ingegneria sismica
International Journal of Earthquake Engineering
Anno XLIII - Num. 2 - 2026

Animation Character Behavior Generation Algorithm Optimized by Deep Reinforcement Learning

Xiaoxuan Guan[1,*] *and Jian Zhang*[1]

[1] Animation Art College, Jilin Animation Institute, Changchun, 130012, Jilin, China

SUMMARY: *With the increasing complexity of animation production, behavior generation has become a key link to maintain the temporal consistency, physical rationality and expression coordination of virtual characters. In this paper, we propose a deep reinforcement learning based animation character behavior generation algorithm, which jointly encodes multi-modal motion cues, scene states and semantic action labels to generate controllable behavior sequences. Based on the actor-critic learning mechanism and hierarchical state representation, a policy learning framework is constructed to improve the quality of action cohesion and scene response accuracy. Experiments are carried out on 12480 segments of motion data composed of CMU Mocap, Mixamo and self-built interactive segments. Action accuracy, macro average F1 value, trajectory offset error and structural similarity are used as evaluation indicators, and GRU, Transformer and PPO-RNN are used as comparison methods. The results show that the motion accuracy of the proposed method reaches 95.84%, the macro-average F1 value is 93.12%, the structural similarity is 0.927, and the trajectory offset error is 3.84 cm. The scoring results show that the proposed method has better performance in smoothness, controllability and visual naturalness.*

KEYWORDS: *Reinforcement learning; Behavior generation; State encoding; Policy optimization*

1 Introduction

In the context of digital content production moving towards data-driven, intelligent and interactive, animation character behavior generation has shifted from rule-driven action splicing to joint modeling oriented to semantics, timing and environmental feedback. Behavior sequence not only determines whether the character movement is coherent, but also affects the narrative rhythm, emotional transmission and the stability of scene response. For the production process of virtual human, game animation and digital film and television, the physical rationality of skeletal trajectory and the consistency of shot expression and action style should be maintained when characters complete actions such as turning, walking, stopping, avoiding and interacting, which makes behavior generation a core task in computer animation.

Kwiatkowski et al. [1] systematically reviewed the reinforcement learning methods in character animation, and pointed out the application value of policy learning in action control, adaptive decision-making and continuous behavior generation. Qin et al. [2] proposed a muscle driven virtual human motion generation method based on deep reinforcement learning, so as to form a closer correspondence between action output and biomechanical constraints.

*animationwuah@126.com
https://doi.org/10.65102/is2026728 Published: 30 April, 2026

Zhu et al. [3] reviewed the research on human motion generation, and summarized the main evolution paths of the temporal modeling, pose representation, and generation framework. Zhang et al. [4] proposed MotionDiffuse model, which introduced text conditions into the human action generation process and enhanced the action expression ability driven by semantics. Gao et al. [5] proposed a gradual enrichment synthesis strategy to improve action detail completion and overall coherent expression in text-driven scenes. Hu et al. [6] proposed a pose-aware attention network oriented to body parts, which provided a more detailed structural support for local coordination in action retargeting. Ji et al. [7] proposed StyleVR method, which uses normalized flow to realize the stylized processing of character animation and enhances the controllability of action style transfer. Zhang et al. [8] studied the speech-driven personalized gesture synthesis method, and improved the degree of adaptation between action generation and individual expression with the help of fuzzy feature inference. Hu et al. [9] proposed a diffusive human action style transfer method with semantic guidance, which made the style mapping process have stronger semantic preservation ability. Sun et al. [10] proposed a language-oriented virtual human action generation method based on the musculoskeletal model, which promoted the deep coupling between language instructions and action mechanisms.

Existing research has provided a foundation for animation character action generation, but the behavior generation for the actual production link still needs to deal with the collaborative relationship between state perception, action semantics and policy update. The generation method relying solely on supervised learning is better at fitting the existing action distribution, but it is difficult to continuously use the reward signal to correct the action selection in the process of scene switching, interactive triggering and long-term behavior expansion. If the character is in a multi-objective driven environment, the behavior sequence also needs to take into account semantic matching, motion smoothness, constraint satisfaction and visual naturalness, and it is difficult to achieve high-quality output by static mapping only.

Based on this background, this paper focuses on deep reinforcement learning to optimize the animation character behavior generation algorithm, and organizes the character state representation, action semantic encoding and strategy learning mechanism into a unified framework. The model extracts joint representations from skeletal motion features, scene context and action labels, and completes policy iteration under the actor-critic structure, so that the actor can dynamically adjust the action output according to the environmental feedback. This method emphasizes the continuous decision attributes in behavior generation and also the controlled expression of action amplitude, direction switch and rhythm change in animation scenes. Through the construction of behavior-driven feature extraction, action constraint modeling and policy optimization modules, the character behavior generation process is transformed into a trainable, evaluable and iteratively updated computing task. The research content of this paper is oriented to the direction of computer animation, intelligent generation and virtual character control, which can provide engineering basis for behavior generation in animation production.

2 Related Research

With the advancement of computer graphics, motion modeling and intelligent decision technology, animation character behavior generation has gradually shifted from keyframe splicing to data-driven sequence synthesis. Action generation is no longer just the connection of local action segments, but needs to maintain motion continuity, semantic consistency and scene adaptation in the time dimension. Around this direction, the research at home and abroad has been carried out from many levels, such as music driving, audio collaboration, action style transfer, 3D asset generation, speech facial animation and expression retargeting, which provides a reference technical basis for deep reinforcement learning to participate in character behavior control.

Yang et al. [11] studied the musce-driven 3D dance generation task, and proposed a keyframe control mechanism to establish a more stable correspondence between the movement rhythm and the music structure. Niu et al. [12] proposed Audio2AB collaborative generation method, which jointly modeled audio information and virtual character animation, and enhanced the synchronous expression between character actions and acoustic cues. Akber et al. [13] systematically combed the tools and technical paths of action style transfer based on deep learning, and summarized the main implementation methods of style mapping, feature decoupling and cross-domain transfer. Lungu-Stan et al. [14] studied the 3D character animation and asset generation process based on deep learning, and proposed the idea of using the generation model to support the creation of character content, which laid the modeling foundation for the automation of character behavior generation.

From the perspective of computational model, these researches roughly form three technical routes. One class of methods relies on conditional generative networks or diffusion models to synthesize action sequences, focusing on distribution learning in high-dimensional action Spaces. One kind of method emphasizes the cross-modal alignment between audio, text, expression and skeleton trajectory, trying to improve the correspondence accuracy between behavior output and external driving signals. Another class of methods incorporate geometric priors, anatomical structures, or style constraints into the generation process to ensure the stability and credibility of the character's actions at the visual level. The above routes enhance the performance tension of character animation, and also show that the behavior generation has developed from a single trajectory prediction to a comprehensive modeling task that integrates semantic, structural, and control goals. For the software system oriented to animation production, the output of character behavior needs to take into account rendering interface, skeleton driver layer and camera rhythm control. Therefore, the generation algorithm cannot only pursue local accuracy, but also need to maintain the executability of the whole action on the timeline. This further highlights the value of strategy learning and state feedback mechanism in character behavior modeling. Related research thus presents obvious computerized characteristics.

In the field of speech-driven character generation, facial animation research provides rich experience in temporal coding. Jiang et al. [15] reviewed audio-driven facial animation and summarized methods such as acoustic feature coding, mouth motion mapping and cross-modal learning. Wu et al. [16] proposed a talking face framework that can generate diverse and realistic facial animations, so that the action synthesis under the same speech condition has stronger natural variation ability. Zhai et al. [17] proposed a talking face generation method based on emotional keypoint inference, which explicitly embedded emotional information into the facial trajectory modeling process. Liu et al. [18] proposed a talking face generation method based on facial anatomical structure, which introduced the geometric structure prior into the animation synthesis process, so that the generated results were closer to the real texture motion.

Existing researches have promoted the development of animation character generation technology from the perspectives of motion control, cross-modal driving, style transfer and structural constraints. In order to more clearly present the technical focus, implementation effects and limitations of related work, this paper summarizes and collates representative studies, as shown in Table 1.

Table 1: Summary of related research on animated character behavior generation

Method Direction	Main Content	Technical Characteristics	Limitations	References
Music-driven motion generation	Dance keyframes are controlled according to musical rhythm and structural patterns.	Strong rhythm alignment	Weak expression of scene interaction	Yang et al. [11]
Audio-collaborative character animation	Audio and character motion information are jointly used for generation.	Good acoustic synchronization	Limited control over long-sequence behaviors	Niu et al. [12]
Motion style transfer	Style mapping and motion expression conversion are achieved.	Flexible style expression	Insufficient continuous decision-making mechanism	Akber et al. [13]
3D character asset generation	Deep models are used to support character content creation.	High degree of automation	Limited involvement in behavior control	Lungu-Stan et al. [14]
Speech-driven facial animation	A mapping between audio and facial movements is established.	Mature cross-modal encoding	Difficult to cover full-body behavior sequences	Jiang et al. [15]–Liu et al. [18]
Expressive virtual characters	Emphasis is placed on the consistency of facial expression, speech, and identity presentation.	Strong realism	Constraint modeling still needs refinement	Pan et al. [19]–Wagner et al. [21]

In terms of modeling expressive characters and retargeting, Pan et al. [19] studied the generation of expressive speaking avataras, which strengthened the coordination between the sense of identity of characters and expression drive. Zhu et al. [20] reviewed the action retargeting technology in 3D character facial animation, and systematically summarized the implementation routes such as parameter mapping, skeletal driving and geometric constraints. Wagner et al. [21] proposed a facial retargeting method combining anatomical constraints and data adaptation, which makes facial action transfer achieve a better balance between structural rationality and role adaptation.

Synthesizing the existing results, it can be seen that the related research has formed a relatively complete genealogy of methods in terms of motion generation, voice driven, style transfer and face retargeting, which provides an important support for the automatic generation of character animation. However, most of the researches focus on a single mode or local control unit, and less attention is paid to the continuous behavior decision, state feedback utilization and constraint consistency maintenance of animated characters in complex scenes. For the generation of character behavior in animation production link, it is also necessary to put action feature extraction, state representation, semantic encoding and policy learning into a unified framework. Based on this technology trend, this paper introduces deep reinforcement learning into the animation character behavior generation process, and uses the actor-critic mechanism to update the behavior strategy, so that the character can form a more stable computational expression between environment response, action continuity and semantic matching.

3 Animation character behavior generation and deep reinforcement learning task modeling

Animation character behavior generation is not a simple action playback, but a continuous calculation process around state perception, action decision and feedback correction. In digital animation, characters not only need to perform displacement, turning, pause, interaction and other actions, but also need to maintain the consistency of expression under the influence of camera context and semantic tasks. Therefore, the modeling basis of behavior generation cannot only stay at the skeleton trajectory interpolation level, but also need to map the action features, environmental information and control targets into a learnable state space, so that the character can form a traceable and updatable behavior chain within a multi-frame sequence. In the case of deep reinforcement learning, this foundation determines what information the policy network receives and how it evaluates the action output.

To illustrate the fundamental structure of role behavior generation in the computational flow, Fig. 1 shows the core path from input data to behavior state formation. In the figure, the left side shows the character skeleton trajectory, the scene interaction signal and the semantic target label, the middle is the feature warping and multi-modal fusion layer, and the right side is the unified state representation for policy learning. This figure emphasizes how an animated character transforms from discrete inputs to behavioral states.

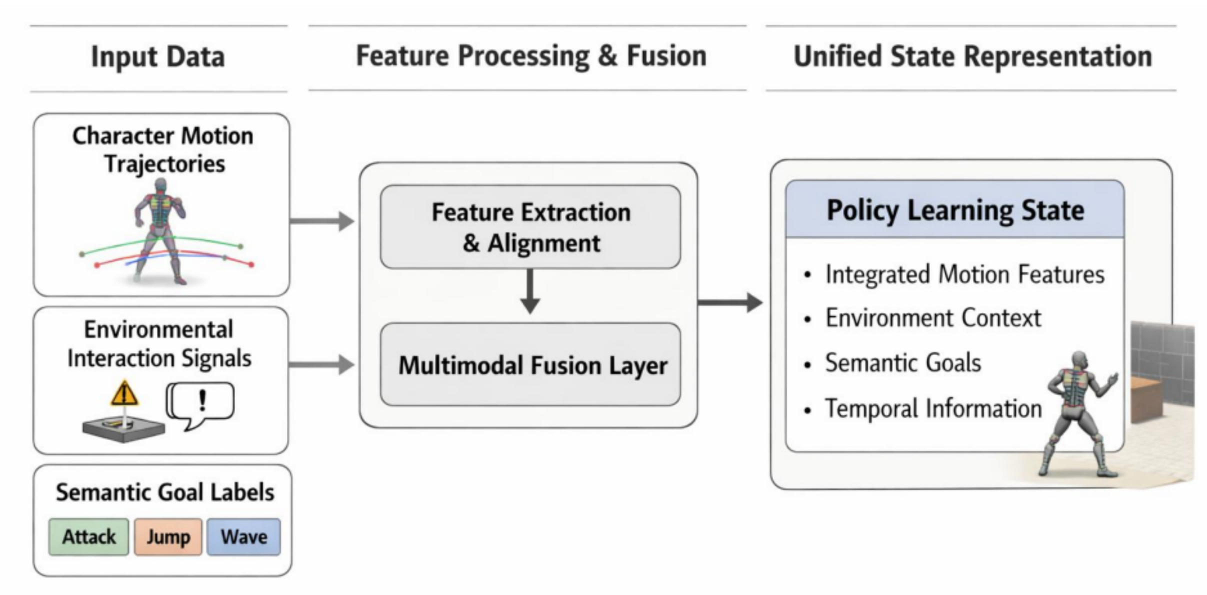

Figure 1: Input-state composition diagram for animated character behavior generation

The state representation in behavior generation task needs to reflect the current pose, historical action trend and external environment feedback simultaneously. If the state encoding is too sparse, the policy network can only learn local action changes. If the state encoding lacks semantic constraints, although the character can complete the movement, it is difficult to maintain the behavior logic consistent with the goal. In order to meet these two requirements, the state construction of animation characters usually introduces the position of skeleton nodes, joint velocities, contact markers, target directions, scene obstacle coding and the previous action memory, and compresdes these information into a low-dimensional latent space.

In order to make the character state simultaneously cover the current pose, motion trend, scene perception and control target information, the state vector in the behavior generation task is defined as follows.

$$s_t = \tanh\left(W_p p_t + W_v v_t + W_o o_t + W_g g_t + W_a a_{t-1} + b_s\right) \tag{1}$$

where s_t represents the state vector at time t, p_t represents the current attitude feature, v_t represents the motion trend feature, o_t represents the scene observation information, g_t represents the semantic control signal, a_{t-1} represents the action coding at the last time, W_p, W_v, W_o, W_g and $W_a$represent the corresponding mapping matrix, and b_s represents the bias term. The expression is used to map the action state, environment information and control goal into the same representation space, so that the current action basis, external scene change and goal driving information can be kept simultaneously in the process of continuous behavior generation, so as to provide structural stable and semantic complete input expression for the subsequent policy network.

To further illustrate the structural relationships in task modeling, Fig. 2 illustrates the state-action-reward association framework for the generation of animated character behaviors. In the figure, the unified state vector first enters the policy network, and then outputs the candidate action distribution, and then gets feedback through constraint verification and environment interaction, and writes back to the value evaluation branch. The procedure illustrates that character behavior is not directly determined by single frame actions, but is formed in state updates.

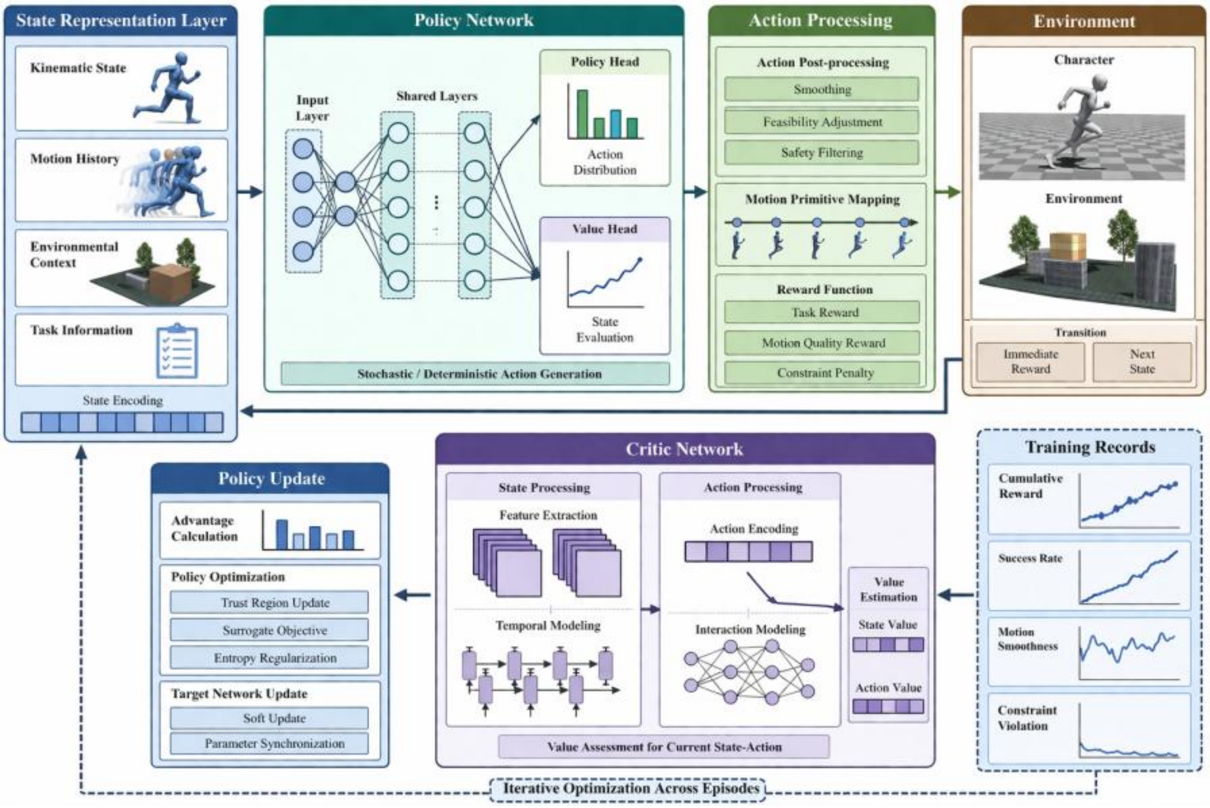

Figure 2: State-action-reward correlation graph generated for animated character behavior

In the scenario of deep reinforcement learning, the design of action space directly affects the quality of behavior generation. If the action is defined as an independent control quantity of an isolated joint, the role is prone to the coexistence of local swing and global imbalance. If actions only retain high-level labels, it is difficult to support fine-grained output. Therefore, a more prudent way is to establish a hierarchical mapping between high-level semantic actions and low-level posture increments, so that the policy network first determines the behavior categories such as move, turn, raise, and avoid, and then completes the parameter expansion by the execution layer. This not only preserves the behavior control boundary, but also ensures that the animation sequence can be executed.

In order to ensure that the policy update takes into account action coherence, semantic consistency and physical feasibility, the unified comprehensive reward function of the behavior generation stage is expressed as follows:

$$R_t = \lambda_1 e^{-\|\hat{p}_t - p_t^*\|_2^2} + \lambda_2 \cos(\hat{d}_t, d_t^*) + \lambda_3 \mathrm{Sim}(q_t, q_{t-1}) - \lambda_4 \max(0, E_t - \tau) - \lambda_5 C_t \quad (2)$$

Where R_t represents the comprehensive reward at time t, $\hat{p}_t$ represents the posture parameters generated by the current strategy, p_t^* represents the reference posture or target trajectory, $\hat{d}_t$ and d_t^* represent the generated action direction and target direction respectively, $\mathrm{Sim}(q_t, q_{t-1})$ represents the continuity evaluation term of two adjacent frames of action. E_t represents the physical error constituted by plantar slip, joint interstiction and collision penalty, τ represents the allowable threshold, C_t represents the semantic constraint deviation, and λ_1 to λ_5 represent the weights of different objective items. The reward function puts action accuracy, direction consistency, temporal smoothness, physical rationality and semantic satisfaction into the same evaluation system, which can avoid policy training only biased to a single index.

To illustrate how the above modeling foundation supports the complete task flow, Fig. 3 illustrates the task closed loop of deep reinforcement learning for animated character behavior generation. The graph unfolds from data input, state encoding, policy sampling, value evaluation to feedback update in turn, forming a closed-loop link for long sequence behavior generation. The key point of this figure is that the behavior generation task is not a one-time prediction, but a learning process of continuous observation and correction in parallel.

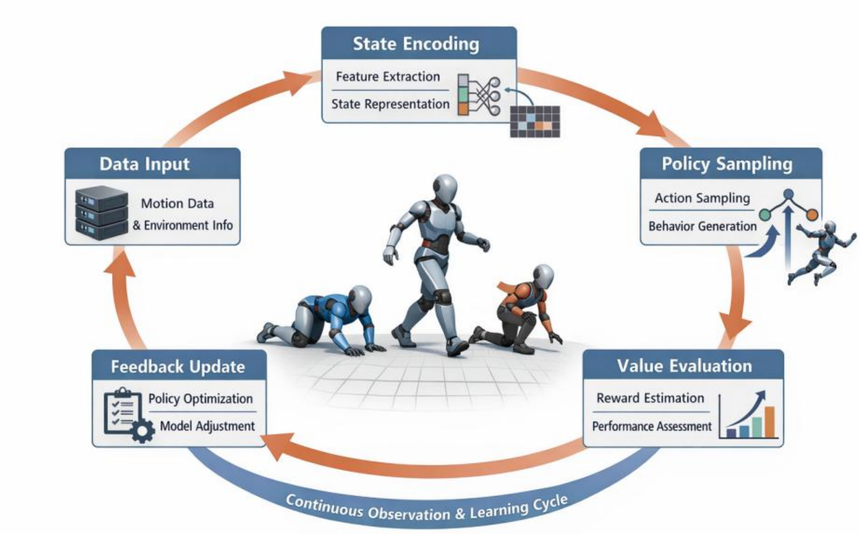

Figure 3: Task closed-loop graph for animated character behavior generation from deep reinforcement learning

In summary, the task modeling basis of animation character behavior generation is composed of multi-modal state representation, hierarchical action space and comprehensive reward function, which respectively undertake observation compression, attitude control and training evaluation. Only when these links are consistent, deep reinforcement learning can exert its ability to model complex animation sequences and provide basic support for subsequent algorithm construction. This basis does not directly give the final action, but determines the scope of the policy search, the precision of the feedback, the boundaries of the action, and the stability and performance of the animation during the rendering phase.

4 Deep reinforcement learning optimized animation character behavior generation algorithm construction

4.1 Animated character behavior drives feature extraction and state representation

The starting point of animation character behavior generation is not the action output, but the stable extraction of driving information. As the character completes moving, turning, pausing, evasive, and interacting in continuous footage, the control system should recognize both velocity changes in skeletal trajectories and the effects of scene goals, contact states, and rhythm cues on the way actions are organized. If the input feature only retains a single frame pose, what the network can learn is often only local changes. If the input features lack temporal correlation, the generated results can maintain the basic shape of the action, but it is difficult to maintain the cohesion between the action paragraphs. Therefore, in this paper, the kinematic information, phase information and environmental perception information are fed into the encoding path in the feature extraction stage, so that the subsequent state representation not only has action details, but also retains the task context.

To illustrate the organization process of behaviordriven features before they enter the network, Fig. 4 shows the complete path of the original animation sequence from sampling and warping to feature tensor construction. The left input of the figure includes bone position, joint rotation, foot contact marker and scene target vector. In the middle part, time resampling, coordinate alignment and local normalization were completed in turn. The right side outputs the multi-channel behavior characteristics for use by the state network. Through this processing chain, discrete action observations are collated into drive inputs in a unified format, which facilitates subsequent networks to continuously model character motion patterns.

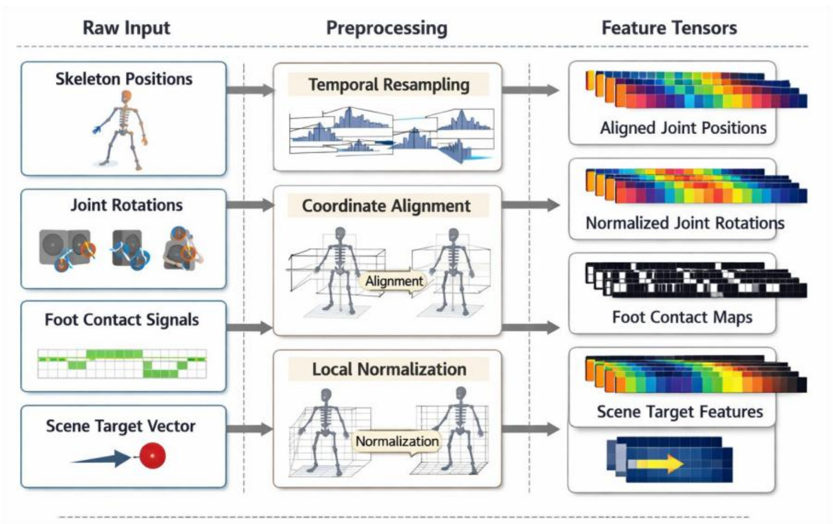

Figure 4: Flowchart of animated character behavior-driven feature extraction

In order to make local attitude, velocity change and target guidance information enter a unified channel, this paper defines the fusion expression form of basic driving features as follows:

$$x_t = \sigma\left(W_q q_t + W_u u_t + W_c c_t + W_r r_t + b_x\right) \tag{3}$$

where x_t represents the local driving feature at time t, q_t represents the joint rotation parameter, u_t represents the linear velocity and angular velocity of the root node, c_t represents the foot contact mark, r_t represents the target direction and relative distance, W_q, W_u, W_c and W_r represent the corresponding mapping matrix, b_x represents the bias term, and $\sigma(\cdot)$ represents the activation function. By compressing the posture, velocity, contact and target information into a unified feature space, the character action not only retains the geometric changes of the skeleton layer, but also reflects the scene drive and behavior trend, which provides a stable input basis for the state network.

After the locally driven features are obtained, the changes in the time dimension still need to be further expressed explicitly. Animation behavior is not a concatenation of discrete frames, but a trajectory process gradually unfolded by a continuous action. The difference between running and slow walking is not only reflected in the single frame attitude, but also reflected in the velocity fluctuation and phase advance of several adjacent frames. To this end, this paper introduces temporal difference and periodic phase encoding in addition to the basic features, so that the system can recognize the advancing rhythm in the action segment and reduce the state drift caused by frame rate changes.

In order to stably describe the advancement trend between consecutive frames, this paper further uses the time difference accumulation method to construct the motion trend feature. The specific expression is as follows:

$$\Delta x_t = \sum_{i=1}^{K} \alpha_i \left(x_{t-i+1} - x_{t-i}\right) \tag{4}$$

Here, Δx_t represents the feature difference between adjacent moments, x_{t-i} represents the feature of the preceding frame, α_i represents the time weighting coefficient, and K represents the length of the look-back window. This formula is used to construct an action trend description term, so that the encoding result not only depends on the current position, but also can perceive the previous motion trajectory. The differential features formed in this way have higher discrimination ability for dynamic segments such as starting, braking, steering and continuous avoidance, and also provide a smoother time reference for the behavior prediction network.

To further illustrate the formation process of the state vector, Fig. 5 shows the convergence relationship of local driving features, trend features, and scene observations after entering the state representation layer. In the figure, the first two branches are responsible for modeling action details and temporal changes, respectively. The third branch is connected to environment occupancy, target point and event label, and the fusion layer outputs a unified state vector. Through this structure, the motion basis, changing direction and external conditions of the character at a certain time are preserved at the same time, so that the state representation is more suitable for the subsequent action selection and update of the policy network.

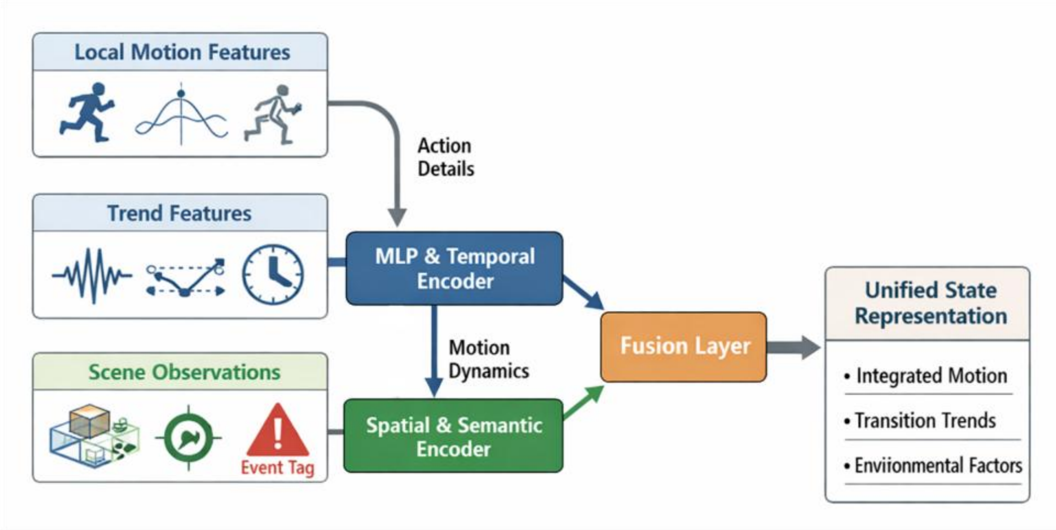

Figure 5: Animation character state representation sink structure diagram

In order to ensure that multi-source behavioral observations form a consistent expression in a unified state space, the fused state mapping relationship is expressed in the following specific mathematical form:

$$z_t = \tanh(W_x x_t + W_d \Delta x_t + W_e e_t + b_z) \tag{5}$$

where z_t represents the fused state vector, W_x, W_d and W_e represent the mapping matrix of driving features, trend features and environmental observations, respectively, e_t represents the scene coding result, b_z represents the state bias term, and $\tanh(\cdot)$ represents the nonlinear compression operation. Through this formula, multi-source inputs can be uniformly projected into the same state space, avoiding expression offset caused by direct superposition of different dimensions. At this stage, the state vector has three types of information: action basis, time trend and environmental conditions, so it can better support subsequent semantic modeling and policy update.

Considering the differences in scale and action style of character behavior in different shots, the state representation should also have certain stability constraints. If characters fluctuate greatly due to coordinate system changes under the same task semantics, policy learning will misjudge geometric offset as behavioral difference. Therefore, this paper normalizes the root node translation, orientation reference and joint amplitude, and introduces a regular balance term in the state layer to weaken the disturbance caused by the inconsistent data distribution across segments.

In order to reduce the disturbance effect on state distribution caused by cross-segment sampling differences, this paper defines the normalized state stability constraint in the state layer as follows:

$$L_{stab} = \|Norm(z_t) - Norm(z_{t-1})\|_2^2 + \eta\|z_t\|_2^2 \tag{6}$$

Here, L_{stab} represents the state stability constraint term, $Norm(\cdot)$ represents the normalization operation, $\eta\|z_t\|_2^2$ represents the regularization weight, and z_{t-1} represents the state vector at the previous time. On the one hand, the abnormal drift of the state distribution at adjacent moments is constrained, on the other hand, the disordered expansion of the state vector amplitude is inhibited, so as to ensure that the behavior driving features enter the subsequent modules in a uniform format. After the above processing, the state representation has trainable, comparable and engineer-adapted input features, which provides a stable foundation for action semantic encoding and reinforcement learning strategy construction.

Taken together, this section completes the construction process from raw action observations to a unified state representation. Skeletal information, velocity trends, contact events, and environmental codes are reorganized in the same state space, so that the input of character behavior no longer stays at the single pose level, but forms a computational expression with time continuity and task awareness. Such state representation can directly support the semantic encoding of subsequent actions, and provide a stable, clear and discriminative input basis for the policy network.

4.2 Animation character action semantic coding and behavior constraint modeling

Behavior generation for animated characters is not equivalent to direct regression on skeletal sequences. Every time a character moves, stays, attacks, looks back or interacts on the screen, it carries explicit action semantics and situational intent. If the model only learns the posture trajectory and does not hierarchically represent the behavior semantics, the generated results are easy to be continuous locally and distorted globally. To this end, this paper continues to establish an action semantic encoding layer after the state representation, which maps action labels, temporal phases, body part responses and scene events together into computable semantic vectors, and imposes constraints on them.

The core of action semantic coding is not to directly attach text labels to action frames, but to decompose character behaviors into computable constituent units. In this paper, high-level actions such as walking, running, turning, reaching, avoiding, and waiting are taken as semantic categories, the starting segment, advancing segment, and bandar segment are taken as temporal stages, and the torso, upper limb, lower limb, and line of sight direction are taken as the dimensions of execution parts. The resulting encoding can simultaneously describe the action type, phase location, and execution site.

In order to more clearly illustrate the composition of action semantic coding, Table 2 summarizes the main semantic elements and constraint objects adopted in this paper.

Table 2: Animation character action semantic encoding and constraint object correspondence table

Semantic Level	Encoded Object	Computational Form	Constraint Objective
Action category level	Walking, running, turning, dodging, interaction	Category embedding vector	Maintain high-level behavioral consistency
Phase position level	Initial phase, progression phase, closing phase	Phase embedding vector	Smooth action boundary transitions
Execution part level	Trunk, upper limbs, lower limbs, gaze	Part embedding vector	Coordinate body response relationships
Direction constraint level	Target orientation and movement direction	Direction loss term	Maintain consistency of motion targets
Contact constraint level	Sole contact and spatial foothold	Contact loss term	Reduce foot sliding and floating
Coordination constraint level	Coordination of upper limbs, lower limbs, and trunk	Coordination loss term	Maintain overall action coherence

In order to make the action category, stage position and body part information enter a unified semantic space, the basic semantic encoding vector is defined as follows:

$$y_t = E_a[l_t] \oplus E_p[p_t] \oplus E_b[b_t] \tag{7}$$

Here, y_t represents the semantic encoding vector at time t, E_a represents the action category embedding matrix, l_t represents the high-level action label, E_p represents the stage embedding matrix, p_t represents the action stage index, E_b represents the body part embedding matrix, b_t represents the execution part labeling, and $\oplus$ represents the splicing operation. This formula maps discrete semantic elements into the same continuous space, so that action labels are no longer just text annotations, but structured inputs that can participate in subsequent computation and policy update.

After semantic encoding is completed, boundary coordination between adjacent actions still needs to be solved. When the character switches from running to stopping, from looking back to avoiding, the model is easy to produce unnatural jitter at local moments if there is no transition mechanism. Therefore, this paper introduces a phase-gating term to model the transition strength of action initiation, advancement and bandaging, so that the semantic layer can explicitly adjust the switching speed and transition ratio, rather than relying on the network to absorb all boundary relations by itself.

In order to maintain a smooth transition in the switching process of adjacent semantic stages, this paper further defines the phase-gated regulation relationship as follows:

$$g_t = \sigma\left(W_s y_t + U_s y_{t-1} + b_g\right) \tag{8}$$

Here, g_t represents the stage gating vector, W_s represents the current semantic mapping matrix, U_s represents the preorder semantic mapping matrix, y_t represents the current semantic coding, y_{t-1} represents the previous semantic coding, b_g represents the gating bias term, and $\sigma(\cdot)$ represents the gating function. This formula is used to calculate the adjustment strength of the current output caused by the change of the action phase, so as to maintain a smoother and consistent relationship between the high-level behavior switch and the underlying motion trajectory, thereby reducing the jump phenomenon of the character on the transition frame.

In addition to semantic boundaries, character behavior must also satisfy geometric and physical constraints. If the torso orientation deviates from the moving direction for a long time, or the foot contact contradicts the root node displacement, the generated results will weaken the credibility of the behavior. In this paper, we construct three kinds of constraints: direction consistency, contact consistency and part coordination, and write these constraints into the unified modeling process to ensure the semantic correctness and the reasonable execution of actions.

To ensure that the movement direction of the character is consistent with the high-level action semantics, the direction consistency constraint in the behavior generation is defined as follows:

$$C_{dir} = 1 - \frac{\hat{d}_t \cdot d_t^*}{\|\hat{d}_t\|_2 \|d_t^*\|_2} \tag{9}$$

Here, C_{dir} represents the direction consistency term. This formula is used to constrain the movement direction of the character to be consistent with the semantic goal, and avoid the phenomenon that the semantic forward trajectory moves to the side. The direction constraint directly acts on the long sequence of motion clips, which has a significant impact on the stability of the character walking and the continuity of the camera presentation.

In order to ensure the coordination between foot contact events and spatial pose changes, the contact consistency constraint term is defined as follows:

$$C_{con} = f_t\left|\hat{h}_t\right| + (1 - f_t)\max\left(0, \epsilon - \hat{h}_t\right) \tag{10}$$

Here, C_{con} represents the contact agreement term, f_t represents the predicted contact marker, $\hat{h}_t$ represents the foot height, and ϵ represents the allowable threshold. This formula is used to constrain the foot contact events of the character to be consistent with the real spatial position, and reduce unnatural performances such as foot slip, suspension and landing misalignment. Contact constraints are particularly critical for run-jump, emergency stop, and steering segments, which are extremely sensitive to contact feedback.

In order to ensure that the responses of upper limbs, lower limbs and trunk change coordinatively under the same task, this paper defines the locality coordinatively constraint term as follows.

$$C_{part} = \left\|A_u\hat{m}_t^u + A_l\hat{m}_t^l - A_k\hat{m}_t^k\right\|_2^2 \tag{11}$$

Here, C_{part} represents the part synergy term; $\hat{m}_t^u$, $\hat{m}_t^l$, and $\hat{m}_t^k$ represent the predicted motion responses of the upper and lower limbs with the trunk, respectively; A_u, A_l, and A_k represent the part association matrix. This formula is used to maintain the response coordination of different body parts under the same semantic task, avoiding the separation phenomenon that the lower limbs have switched while the upper limbs are still stuck in the old action stage. Part collaborative modeling makes the behavior sequence look and feel more complete, and keeps the action semantics clearly expressed in the time axis.

After completing the local constraints, this paper further constructs the constraint energy, and aggregates the direction, contact and part synergy terms by weighting. The goal is that constraints are no longer scattered across different modules, but instead form an adjustable and comparable structure within the same objective function. After semantic encoding and constraint modeling, the high-level meaning, phase change and execution boundary of role behavior are stably expressed, which establishes the premise for subsequent policy update and value evaluation.

4.3 Learning Strategy for animation character behavior Generation based on deep reinforcement learning

After feature extraction, state representation, and semantic constraint modeling, animated character behavior generation can be formulated as a continuous decision task. At each moment, the role does not directly output the final action clip, but selects the next action response according to the current state, and then modifies the subsequent strategy through the environmental feedback. There are obvious differences between this modeling approach and the traditional frame-by-frame regression. Frame-by-frame regression relies more on the local correspondence of training samples, while deep reinforcement learning emphasizes the accumulation of rewards under long-term temporal conditions, so that action continuity, semantic satisfaction and physical stability can be coordinated in a unified training objective.

In this paper, the actor-critic structure is used in the policy layer. The actor network is responsible for sampling the behavior action according to the current state, and the critic network is responsible for evaluating the value of the action in the current context. The two are updated in a closed-loop through the reward signal. For animated characters, the significance of this structure is not to simply imitate the existing sequence, but to enable the system to select the output among multiple candidate actions that is more suitable for the current scene and semantic goal. If the action only meets the instantaneous pose requirements, but destroys the smoothness of the subsequent trajectory, the critic network will reduce its score in the value estimation, so as to promote the policy to gradually converge to a more stable behavior path.

To illustrate how the learned policy operates during the training phase, Fig. 6 illustrates the closed-loop flow from state input to policy update in character behavior generation. The unified state vector is shown on the left, actor sampling, action execution, and reward calculation in the middle, and critic evaluation and parameter writeback on the right. Through this training link, state information, action results and value feedback constantly interact in the same loop, and the character behavior strategy gradually converges.

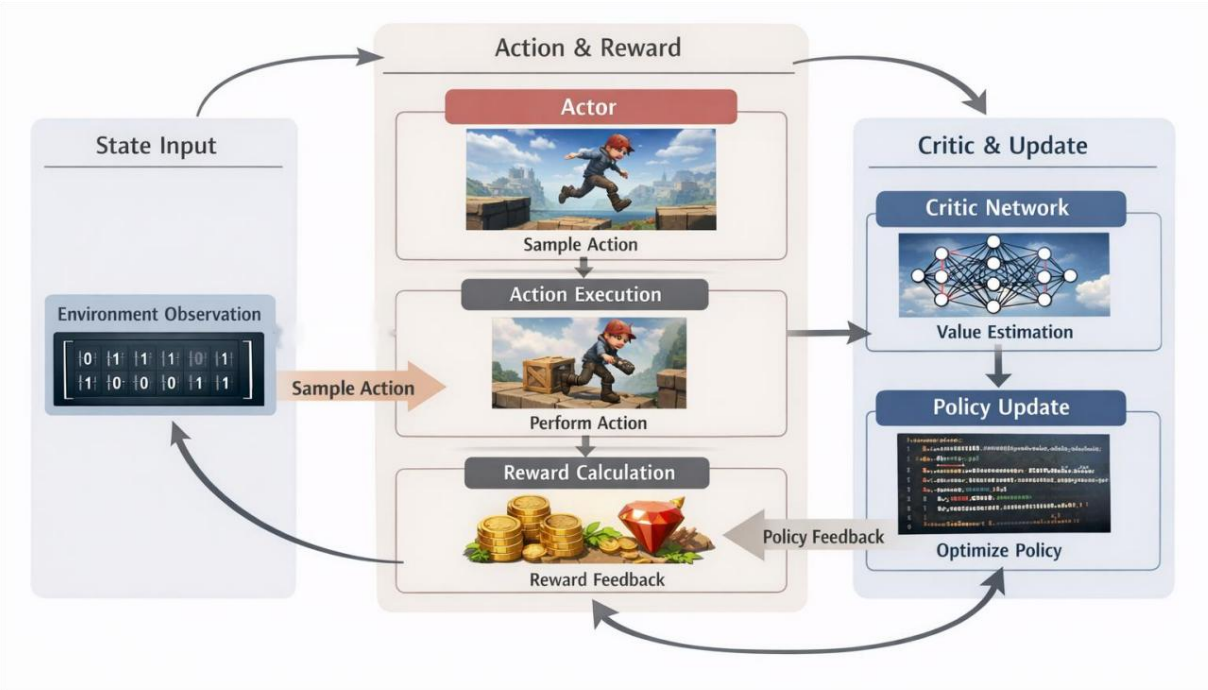

Figure 6: Closed-loop flow diagram of deep reinforcement learning training

In order to enable the actor network to output an explorable action distribution in continuous states, the policy sampling function is defined as follows:

$$a_t \sim \pi_\theta(a_t|s_t) = \mathcal{N}(\mu_\theta(s_t), \Sigma_t) \tag{12}$$

Here, $\pi_\theta(a_t|s_t)$ denotes the conditional policy distribution of the actor network output action a_t state s_t, θ denotes the policy parameters,$\mu_\theta(s_t)$ denotes the policy mean, and Σ_t denotes the covariance matrix. This formula makes the action sampling not limited to fixed output, but to cover a variety of feasible behaviors in the form of distribution, which provides exploration space for the action selection of characters in turning, stopping and interaction scenes. The existence of policy distribution enables the model to maintain sufficient search ability in the early stage of training, and gradually shrink to a more stable behavior region in the later stage.

After obtaining the action sampling, the critic network needs to judge whether the current action conforms to the long sequence of revenue maximization direction. Relying only on the immediate error will make the model biased towards the local optimum, while the value estimation can take the motion benefit of several future frames into account simultaneously. Thus, instead of statically scoring a single pose, the critic evaluates the long-term significance of the current action in terms of state transition and cumulative reward.

In order to enable the critic network to estimate the impact of the current action on the future payoff, the target value recurrence relation is defined as follows.

$$y_t = r_t + \gamma Q_{\bar{\omega}}(s_{t+1}, \mu_{\bar{\theta}}(s_{t+1})) \tag{13}$$

where y_t represents the critic target value, r_t represents the immediate reward, γ represents the discount factor, $Q_{\bar{\omega}}(s_{t+1}, \mu_{\bar{\theta}}(s_{t+1}))$ represents the value estimate of the target critic network for the state-action pair at the next time, ω represents the target critic parameters, θ represents the target actor parameters. This formula introduces future benefits into current evaluation and makes value learning have the property of time recurrence. The critic network can suppress the action output that is reasonable in the short term but will lead to subsequent imbalance, and enhance the consistency of the character's behavior in the long time axis.

To further illustrate the execution path of the inference stage, Fig. 7 shows the online inference process of animated character behavior generation after the training is completed. In the figure, the state input first enters the actor network to get action suggestions, and then generates the final skeleton sequence through semantic constraint verification and kinematic decoding, and then feeds back to the next time state. Along this execution path, the actor still maintains the continuous connection between state recursion, action modification and result output in the inference stage, so it can maintain good action stability in long sequence generation.

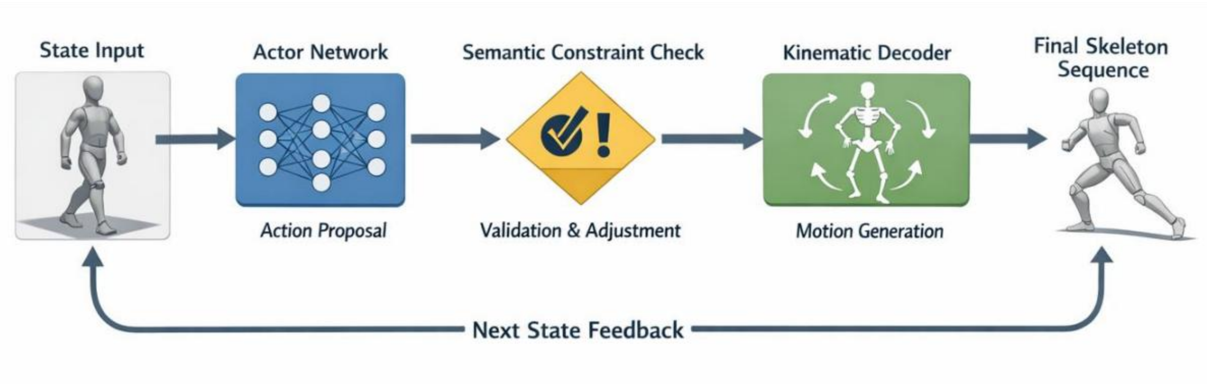

Figure 7: Flowchart of online reasoning for animated character behavior generation

In order to make the critic network obtain stable and reliable value judgments in the training phase, this paper defines the loss function as follows:

$$L_Q = \frac{1}{N}\sum_{t=1}^{N}(Q_\omega(s_t, a_t) - y_t)^2 \tag{14}$$

Here, L_Q represents the critic loss, $Q_\omega(s_t, a_t)$ represents the value estimate of the current critic network output, and N represents the number of samples in the batch. This formula drives the critic network to obtain more reliable evaluation ability by minimizing the error between the predicted value and the target value. Once the value estimation is stable, the update direction received by the policy network will be clearer, and the large oscillations and false switching in the character's behavior sequence will be significantly reduced.

In order to make the actor network continue to complete the policy update along the high-value direction, the optimization objective with entropy regularization term is defined as follows.

$$L_{\pi} = -\frac{1}{N}\sum_{t=1}^{N} Q_{\omega}\left(s_t, \mu_{\theta}(s_t)\right) - \beta H(\pi_{\theta}) \tag{15}$$

Here, L_{π} represents the actor network objective, $Q_{\omega}(s_t, \mu_{\theta}(s_t))$ represents the value estimate under the current policy action, $H(\pi_{\theta})$ represents the policy entropy regularization term, and β represents the entropy weight. This formula encourages the policy to move towards high-value regions, while retaining a moderate exploration capacity to avoid getting stuck in a single path too early in the training process. After the alternating update of actor and critic, the behavior generation strategy can form a more balanced output mechanism between action naturalness, semantic consistency and constraint satisfaction.

Overall, this section establishes a complete learning chain from policy sampling, value evaluation to parameter update. The actor network is responsible for the direction of action generation, and the critic network is responsible for the long-term benefit judgment. Both of them work together on the character behavior sequence driven by uniform reward. After the construction of this learning strategy, the model no longer stays at the static pose fitting level, but has the ability to make continuous decisions and continuous corrections for complex scenes.

5 Performance verification and effect analysis of animation character behavior generation algorithm based on deep reinforcement learning

5.1 Verification of algorithm performance

In order to verify the effectiveness of the proposed deep reinforcement learning animation character behavior generation algorithm in multi-scene sequence control, CMU Mocap, Mixamo and self-built interactive action clips are selected to form a unified data set, which contains 12480 behavior sequences. The training set, validation set and test set are divided by 6：2：2. Each segment sample contains skeleton trajectories, action labels, scene targets, and contact markers, which are uniformly resampled to 30 fps. The comparison methods are selected as GRU, Transformer, PPO-RNN and the proposed method. The batch size in the training phase is set to 256, the initial learning rate is 0.0003, the discount factor is 0.99, the policy temperature coefficient is 0.01, and the maximum number of training rounds is 180. The evaluation indexes are action accuracy, macro average F1 value, trajectory offset error and structural similarity to investigate the classification consistency, temporal stability and action recovery quality of the model.

To visually present the convergence differences in different training stages, Fig. 8 shows the accuracy versus F1 change heatmaps of the four methods on the test set.

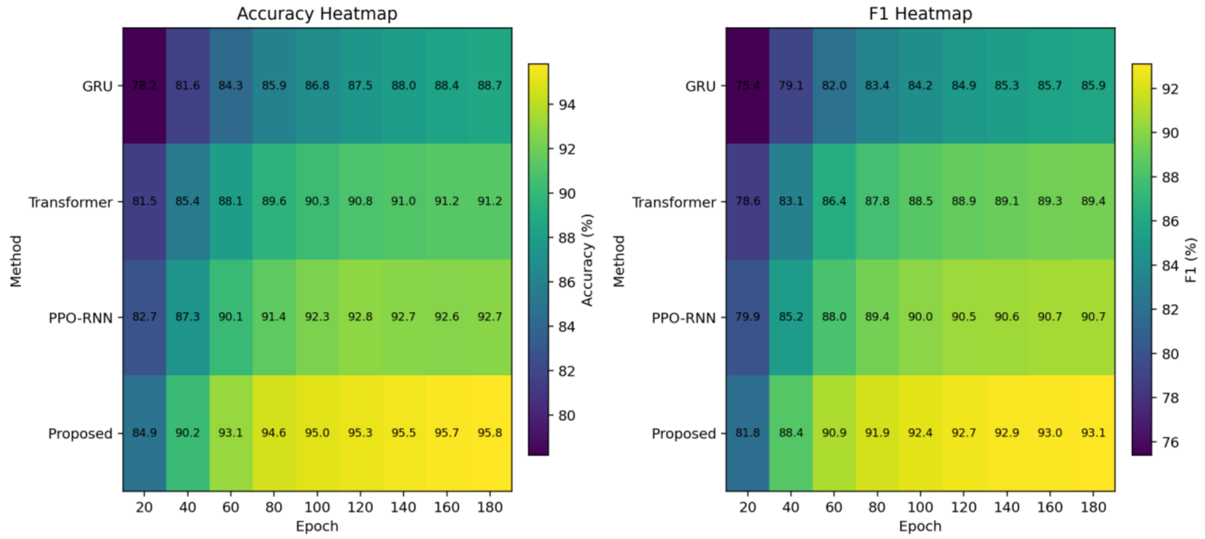

Figure 8: Heatmaps of accuracy versus F1 change for different algorithms

It can be seen from Fig. 8 that the proposed method has entered a stable improvement interval after the 40th round, the accuracy is maintained at more than 94.5% after the 80th round, and the F1 value is stable at more than 91.8%. The improvement of Transformer slows down after the 60th round, and the accuracy stays at about 91.2% in the later stage. PPO-RNN climbed faster in the early stage, but fluctuated obviously after the 100th round. The overall convergence of GRU is slow, and the accuracy is still lower than 89.0% at the end of training. This shows that the proposed method establishes synergy between state modeling and policy update.

To compare the distribution of action quality and trajectory deviation, box plots are used in Fig. 9 to show the trajectory deviation errors of the four methods on the test set.

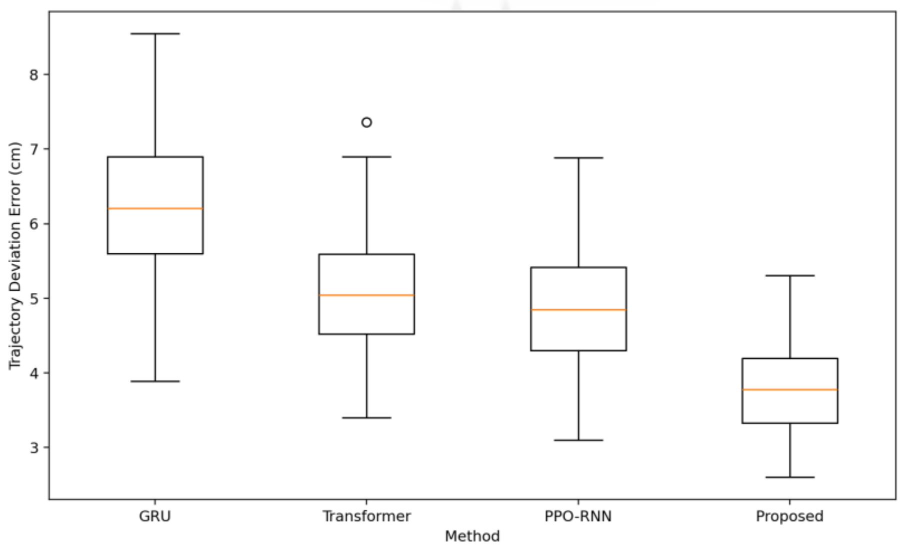

Figure 9: Box plots of trajectory offset errors for different algorithms

Fig. 9 shows that the median error of the proposed method is 3.84 cm, the interquartile range is concentrated between 3.12 cm and 4.57 cm, and the number of outliers is the least.

The median error of Transformer is 5.06 cm. PPO-RNN was 4.73 cm; The GRU reaches 6.18 cm. After counting the emergency stop, turn and avoidance segments separately, the extreme error of the proposed method is still controlled within 8.41 cm, while the maximum error of the other methods on the same segment is more than 10 cm. The error distribution results show that the deep reinforcement learning strategy has higher temporal stability when dealing with action transition and goal modification.

To compare the main performance metrics, Table 3 summarizes the quantified results of the four algorithms on the test set. The data in the table are from the average of five independent repeated experiments, which can reflect the comprehensive performance of the model in terms of accuracy, smoothness and reconstruction quality.

Table 3: Test set performance comparison of different algorithms

Method	Action Accuracy / %	Macro-F1 / %	Trajectory Deviation Error / cm	Structural Similarity
GRU	88.73	85.91	6.18	0.861
Transformer	91.24	89.37	5.06	0.889
PPO-RNN	92.68	90.74	4.73	0.901
Proposed Method	95.84	93.12	3.84	0.927

Table 3 shows that the proposed method achieves the highest results in the three indicators of action accuracy, F1 value and structural similarity, while maintaining the lowest trajectory offset error. Compared with Transformer, the action accuracy of the proposed method is increased by 4.60 percentage points, and the trajectory offset error is reduced by 1.22 cm. Compared with PPO-RNN, the structural similarity is increased by 0.026, indicating that the critic value evaluation module plays a more obvious role in action detail retention and long-term temporal stability control.

The comprehensive comparison results show that the proposed method has a relatively stable performance advantage in the task of animation character behavior generation. The action details, trend changes and environmental conditions were jointly encoded in the state representation layer, so that the model could still maintain high recognition accuracy in complex segments. The valuation provided by the critic network, in turn, further compresses the cumulative bias in the generation of long sequences. Compared with several comparison methods, the performance of the proposed method in accuracy, F1 value, trajectory offset error and structural similarity is more balanced, indicating that the deep reinforcement learning framework not only improves the local action judgment results, but also enhances the continuous control ability of the whole behavior sequence. This result provides a more reliable performance basis for the subsequent generation effect analysis.

5.2 Generation effect analysis

After completing the algorithm performance verification, this paper further analyzes the actual effect of the model from three levels of generation results, emotional response and user perception. In the testing phase, five types of behavior segments including walking, stopping, looking back, avoiding and interacting are selected, and 960 representative samples are selected from the test set for visual evaluation. To ensure a fair comparison, the same skeleton topology, action decoder and rendering parameters are used in all four methods. In addition to the conventional accuracy indicators, this section further calculates the behavior matching rate, keypoint mean square error, action fluency score, and visual naturalness score, so as to conduct a more detailed test of the proposed method from the generation quality level.

To observe the overall differences of different methods in multi-dimensional visual indicators, radar plots are used in Fig. 10 to show the average performance of the four algorithms in behavior matching rate, pose coherence, trajectory smoothness, semantic consistency and visual naturalness.

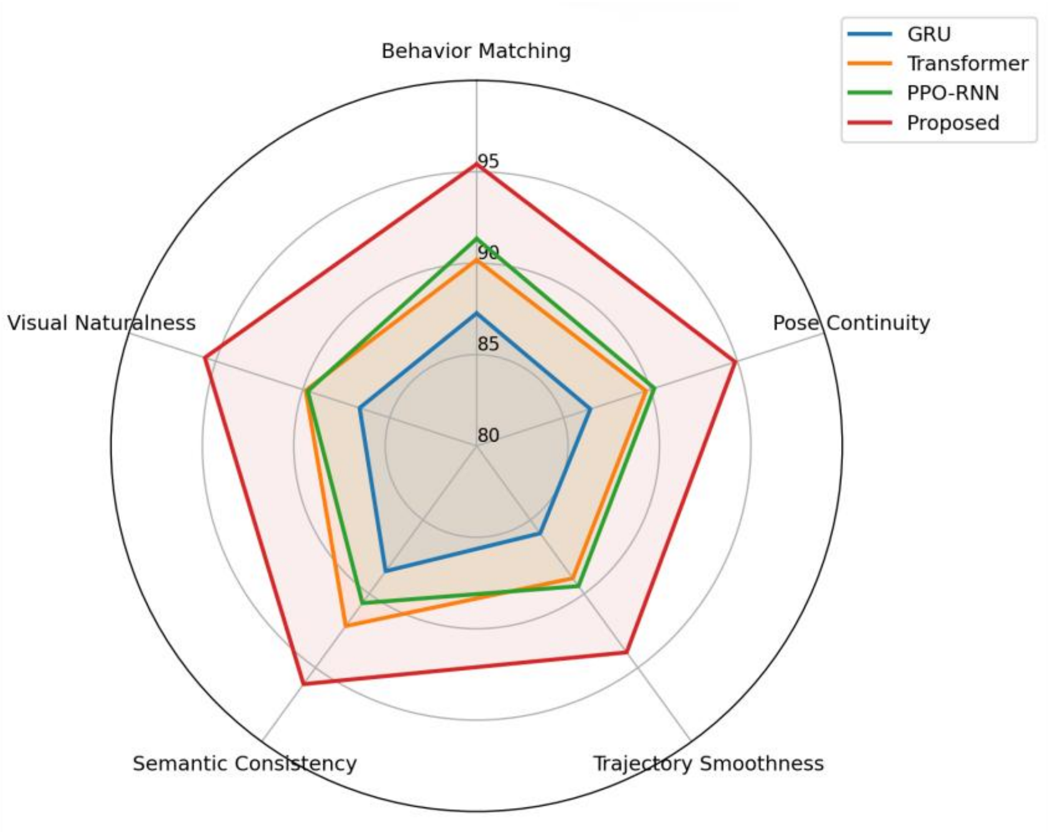

Figure 10: Comprehensive radar chart of different algorithm generation effects

Fig. 10 shows that the proposed method achieves the highest scores in five dimensions, where the behavior matching rate is 95.42, the pose coherence is 94.87, the trajectory smoothness is 93.96, the semantic consistency is 96.11, and the visual naturalness is 95.63. Transformer has a strong performance in the semantic consistency dimension with a score of 92.18, but it is lower than 90 in the trajectory smoothness dimension. The behavior matching rate of PPO-RNN reaches 91.34, but the visual naturalness decreases significantly. GRU did not exceed 89 in any of the five measures. It can be seen that the proposed method establishes a more balanced generation result between action shape, temporal continuity and high-level semantics.

To further examine the distribution of keypoint errors in different contexts, Fig. 11 shows the violin plots of keypoint mean square errors of the four methods on five types of behavior segments.

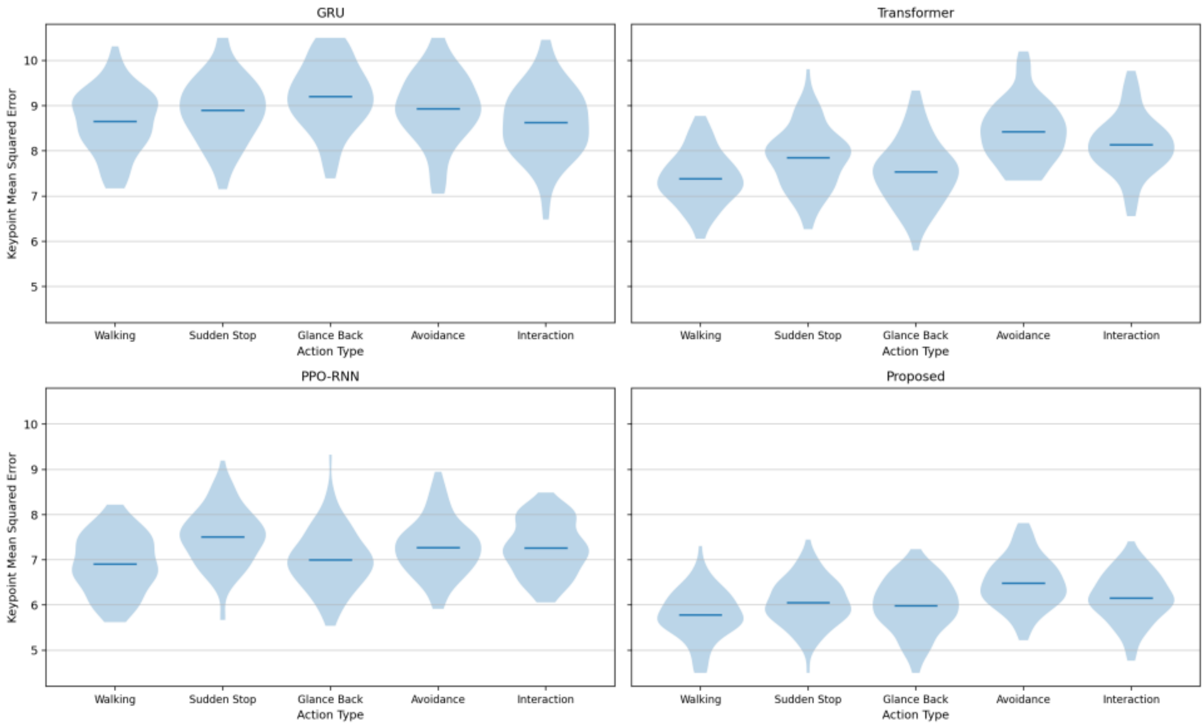

Figure 11: Violin plot of keypoint error for different action types

It can be seen from Fig. 11 that the error distribution of the proposed method on the five types of behaviors is relatively concentrated, and the mean interval is maintained between 5.82 and 6.47. Transformer has large fluctuations in avoidance and interaction actions, and the maximum error is close to 8.90. The bimodal distribution of PPO-RNN in the emergency stop segment indicates that it still has unstable output at the action switching node. The overall error of GRU is high, especially in the look-back action, the mean value reaches 9.14. The results show that the proposed method has stronger detail preservation ability for complex behavior switching.

To examine the contribution of each component module to the final effect, Table 4 presents the ablation experiment results. In the experiment, the state trend coding module, the semantic constraint module and the critic value evaluation module were removed in turn to observe the changes of key indicators. The ablation results were able to illustrate the role of each module in behavioral continuity and visual stability.

Table 4: Ablation experimental results of the proposed method

Model Setting	Behavior Matching Rate / %	Keypoint Mean Squared Error	Fluency Score	Naturalness Score
Full Model	95.42	6.11	9.36	9.28
Without State Trend Encoding Module	92.37	7.24	8.91	8.84
Without Semantic Constraint Module	91.68	7.83	8.76	8.61
Without Critic Value Evaluation Module	90.94	8.12	8.55	8.47

It can be seen from Table 4 that the three key modules have direct contributions to the generation quality. Among them, after the critic value evaluation module is removed, the behavior matching rate decreases by 4.48 percentage points, and the mean square error of key points increases to 8.12, indicating that long-term value judgment has a strong constraint effect on action selection direction and timing balance. When the semantic constraint module is removed, the naturalness score decreases most obviously, reflecting that high-level semantic control has a clear impact on behavioral appearance coordination.

Combined with the experimental results, it can be seen that the proposed method maintains good action consistency and visual completion in the generation stage. The state trend encoding ensures the rhythm continuity in the process of action advancement, the semantic constraint module maintains the correspondence between the behavior category and the posture expression, and the critic value evaluation forms a continuous modification to the stable output of the long-term sequence of actions. After the three parts work together, the action cohesion of the character in the walking, avoiding, looking back and interaction segments is more natural, and the fluctuation of local key points is effectively controlled. Therefore, the proposed method not only performs better in numerical indicators, but also has a strong supporting role in behavioral coherence and visual credibility after actual rendering.

6 Discussion

The animation character behavior generation algorithm based on deep reinforcement learning proposed in this paper performs well in terms of action accuracy, trajectory offset error and visual naturalness, indicating that a stable synergistic relationship has been formed between state representation, semantic constraint and value evaluation. Compared with the traditional sequence generation method, the proposed framework has stronger adaptability in long-term action cohesion, scene response and continuous behavior control. However, the current model still has room for improvement. At present, the training process mainly relies on offline samples, and when facing new action combinations and complex interactions in open scenes, policy generalization is still limited by data distribution. Although the semantic constraint module improves the consistency between behavior categories and action trajectories, the description of fine-grained style differences, character habits and performance rhythm changes is not sufficient, and some segments still have local rigidity when stopping, turning and switching between interactions. Critic network can compress the cumulative error in long sequences, but it is expensive to train and sensitive to the size of video memory and batch scheduling. The follow-up research can be promoted from three aspects. First, a more detailed hierarchical action prior is constructed, and the role style, shot semantics and task goal are integrated into the state modeling. Secondly, online incremental learning and distributed training are introduced to improve the update efficiency of the model in large-scale data and dynamic scenes. Thirdly, physical constraints, emotional signals and interactive events are further integrated to form a real-time responsive behavior generation framework, which makes the generation results achieve a more balanced performance between continuity, controllability and rendering stability.

Funding

Principal Investigator of the 2025 Scientific Research Project of Jilin Provincial Department of Education. Project Title: Research on the Path of Integrating Animation Works into Aesthetic Education and Ideological & Political Education from the Perspective of "Three Comprehensive Education" (Project No.: 20251993G270165)

About the Author

Guan Xiaoxuan was born in Changchun, Jilin Province in 1982. He received her bachelor's degree from Jilin University of the Arts in 2005 and her master's degree from the same university in 2012, with a research focus on 2D Digital Animation. He is currently a teacher in the 2D Animation Department, School of Animation Art, Jilin Animation Institute.

Zhang Jian was born in Fenyang, Shanxi Province in 1986. He received his Bachelor's degree from Jilin University of the Arts in 2010 and his Master's degree from Changchun Normal University in 2020, majoring in 2D Digital Animation. He is currently Deputy Director of the 2D Animation Department, School of Animation Art, Jilin Animation Institute.

References

[1] Kwiatkowski A, Alvarado E, Kalogeiton V, et al. A survey on reinforcement learning methods in character animation[C]//Computer graphics forum. 2022, 41(2): 613-639.

[2] Qin W, Tao R, Sun L, et al. Muscle-driven virtual human motion generation approach based on deep reinforcement learning[J]. Computer Animation and Virtual Worlds, 2022, 33(3-4): e2092.

[3] Zhu W, Ma X, Ro D, et al. Human motion generation: A survey[J]. IEEE Transactions on Pattern Analysis and Machine Intelligence, 2023, 46(4): 2430-2449.

[4] Zhang M, Cai Z, Pan L, et al. Motiondiffuse: Text-driven human motion generation with diffusion model[J]. IEEE transactions on pattern analysis and machine intelligence, 2024, 46(6): 4115-4128.

[5] Gao X, Yang Y, Xie Z, et al. Guess: Gradually enriching synthesis for text-driven human motion generation[J]. IEEE Transactions on Visualization and Computer Graphics, 2024, 30(12): 7518-7530.

[6] Hu L, Zhang Z, Zhong C, et al. Pose-aware attention network for flexible motion retargeting by body part[J]. IEEE Transactions on Visualization and Computer Graphics, 2023, 30(8): 4792-4808.

[7] Ji B, Pan Y, Yan Y, et al. Stylevr: Stylizing character animations with normalizing flows[J]. IEEE Transactions on Visualization and Computer Graphics, 2023, 30(7): 4183-4196.

[8] Zhang F, Wang Z, Lyu X, et al. Speech-driven personalized gesture synthetics: Harnessing automatic fuzzy feature inference[J]. IEEE Transactions on Visualization and Computer Graphics, 2024, 30(10): 6984-6996.

[9] Hu L, Zhang Z, Ye Y, et al. Diffusion-based Human Motion Style Transfer with Semantic Guidance[C]//Computer Graphics Forum. 2024, 43(8): e15169.

[10] Sun L, Wang Y, Qin W. A language-directed virtual human motion generation approach based on musculoskeletal models[J]. Computer Animation and Virtual Worlds, 2024, 35(3): e2257.

[11] Yang Z, Wen Y H, Chen S Y, et al. Keyframe control of music-driven 3d dance generation[J]. IEEE Transactions on Visualization and Computer Graphics, 2023 (7): 3474-3486.

[12] Niu L, Xie W, Wang D, et al. Audio2AB: Audio-driven collaborative generation of virtual character animation[J]. Virtual Reality & Intelligent Hardware, 2024, 6(1): 56-70.

[13] Akber S M A, Kazmi S N, Mohsin S M, et al. Deep learning-based motion style transfer tools, techniques and future challenges[J]. Sensors, 2023, 23(5): 2597.

[14] Lungu-Stan V C, Mocanu I G. 3D character animation and asset generation using deep learning[J]. Applied Sciences, 2024, 14(16): 7234.

[15] Jiang D, Chang J, You L, et al. Audio-driven facial animation with deep learning: A survey[J]. Information, 2024, 15(11): 675.

[16] Wu R, Yu Y, Zhan F, et al. Audio-driven talking face generation with diverse yet realistic facial animations[J]. Pattern Recognition, 2023, 144: 109865.

[17] Zhai S, Liu M, Li Y, et al. Talking face generation with audio-deduced emotional landmarks[J]. IEEE Transactions on Neural Networks and Learning Systems, 2023, 35(10): 14099-14111.

[18] Liu S, Wang H. Talking face generation via facial anatomy[J]. ACM Transactions on Multimedia Computing, Communications and Applications, 2023, 19(3): 1-19.

[19] Pan Y, Tan S, Cheng S, et al. Expressive talking avatars[J]. IEEE Transactions on Visualization and Computer Graphics, 2024, 30(5): 2538-2548.

[20] Zhu C A, Joslin C. A review of motion retargeting techniques for 3D character facial animation[J]. Computers & Graphics, 2024, 123: 104037.

[21] Wagner N, Schwanecke U, Botsch M. Anacondar: Anatomically-constrained data-adaptive facial retargeting[J]. Computers & Graphics, 2024, 122: 103988.