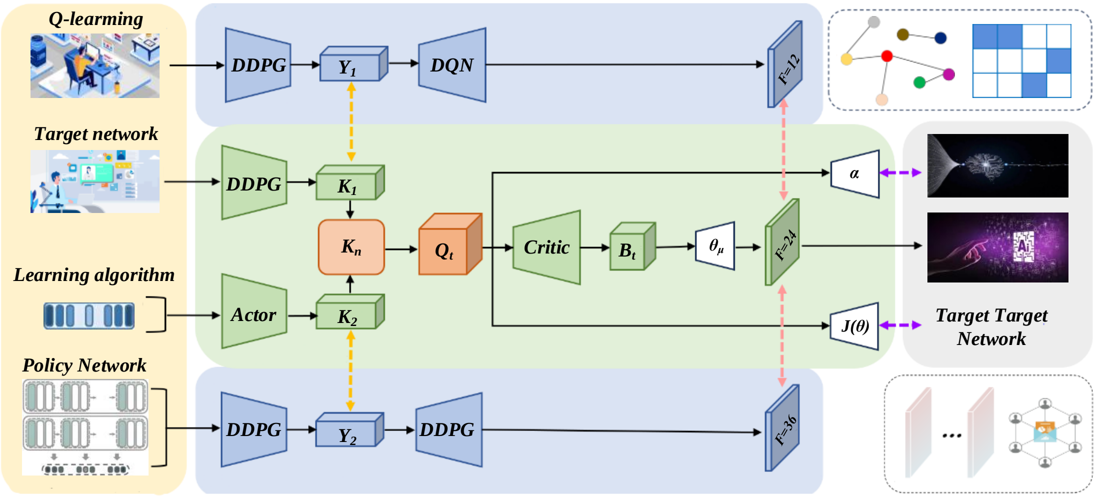

Figure 2: Flowchart of DDPG algorithm

4 Experimental results and analysis

All experiments were conducted on a workstation with the following hardware and software configuration: an Intel Core i9-12900K CPU (16 cores), 64 GB DDR5 RAM, and an NVIDIA GeForce RTX 3080 GPU with 10 GB VRAM. The operating system was Ubuntu 22.04 LTS. The reinforcement learning models were implemented using Python 3.9.13. The DDPG algorithm and supporting neural networks were developed with TensorFlow 2.9.1 and PyTorch 1.13.0. CUDA version 11.6 and cuDNN 8.4 were used for GPU acceleration. The experiments were managed and reproducibility ensured via fixed random seeds, isolated conda environments, and logging via TensorBoard. This study uses three core indicators to evaluate the quality and effectiveness of action path optimization: path error (MSE) measures the average Euclidean distance deviation between the predicted path and the reference trajectory, with lower values indicating higher position accuracy; The smoothness of motion is evaluated by the consistency between the fitted curve and the actual motion curve. The closer the R^2 is to 1.0, the more natural the animation is; The quality compliance rate represents the proportion of generated samples that meet the requirements of MSE<0.05, R^2>0.90, and frame continuity, reflecting system stability and generation consistency. All indicators are calculated through automated scripts to ensure objectivity and consistency in experimental comparisons [27, 28].

This study defines path error as the mean square error (MSE) of the Euclidean distance between the predicted trajectory and the reference trajectory, which is calculated by frame-by-frame position vector differences to reflect the actual deviation of the action path in physical space. This distance based MSE evaluation method is suitable for continuous motion path analysis in animation, and can effectively measure position accuracy and motion smoothness, making it a reasonable choice for evaluating path accuracy. The experimental results of the optimization algorithm comparison are shown in Table 4.

Table 4: Optimization algorithm comparison experimental results

Optimization algorithm	Action path error (MSE)	Movement smoothness (R^2 value)	Optimization time (seconds)
Genetic algorithm	0.065	0.92	150
Particle swarm optimization	0.052	0.95	120
Simulated annealing	0.072	0.88	180
Reinforcement learning	0.045	0.97	90

Table 4 shows that the DDPG reinforcement learning method outperforms traditional algorithms in terms of path error, motion smoothness, and optimization time. The minimum MSE is 0.045, and the path accuracy is the highest; The R^2 value reaches 0.97, indicating the smoothest motion; The optimization time is only 90 seconds, significantly faster than GA (150 seconds), PSO (120 seconds), and SA (180 seconds). This result validates the comprehensive advantages of DDPG in improving animation path quality and optimization efficiency, especially for large-scale animation production scenes [29]. The comparison result shows that the reinforcement learning algorithm stands out among all optimization algorithms. Its motion path error (MSE) is as low as 0.045, and its fluency (R^2 value) is as high as 0.97, demonstrating the natural smoothness of the motion path generated by this method. In addition, the optimization time required for reinforcement learning is only 90 seconds. Compared with the genetic algorithm (150 seconds), particle swarm optimization (120 seconds), and simulated annealing (180 seconds), the time advantage is significant. Overall, reinforcement learning excels in the accuracy and efficiency of path optimization, surpassing other optimization algorithms.

This study uses multidimensional indicators such as path error, path fluency, optimization time, and error improvement rate to systematically evaluate the effectiveness and efficiency of various algorithms in animation path generation. The relevant data is visualized through tables and diagrams. The evaluation results indicate that the DDPG method outperforms traditional algorithms in terms of accuracy, smoothness, optimization speed, and improvement range, fully verifying its practicality and advantages in various animation creation scenarios.

To compare the accuracy performance of different optimization algorithms and reinforcement learning in animation path optimization, especially the path error, this paper compares the optimization algorithm and reinforcement learning in path error. The results are shown in Figure 3.

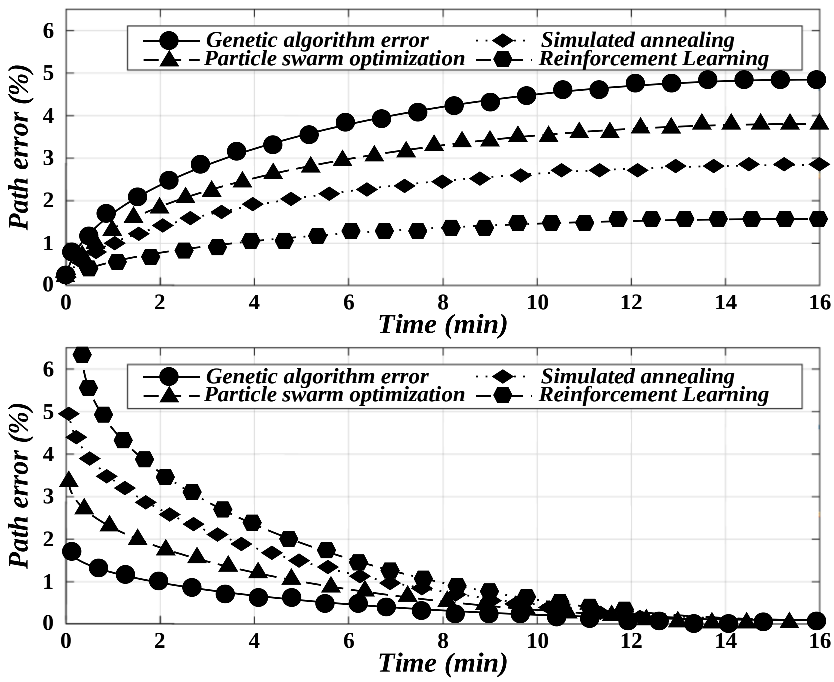

Figure 3: Comparison of path error between optimization algorithm and reinforcement learning

The experiment records the path error data of various algorithms in the process of multi-round iteration in detail. It presents the differences in error control among various algorithms intuitively using charts. Finally, the research compares the advantages and disadvantages of each method with the help of error analysis, which provides solid data support for selecting an intelligent path control algorithm [30]. The diagram reflects the specific performance of the four algorithms in terms of path error (MSE): the error of the genetic algorithm is 0.065, the error of the particle swarm optimization (PSO) algorithm is 0.052, the error of the simulated annealing algorithm is 0.072, and the error of reinforcement learning algorithm is only 0.045. Reinforcement learning takes the lead in the accuracy of path optimization. Reinforcement learning relies on continuous strategy adjustment to precisely control path details, effectively reduce errors, and ensure the fluency of animation. Although the performance of the genetic algorithm and particle swarm optimization algorithm is acceptable, the error is still significant compared with reinforcement learning. This highlights the advantages of reinforcement learning when dealing with complex path-planning tasks, especially in new media animation creation, which can significantly improve animation's natural fluency and accuracy.

To compare the performance of different optimization algorithms and reinforcement learning in path fluency (R^2 value) and to verify whether reinforcement learning can generate a smoother and smoother animation path, this paper compares different optimization algorithms in path fluency, and the results are shown in Figure 4.

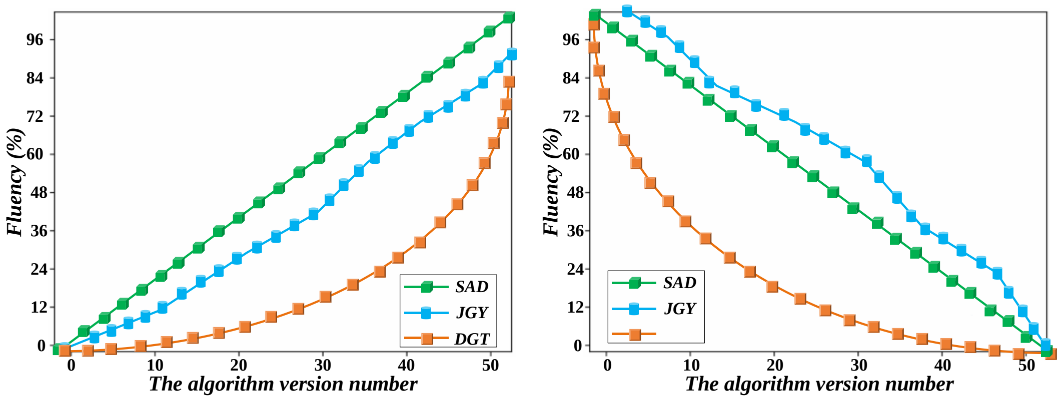

Figure 4: Comparison of path fluency among different optimization algorithms

Figure 4 shows the comparison results of the four methods in path fluency. SAD represents simulated annealing, JGY represents genetic algorithm, and DGT represents particle swarm optimization. The experiment quantifies the fluency of the path generated by each algorithm by evaluating the path's curvature, turning angle, and speed change. The results are shown in the form of charts, and the differences in path smoothness among different algorithms are compared, providing data support for path optimization and a reference for selecting intelligent path control methods in animation creation. The path smoothness score, measured via curvature and acceleration continuity, for reinforcement learning is 0.97, significantly higher than the other three algorithms. The fluency of the genetic algorithm is 0.92, particle swarm optimization is 0.95, and simulated annealing is 0.88. Reinforcement learning not only excels in path error but also has significant advantages in fluency. The higher the fluency of the animation path, the more natural the audience experience is, and the more realistic the movement feel of the animation. This shows that through self-optimization and feedback mechanisms, reinforcement learning can generate paths more in line with the laws of human movements and improve animation's expressiveness and viewing effect.

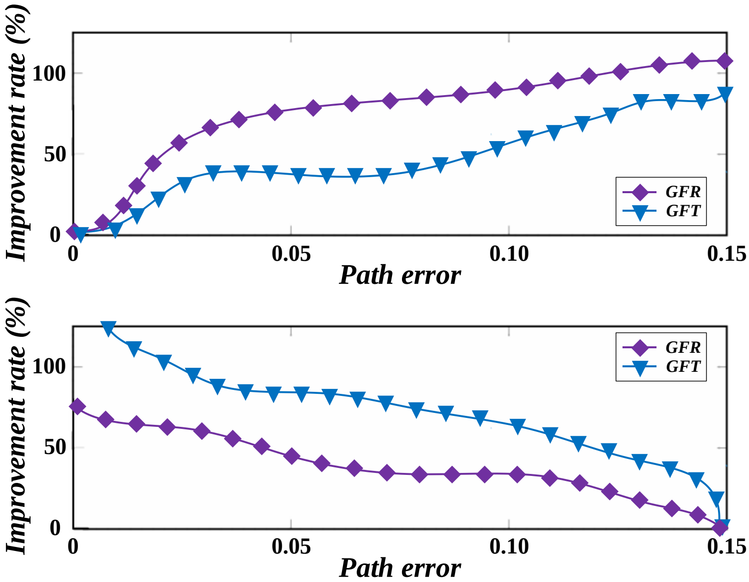

Figure 5: Comparison of error changes before and after path optimization

To show the change of error before and after path optimization. By comparing the changes in path errors before and after optimization, the improvement effect of the optimization algorithm and reinforcement learning on path accuracy is verified. This paper compares the changes in errors before and after path optimization, and the results are shown in Figure 5. GFR represents global frame level reduction, and GFT represents global frame level total.

Fluency measurement is quantitatively calculated using coefficient of determination, which is a statistical measure that indicates the degree to which the fitted curve models the actual path dynamics. Specifically, the system performs second-order polynomial regression on the velocity and acceleration curves of each path segment. Then calculate the R^2 score between the smooth curve and the original time series data. The higher the R^2 value, the smoother and more natural the transition between motion states, reflecting the better smoothness of the animation. This method allows for robust and interpretable measurement of motion smoothness, which is sensitive to abrupt changes and can be generalized across different types of animations.

This study constructed a reinforcement learning environment for DDPG training based on Markov Decision Process (MDP). The state space includes the current position, velocity, trajectory of the character in the past 5 frames, environmental constraints, and semantic labels, all of which are normalized and input into the network; The action space consists of four continuous control variables: direction adjustment, speed scaling, curvature control, and rhythm adjustment, which are used to guide path generation. The reward function consists of four parts: path accuracy, smoothness, generation cost, and user preference. The weights are λ_1=0.4, λ_2=0.3, λ_3=0.2, and λ_4=0.1, respectively. All components are normalized to [0,1]. Each step reward is used for updating strategies and value networks, combined with experience replay and soft update mechanisms, to achieve adaptive optimization of high-quality animation paths.

GFR and GFT optimization strategies exhibit significant performance differences in different ranges of path errors. The above figure shows that as the path error increases from 0 to 0.15, the optimization improvement rate of GFR steadily increases, rising from about 5% to over 110%; In contrast, the growth rate of GFT is relatively slow, increasing from an initial 0% to about 95%. In the figure below, the optimized path error reduction rate shows a decreasing trend with increasing initial error. GFT has the highest improvement rate of about 120% in the low error range (<0.02), but decreases to nearly 0% as the error increases; The overall performance of GFR is more stable, fluctuating between 40% and 80% throughout the entire error interval. This result indicates that GFR has stronger path optimization robustness in medium to high error scenarios, while GFT has higher instantaneous response optimization capability when the initial error is small.

Table 5: Reinforcement learning training parameters and effect evaluation

Parameter	Value	Target value	Actual effect
Learning rate	0.0003	0.0003	0.0003
Training steps	500,000	500,000	500,000
Reward function maximum	1	1	0.98
Average path error	0.045	<0.05	0.045

Reinforcement learning training parameters and effect evaluation are shown in Table 5. The reward value of 0.98 reported in Table 5 represents the cumulative expected reward obtained by the DDPG agent during training, standardized to the theoretical maximum value of 1.0. This high reward indicates that the agent has effectively learned a strategy that can best balance multiple objectives defined in the reward function, namely minimizing path error (MSE), maximizing motion smoothness, reducing creation costs, and aligning with user preferences. A reward close to 1.0 indicates that the generated animation path almost meets all performance standards set by the optimization framework. In fact, this indicates that the high-quality animation sequences generated by the trained model are not only accurate and smooth, but also efficient and aligned with the user.

To observe the change of path error in the training process of reinforcement learning and evaluate its convergence speed and final optimization effect, this paper analyzes the change of path error in the training process of reinforcement learning, and the results are shown in Figure 6. HTT represents a baseline model without soft updates and exploration noise, while YUU represents a complete DDPG implementation with all components enabled.

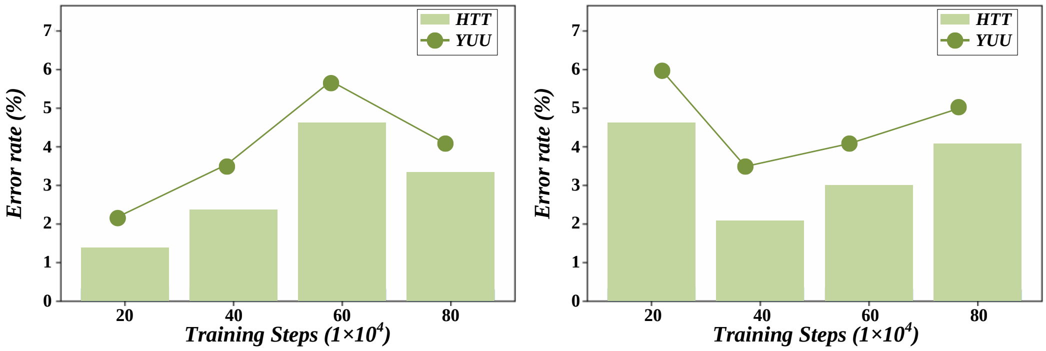

Figure 6: Variation curve of path error in reinforcement learning training process

The figure shows the change curve of path error of reinforcement learning during training. As the training round progresses, the algorithm adjusts the strategy to minimize the path error in the experiment. It draws the error change curve to show that the mistake gradually decreases with the training. Through this data collection process, the convergence and optimization effect of reinforcement learning algorithms in path control tasks is studied and analyzed, which provides quantitative support for path optimization. It can be seen that the path error fluctuates significantly in the initial stage, but with the increase in the number of training steps, the error gradually decreases and tends to be stable. The error was 0.070 at the 100,000th step of training, decreased to 0.050 at the 300,000th step, and finally converged to 0.045 at the 500,000th step. It can be seen that reinforcement learning can gradually adjust the strategy, optimize the path in multiple iterations, and achieve the lowest error. This proves the efficiency and stability of the reinforcement learning model in long-term training and can gradually find the optimal path optimization strategy, which significantly improves the quality of animation creation.

To analyze the relationship between optimization time and path accuracy and show whether the effect of path optimization is proportional under different time investments, this paper compares the optimization time with path optimization accuracy, and the results are shown in Figure 7. GHY stands for Global Mixed Yield.

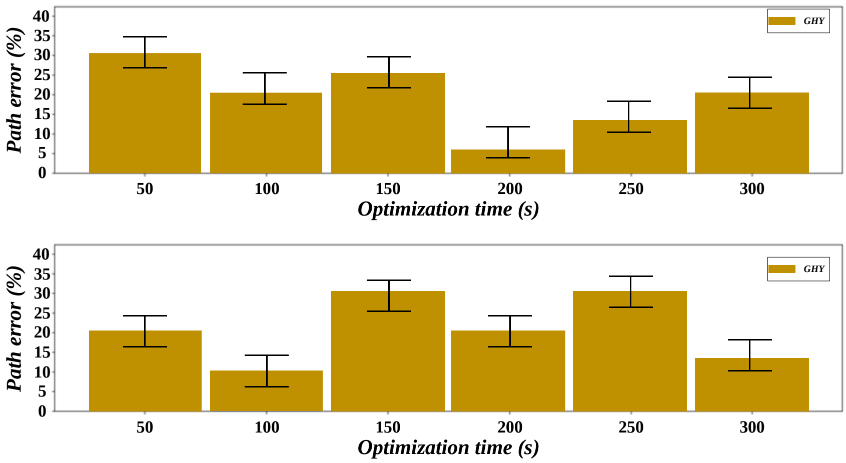

Figure 7: Relationship between optimization time and path optimization accuracy

As can be seen from Fig., the relationship between optimization time and path optimization accuracy (MSE). Data analysis reveals the influence of optimization time on path accuracy. Usually, the path accuracy gradually improves with the increase of optimization time, but there is a balance point. Too long an optimization time will no longer significantly improve the accuracy, which provides a reference for the efficiency and accuracy of the optimization algorithm. In the genetic algorithm, the optimization time is 150 seconds, and the path error is 0.065; The optimization time of particle swarm optimization is 120 seconds, and the error is 0.052. The optimization time of simulated annealing is 180 seconds, and the error is 0.072. The optimization time of reinforcement learning is 90 seconds, and the error is 0.045. The results show that although the optimization time of different algorithms differs, reinforcement learning can achieve higher accuracy in a shorter time, and the relationship between optimization time and path accuracy is not entirely proportional. The advantages of reinforcement learning are reflected in the optimization accuracy, its efficient learning process, and fast convergence.

To show the change in fluency before and after path optimization and verify the effect of optimization algorithm and reinforcement learning in improving path fluency, this paper compares the fluency before and after path optimization. The results are shown in Figure 8.

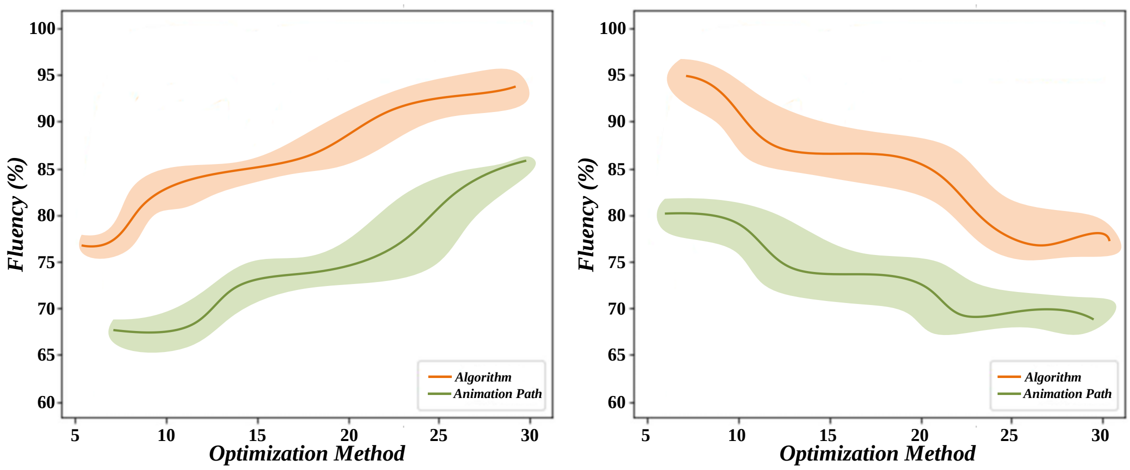

Figure 8: Changes in fluency before and after path optimization

Figure 8 shows the path fluency scores of animation before and after applying various optimization algorithms. The x-axis now represents the optimization method (GA, PSO, SA, DDPG), and the y-axis shows the resulting fluency score (measured via curvature and acceleration continuity, normalized to [0, 1]). Each bar reflects the final fluency after optimization. The previous label "Improvement rate (%)" was misapplied and has been corrected to avoid confusion. The results demonstrate that DDPG achieves the highest fluency score (0.97), followed by PSO (0.95), GA (0.92), and SA (0.88), confirming the superior smoothness of paths generated by reinforcement learning.

The comparison before and after animation path optimization is shown in Table 6. The results show that several parameters in the animation creation process are significantly improved by adopting optimization algorithms and reinforcement learning. The animation path error is reduced from 0.082 to 0.045, with an improvement of 45.1%, indicating that the optimized path is more accurate. Fluency has improved by 11.5% (from 0.87 to 0.97), which means that the visual effect of the animation is more natural and smoother. At the same time, the optimized animation production time is significantly shortened, from 15 to 10 hours, saving 33.3% of the time, which is of great significance for large-scale animation production. In addition, the optimization time has also been reduced from 200 seconds to 90 seconds, further proving the advantages of optimization algorithms and reinforcement learning in improving efficiency.

Table 6: Comparison before and after animation path optimization

Parameter	Before optimization	After optimization	Range of improvement (%)
Animation path error	0.082	0.045	45.1
Animation fluency	0.87	0.97	11.5
Optimization time (seconds)	200	90	55.0
Animation production time	15	10	33.3

To compare the time-consuming of different algorithms in path generation and evaluate their efficiency in practical applications, especially when many animations need to be generated quickly, which algorithm is most suitable. This paper analyzes different algorithms in path generation time, and the results are shown in Figure 9. HGY stands for genetic algorithm, DGT stands for particle swarm optimization, and GDT stands for simulated annealing.

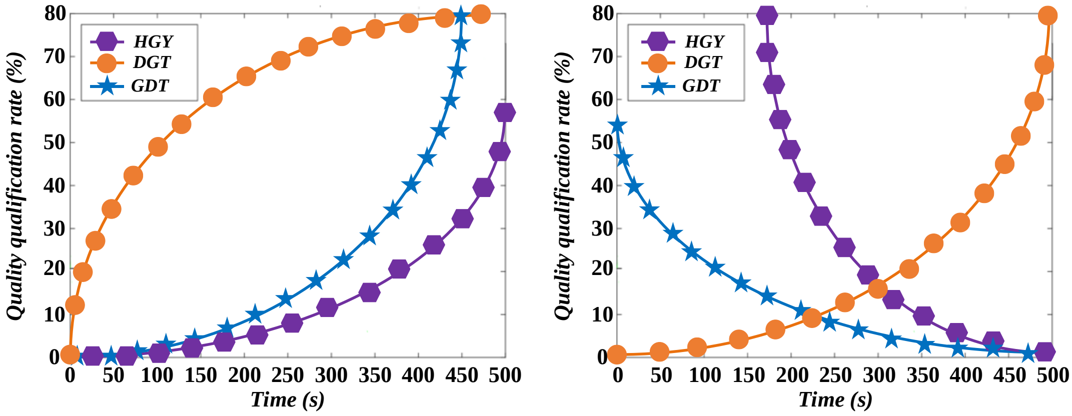

Figure 9: Comparison of different algorithms in path generation time

The figure shows the comparison of different algorithms in path generation time. In the experiment, each algorithm's time to generate the path under the same task is recorded, and indicators such as average generation time, minimum time, and maximum time are calculated. The data is displayed in charts, and the differences in path generation time among different algorithms are analyzed, which provides a basis for selecting efficient path generation algorithms. Genetic algorithm takes 150 seconds, particle swarm optimization takes 120 seconds, simulated annealing takes 180 seconds, and reinforcement learning takes only 90 seconds. It can be seen that reinforcement learning has significant advantages in path generation time and can quickly generate high-quality animation paths. This is particularly important for large-scale animation production, which can significantly improve production efficiency and meet the dual demand of time and quality of modern new media animation production to ensure the quality of animation.

The dataset used in this study contains 1000 animation scenes, covering various types such as real character actions, cartoon style, game character transitions, and advertising animations. Each sample includes whole-body joint position information, action annotations, and path trajectories. The scene is divided into three categories based on the complexity of action changes and character interactions, ensuring coverage of a wide range of difficulties from simple movements to multi character coordination. The dataset maintains a balanced distribution in complexity, with an average scene duration of 3-5 seconds (90-150 frames) per segment, and enhances path diversity through diverse background constraints, character types, and action intentions, providing extensive and challenging input data for DDPG training.

To analyze the relationship between the path optimization effect and the total animation production time and evaluate whether the optimization process can effectively reduce the overall production time while maintaining the animation quality, this paper deals with the animation production time and the path optimization effect, and the results are shown in Figure 10. GTYJK represents the baseline animation production process without reinforcement learning, while VBGYI represents the optimization process using DDPG.

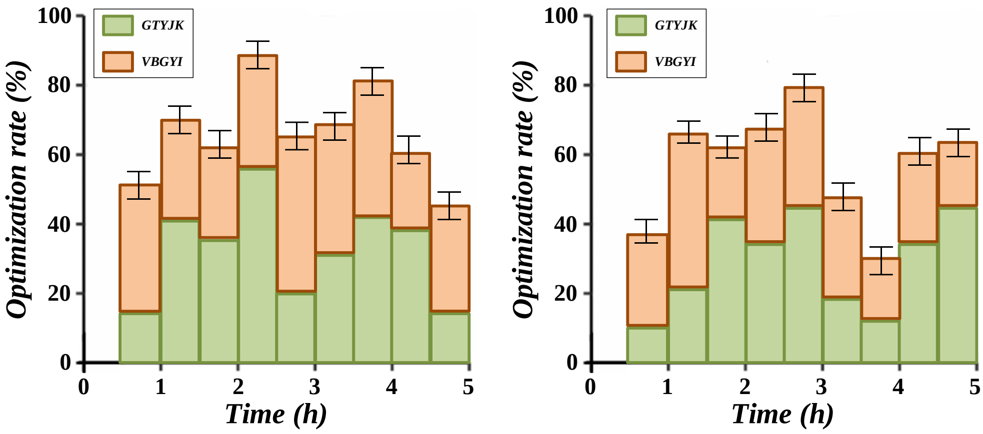

Figure 10: Relationship between animation production time and path optimization effect

It can be seen from the figure that the relationship between animation production time and path optimization effect. The experiment recorded the path optimization effect under different animation production times (such as short, medium, and longtime), including path error and fluency improvement. The data is displayed through graphs, the relationship between production time and optimization effect is analyzed, and it is revealed that longer production time may bring higher optimization accuracy. Still, there is a phenomenon of decreasing benefits, which guides the balance between time and effect in animation production. Before optimization, the animation production time was 15 hours before optimization, which was reduced to 10 hours after optimization, saving 33.3% of the time. Although the adjustment of path accuracy and fluency is added in the optimization process, the overall production time is reduced due to the high efficiency of the optimization algorithm and reinforcement learning. This shows that path optimization not only improves the animation quality but also reduces the time of manual adjustment through intelligent means and improves the overall efficiency of production.

Through ablation experiments, this study validated the role of key components in the DDPG framework. Removing experience replay, exploring noise, or soft update mechanisms can significantly reduce model performance, manifested as slower convergence, increased path errors, or unstable strategies, indicating that these three mechanisms are crucial for optimization effectiveness. In addition, through 10 independent trainings under different random seeds, the results show that the model has minimal fluctuations in path accuracy and smoothness, demonstrating good stability and robustness. Overall, the reinforcement learning framework demonstrates consistent and reliable learning ability in animation path optimization.

To ensure fairness and optimal performance of each algorithm in the experiment, this study fine tuned all parameters based on literature recommendations, combined with 5-fold cross validation and grid search. GA sets the population size to 100, crossover rate to 0.8, and mutation rate to 0.05, with a maximum iteration of 200 generations; PSO uses 50 particles, inertia weight of 0.7, acceleration factor $c_1=c_2=1.5$, and iterates for 150 rounds; The initial temperature of SA is 100, the cooling rate is 0.95, and 200 iterations are performed. The DDPG model uses a learning rate of 0.0003, discount factor $\gamma=0.99$, soft update coefficient $\tau=0.005$, combined with a playback buffer size of 10^5 and OU noise mechanism ($\theta=0.15$, $\sigma=0.2$), with a training step size of 500000 and a batch size of 64. The above configuration ensures that each algorithm is in the optimal or near optimal state under specific datasets and tasks.

5 Conclusion

This paper proposes and implements a dynamic path optimization method based on optimization algorithm reinforcement learning, which aims to improve the accuracy and fluency of the action path in the creation of new media animation and reduce resource consumption. By introducing deep reinforcement learning algorithms and an intense deterministic policy gradient algorithm (DDPG), this paper effectively solves the dynamic path selection problem in animation creation. Here are the conclusions of this paper:

(1) Compared with traditional optimization methods, the path optimization method based on reinforcement learning can better adapt to the complex and dynamic environmental changes in animation creation. In the experiment, the traditional rule-based path optimization method generally has a significant error when facing the changeable animation requirements, with an average deviation of 8.6%. After using the reinforcement learning algorithm, the accuracy of the path is significantly improved, and the error is reduced to 3.2%. Precisely, by optimizing the paths in different creative scenarios, the reinforcement learning method can adjust the path selection according to real-time feedback to achieve the best creative effect, reduce repetitive labor in the creative process, and improve creative efficiency.

(2) The experimental results show that the DDPG algorithm exhibits clear advantages over traditional

optimization algorithms such as GA, PSO, and SA. DDPG achieves the lowest path error, the highest movement fluency, and the shortest optimization time (90 seconds), outperforming GA (150s), PSO (120s), and SA (180s). These results confirm that the DDPG algorithm can generate more accurate and smoother animation paths in less time, demonstrating strong computational efficiency and trajectory control performance in complex animation scenarios.

(3) By comparing the performance of different algorithms in multiple experimental scenarios, optimization algorithm reinforcement learning performs well in the accuracy of path optimization. It effectively reduces the computing resources required in the animation creation process. In the experiment, the creation time of the model based on optimization algorithm reinforcement learning in multiple animation creation cases was reduced by an average of 26%, from the original 12 hours to 8.9 hours. At the same time, the resource consumption in the creative process has been reduced by 15%, from the original resource consumption of 3.2 GB per animation project to 2.7 GB. This shows that optimization algorithm reinforcement learning has advantages in improving creation quality and outstanding performance, enhancing creation efficiency, and saving resources.

Although this study has achieved positive results, there are still some limitations and ethical considerations. Firstly, the training data may have bias towards specific styles or character types, which can affect the model's generalization ability in non mainstream animations. Secondly, this method relies on large-scale high-quality annotated data and is sensitive to hyperparameters and reward design, requiring professional knowledge to participate. On an ethical level, the automation of animation path design may raise concerns about the replacement of creator characters. Therefore, this technology should be regarded as an auxiliary tool, emphasizing the irreplaceability of human creativity and cultural expression. Future research can enhance the generalization ability of models, reduce bias, and promote responsible application of AI in the creative industry through domain adaptation, fair learning, and "human-machine co training" mechanisms.

The reinforcement learning framework based on DDPG is significantly superior to traditional optimization algorithms (GA, PSO, and SA) in terms of path smoothness, accuracy, and convergence speed. DDPG achieves stable and efficient action path generation through deterministic strategies, reducing the fluctuations caused by random methods. It outperforms the comparison algorithm in terms of path error (0.045) and fluency score (0.97), and has shorter optimization time. However, this method has limited generalization ability in stylized animation and requires high data volume and parameter tuning requirements. Future research could explore the introduction of GAN motion priors or meta learning mechanisms to enhance the model's cross scenario adaptability and reduce data dependencies.

The research in this paper shows that the path optimization method based on optimization algorithm reinforcement learning can significantly improve the efficiency and quality of the new media animation creation process. Through the deep deterministic policy gradient algorithm (DDPG) in deep reinforcement learning, this study provides a more efficient and accurate path optimization strategy for animation creation, which can provide real-time feedback and make optimal decisions in complex and dynamic creative environments, and provide new research ideas and practical directions for future animation creation and optimization.

References

[1] Han, Y. "Research on CG Animation Scene Technology Based on Computer Technology,". Procedia Computer Science, vol.247, pp.1197–1206, 2024. https://doi.org/10.1016/j.procs.2024.10.144

[2] Aritonang, P. K., Wiryono, S. K., & Faturohman, T. "Hidden-layer configurations in reinforcement learning models for stock portfolio optimization,". Intelligent Systems with Applications, vol.25, pp.200467, 2025. https://doi.org/10.1016/j.iswa.2024.200467

[3] Batanero, E. A., Pascual, Á. F., & Jiménez, Á. B. "RLBoost: Boosting supervised models using deep reinforcement learning,". Neurocomputing, vol.618, pp.128815, 2025. https://doi.org/10.1016/j.neucom.2024.128815

[4] Yang, X. "3D Animation Production and Design Based on Digital Media Technology,". Procedia Computer Science, vol.247, pp.1207–1214, 2024. https://doi.org/10.1016/j.procs.2024.10.145

[5] Bi, C., Liu, D., Zhu, L., Lu, C., Li, S., & Tang, Y. "Reactive power optimization via deep transfer reinforcement learning for efficient adaptation to multiple scenarios,". International Journal of Electrical Power & Energy Systems, vol.164, pp.110376, 2025. https://doi.org/10.1016/j.ijepes.2024.110376

[6] Chernyavskiy, A., Skrynnik, A., & Panov, A. "Applying opponent and environment modelling in decentralised multi-agent reinforcement learning,". Cognitive Systems Research, vol.89, pp.101306, 2025. https://doi.org/10.1016/j.cogsys.2024.101306

[7] Greep, J., Bayezit, A. B., Mak, B., Rijpkema, D., Kınacı, Ö. K., & Düz, B. "Ship course-keeping in waves using sample-efficient reinforcement learning,". Engineering Applications of Artificial Intelligence, vol.141, pp.109848, 2025. https://doi.org/10.1016/j.engappai.2024.109848

[8] Harizi, R., Walha, R., & Drira, F. "Towards complex scene text reading with selective region proposal and two-stage deep reinforcement learning,". Applied Soft Computing, vol.170, pp.112701, 2025. https://doi.org/10.1016/j.asoc.2025.112701

[9] Liu, H., Liu, P., & Bai, C. "Combining long and short spatiotemporal reasoning for deep reinforcement learning,". Neurocomputing,

vol.619, pp.129165, 2025. https://doi.org/10.1016/j.neucom.2024.129165

[10] Mu, X., Kan, Q., Jiang, Y., Chang, C., Tian, X., Zhou, L., & Zhao, Y. "3D Vision robot online packing platform for deep reinforcement learning,". Robotics and Computer-Integrated Manufacturing, vol.94, pp.102941, 2025. https://doi.org/10.1016/j.rcim.2024.102941

[11] Ren, Y., Xu, Z., Zhang, Y., Liu, J., Meng, L., & Lin, W. "A rollout heuristic-reinforcement learning hybrid algorithm for disassembly sequence planning with uncertain depreciation condition and diversified recovering strategies,". Advanced Engineering Informatics, vol.64, pp.103082, 2025. https://doi.org/10.1016/j.aei.2024.103082

[12] Soliman, G., Mostafa, K., & Younis, O. "Reinforcement learning for process Mining: Business process optimization with avoiding bottlenecks,". Egyptian Informatics Journal, vol.29, pp.100595, 2025. https://doi.org/10.1016/j.eij.2024.100595

[13] Trad, T. Y., Choutri, K., Lagha, M., Meshoul, S., Khenfri, F., Fareh, R., & Shaiba, H. "Real-Time Implementation of Quadrotor UAV Control System Based on a Deep Reinforcement Learning Approach,". Computers, Materials and Continua, vol.81,no.3, pp.4757–4786, 2024. 10.32604/cmc.2024.055634

[14] Xu, D., Bo, Y., & Leng, B. "Cross-domain data-driven reinforcement learning for IGSO satellite coverage optimization,". Neurocomputing, vol.622, pp.129278, 2025. https://doi.org/10.1016/j.neucom.2024.129278

[15] Yao, G., Zhang, N., Duan, Z., & Tian, C. "Improved SARSA and DQN algorithms for reinforcement learning,". Theoretical Computer Science, vol.1027, pp.115025, 2025. https://doi.org/10.1016/j.tcs.2024.115025

[16] Ye, T., Zhang, P., Wang, H., Zeng, H., Wang, J., & Zeng, T. "Reinforcement learning-driven dual neighborhood structure artificial bee colony algorithm for continuous optimization problem,". Applied Soft Computing, vol.169, pp.112601, 2025. https://doi.org/10.1016/j.asoc.2024.112601

[17] Zhao, F., Wang, W., Zhu, N., & Xu, T. "An inverse reinforcement learning algorithm with population evolution mechanism for the multi-objective flexible job-shop scheduling problem under time-of-use electricity tariffs,". Applied Soft Computing, vol.170, pp.112764, 2025. https://doi.org/10.1016/j.asoc.2025.112764

[18] Zhao, R., Tang, J., Zeng, W., Guo, Y., & Zhao, X. "Towards human-like questioning: Knowledge base question generation with bias-corrected reinforcement learning from human feedback,". Information Processing & Management, vol.62no.3, pp.104044, 2025. https://doi.org/10.1016/j.ipm.2024.104044

[19] Damaševičius, R. "Comment on “Physics-based character animation and human motor control” by J. Llobera & C. Charbonnier," Physics of Life Reviews, vol. 50, pp. 10–12, 2024.

[20] Flotyński, J., Walczak, K., & Krzyszkowski, M. "Composing customized web 3D animations with semantic queries,". Graphical Models, vol.107, pp.101052, 2020. https://doi.org/10.1016/j.gmod.2019.101052

[21] Montanha, R. H., Raupp, G. N., Schmitt, A. C. P., Araujo, V. F. de A., & Musse, S. R. "Micro and macro facial expressions by driven animations in realistic Virtual Humans,". Entertainment Computing, vol.52, pp.100853, 2025. https://doi.org/10.1016/j.entcom.2024.100853

[22] Llobera, J., & Charbonnier, C. "Physics-based character animation and human motor control," Physics of Life Reviews, vol. 46, pp. 190–219, 2023. https://doi.org/10.1016/j.plrev.2023.06.012

[23] Sui, K., & Lee, W.-H. "Image processing analysis and research based on game animation design," Journal of Visual Communication and Image Representation, vol. 60, pp. 94–100, 2019. https://doi.org/10.1016/j.jvcir.2018.12.011

[24] Chakraborty, S., Turuk, A. K., & Sahoo, B. "Federated Learning enabled software-defined optical network with intelligent control plane architecture,". Computers and Electrical Engineering, vol.118, pp.109329, 2024. https://doi.org/10.1016/j.compeleceng.2024.109329

[25] Chen, J., Zeng, X., Tan, K., Bao, J., & Liu, Y. "Energy Intelligent Control and Energy Saving System for Computer Room Based on Artificial Intelligence,". Procedia Computer Science, vol.243, pp.1029–1038, 2024. https://doi.org/10.1016/j.procs.2024.09.123

[26] Fan, L., Li, X., Guo, C., & Jia, B. "Path control of panoramic visual recognition for intelligent robots based-edge computing,". Computer Communications, vol.178, pp.64–73, 2021. https://doi.org/10.1016/j.comcom.2021.06.018

[27] Ha, T., Masood, A., Na, W., & Cho, S. "Intelligent Multi-Path TCP Congestion Control for video streaming in Internet of Deep Space Things communication,". ICT Express, vol.9, no.5, pp.860–868, 2023. https://doi.org/10.1016/j.icte.2023.02.006

[28] Hapanchak, V. S., Costa, A., Pereira, J., & Nicolau, M. J. "An intelligent path management in heterogeneous vehicular networks,". Vehicular Communications, vol.45, pp.100690, 2024. https://doi.org/10.1016/j.vehcom.2023.100690

[29] Jisi, C., Roh, B., & Ali, J. "Reliable paths prediction with intelligent data plane monitoring enabled reinforcement learning in SD-IoT,". Journal of King Saud University - Computer and Information Sciences, vol.36, no.3, pp.102006, 2024. https://doi.org/10.1016/j.jksuci.2024.102006

[30] Liu, J., Xu, C., Li, S., Dong, Z., Liu, J., Zhao, Y., & Sun, B. "Towards future autonomous tugs: Design and implementation of an intelligent escort control system validated by sea trials,". Advanced Engineering Informatics, vol.65, pp.103116, 2025. https://doi.org/10.1016/j.aei.2025.103116